COMPUTER NETWORKS
LECTURE NOTES

B.TECH III YEAR – I SEM (R18)

(2021-22)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

SREYAS INSTITUTE OF ENGINEERING & TECHNOLOGY

CS503PC: COMPUTER NETWORKS

III Year B.Tech. CSE I-Sem

L	T	P	C
3	0	0	3

Prerequisites

1. A course on “Programming for problem solving”
2. A course on “Data Structures”

Course Objectives

1. The objective of the course is to equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

Course Outcomes

1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT - I

Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.
Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

UNIT - II

Data link layer: Design issues, framing, Error detection and correction.
Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.
Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.
Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III

Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, The Network layer in the internet.

UNIT - IV

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V

Application Layer –Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.

TEXT BOOK:

1. Computer Networks -- Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI

REFERENCE BOOKS:

1. An Engineering Approach to Computer Networks-S. Keshav, 2nd Edition, Pearson Education
2. Data Communications and Networking – Behrouz A. Forouzan. Third Edition TMH.

UNIT -1 IMPORTANT QUESTIONS (LONG ANSWER) FOR END SEMESTER/FINAL EXAM

1) EXPLAIN FUNCTIONALITIES OF LAYERS IN ISO OSI MODEL
2) EXPLAIN FUNCTIONALITIES OF LAYERS IN TCP/IP MODEL
3) EXPLAIN VARIOUS GUIDED TRANSMISSION MEDIA
4) EXPLAIN VARIOUS UNGUIDED TRANSMISSION MEDIA
5) EXPLAIN TYPES OF NETWORKS LIKE LAN,MAN,WAN

UNIT - I

NETWORKS

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

"Computer network" to mean a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information.

The connection need not be via a copper wire; fiber optics, microwaves, infrared, and communication satellites can also be used.

Networks come in many sizes, shapes and forms, as we will see later. They are usually connected together to make larger networks, with the **Internet** being the most well-known example of a network of networks.

There is considerable confusion in the literature between a **computer network** and a **distributed system**. The key distinction is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Usually, it has a single model or paradigm that it presents to the users. Often a layer of software on top of the operating system, called **middleware**, is responsible for implementing this model. A well-known example of a distributed system is the **World Wide Web**. It runs on top of the Internet and presents a model in which everything looks like a document (Web page).

USES OF COMPUTER NETWORKS

1. Business Applications

to distribute information throughout the company (**resource sharing).** sharing physical resources such as printers, and tape backup systems, is sharing information

client-server model. It is widely used and forms the basis of much network usage.

communication medium among employees.**email** (**electronic mail**), which employees generally use for a great deal of daily communication.

Telephone calls between employees may be carried by the computer network instead of by the phone company. This technology is called **IP telephony** or **Voice over IP** (**VoIP**) when Internet technology is used.

Desktop sharing lets remote workers see and interact with a graphical computer screen

doing business electronically, especially with customers and suppliers. This new model is called **e-commerce** (**electronic commerce**) and it has grown rapidly in recent years.

2 **Home Applications**

peer-to-peer communication

person-to-person communication

electronic commerce
entertainment.(game playing,)

3 **Mobile Users**
Text messaging or texting
Smart phones,
GPS (Global Positioning System)
m-commerce
NFC (Near Field Communication)

4 **Social Issues**

With the good comes the bad, as this new-found freedom brings with it many unsolved social, political, and ethical issues.

Social networks, message boards, content sharing sites, and a host of other applications allow people to share their views with like-minded individuals. As long as the subjects are restricted to technical topics or hobbies like gardening, not too many problems will arise.

The trouble comes with topics that people actually care about, like politics, religion, or sex. Views that are publicly posted may be deeply offensive to some people. Worse yet, they may not be politically correct. Furthermore, opinions need not be limited to text; high-resolution color photographs and video clips are easily shared over computer networks. Some people take a live-and-let-live view, but others feel that posting certain material (e.g., verbal attacks on particular countries or religions, pornography, etc.) is simply unacceptable and that such content must be censored. Different countries have different and conflicting laws in this area. Thus, the debate rages.

Computer networks make it very easy to communicate. They also make it easy for the people who run the network to snoop on the traffic. This sets up conflicts over issues such as **employee rights versus employer rights**. Many people read and write email at work. Many employers have claimed the right to read and possibly censor employee messages, including messages sent from a home computer outside working hours. Not all employees agree with this, especially the latter part.

Another conflict is centered around government versus citizen's rights.

A new twist with mobile devices is location privacy. As part of the process of providing service to your mobile device the network operators learn where you are at different times of day. This allows them to track your movements. They may know which nightclub you frequent and which medical center you visit.

The effectiveness of a data communications system depends on four fundamental characteristics: delivery, accuracy, timeliness, and jitter.

I. **Delivery.** The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user.

2 **Accuracy.** The system must deliver the data accurately. Data that have been altered in

transmission and left uncorrected are unusable.

3. **Timeliness**. The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called *real-time* transmission.

4. **Jitter**. Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

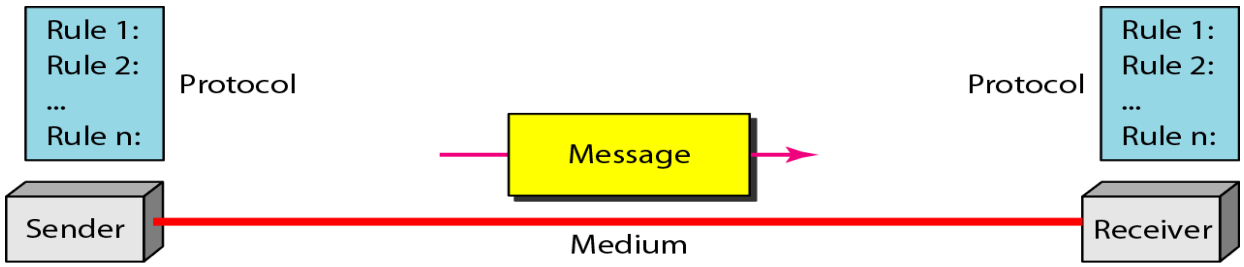
A data communications system has five components

I. **Message**. The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video. 2 **Sender**. The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.

3. **Receiver.** The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

4. **Transmission medium**. The transmission medium is the physical path by which a message travels from sender to receiver. Some examples of transmission media include twisted-pair wire, coaxial cable, fiber-optic cable, and radio waves.

5. **Protocol.** A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.



Data Representation

Text

Numbers

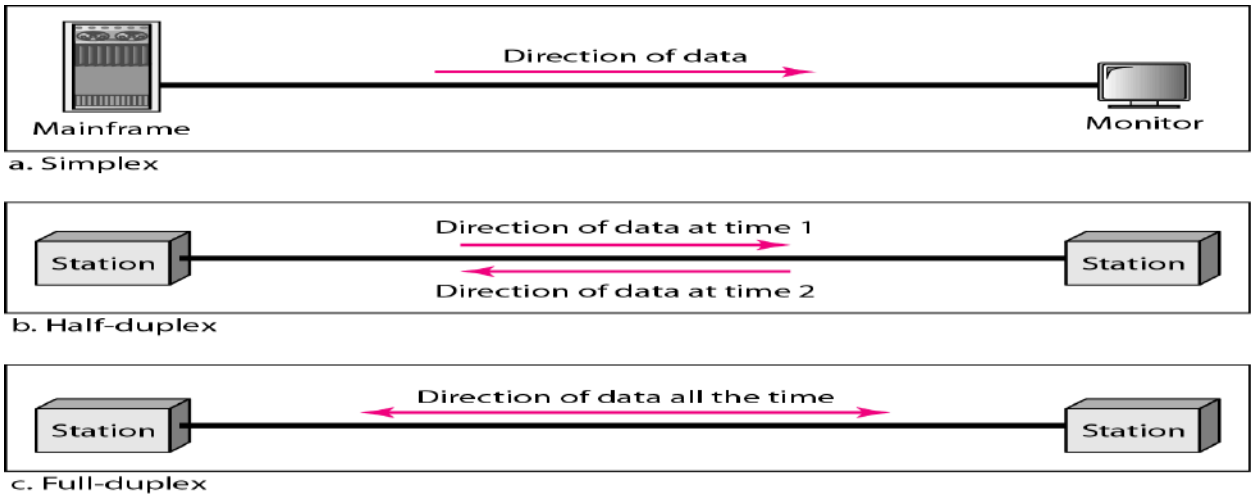
Images

Audio

Video

Data Flow

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



Simplex In simplex mode, the communication is unidirectional, as on a one- way street. Only one of the two devices on a link can transmit; the other can only receive (Figure a). Keyboards and traditional monitors are examples of simplex devices.

Half-Duplex

In half-duplex mode, each station can both transmit and receive, but not at the same time. When one device is sending, the other can only receive, and vice versa (Figure b). Walkie-talkies and CB (citizens band) radios are both half- duplex systems.

Full-Duplex

In full-duplex, both stations can transmit and receive simultaneously (Figure c). One common example of full-duplex communication is the telephone network. When two people are communicating by a telephone line, both can talk and listen at the same time. The full-duplex mode is used when communication in both directions is required all the time.

Network Criteria

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

Performance

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

an inquiry and a response. The performance of a network depends on a number of factors, including the number of users, the type of transmission medium, the capabilities of the connected hardware, and the efficiency of the software.

Performance is often evaluated by two networking metrics: **throughput and delay**. We often need more throughput and less delay. However, these two criteria are often contradictory. If we try to send more data to the network, we may increase throughput but we increase the delay because of traffic congestion in the network.

Reliability: In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

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

Physical Structures

Before discussing networks, we need to define some network attributes.

Type of Connection

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

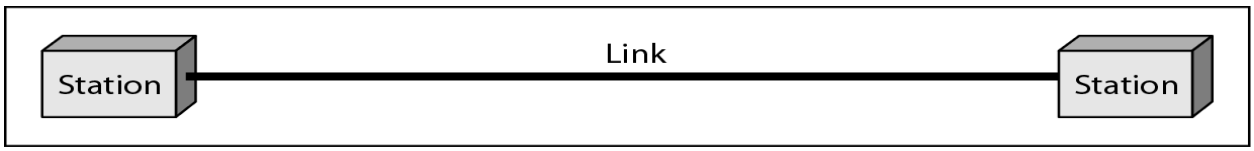
There are two possible types of connections: point-to-point and multipoint.

Point-to-Point A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible

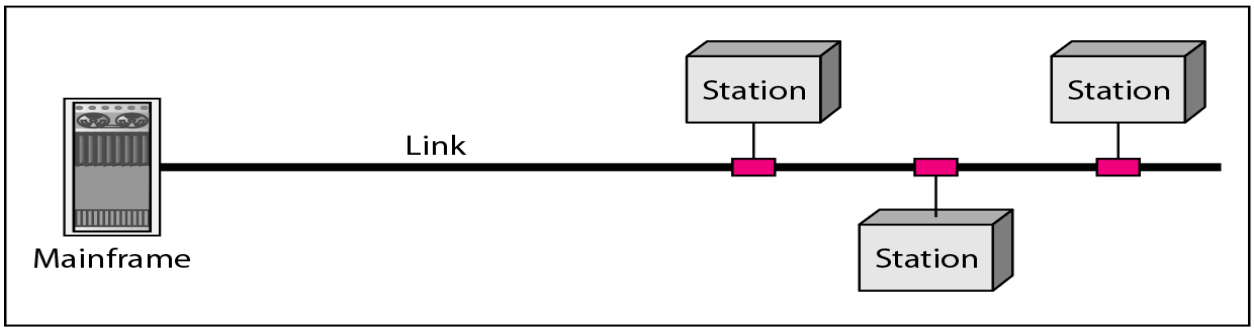
When you change television channels by infrared remote control, you are establishing a point-to-point connection between the remote control and the television's control system.

Multipoint A multipoint (also called multi-drop) connection is one in which more than two specific devices share a single link

In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a *spatially shared* connection. If users must take turns, it is a *timeshared* connection.



a. Point-to-point

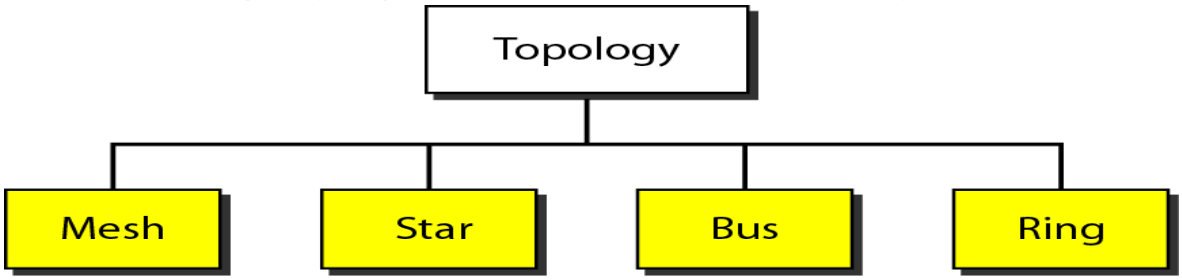


b. Multipoint

Physical Topology

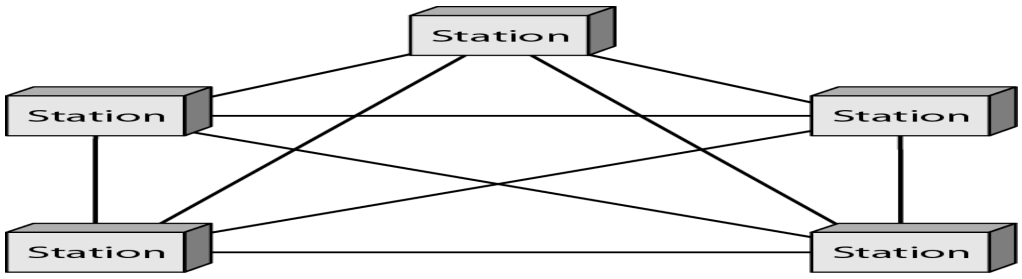
The term *physical topology* refers to the way in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.

There are four basic topologies possible: mesh, star, bus, and ring



MESH:

A mesh topology is the one where every node is connected to every other node in the network.



A mesh topology can be a **full mesh topology** or a **partially connected mesh topology**.

In a *full mesh topology*, every computer in the network has a connection to each of the other computers in that network. The number of connections in this

network can be calculated using the following formula (***n*** is the number of computers in the network): **n(n-1)/2**

In a *partially connected mesh topology*, at least two of the computers in the network have connections to multiple other computers in that network. It is an inexpensive way to implement redundancy in a network. In the event that one of the primary computers or connections in the network fails, the rest of the network continues to operate normally.

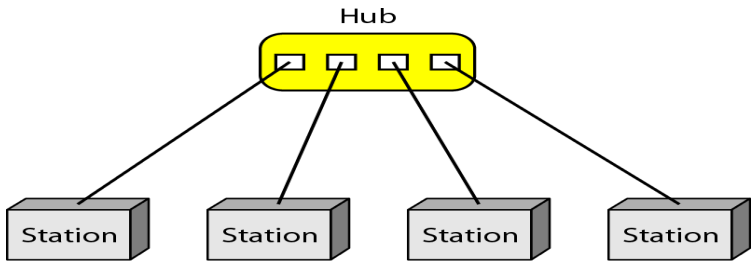
Advantages of a mesh topology

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

Disadvantages of a mesh topology

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

STAR:



A star network, **star topology** is one of the most common network setups. In this configuration, every node connects to a central network device, like a hub, switch, or computer. The central network device acts as a server and the peripheral devices act as clients. Depending on the type of network card used in each computer of the star topology, a coaxial cable or a RJ-45 network cable is used to connect computers together.

Advantages of star topology

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

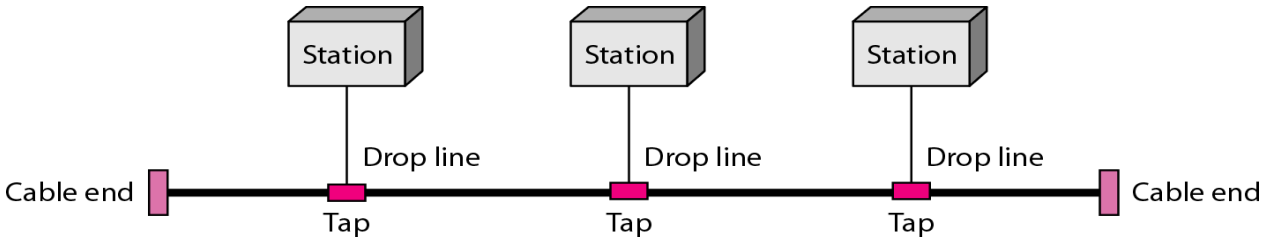
Disadvantages of star topology

Can have a higher cost to implement, especially when using a switch or router as the central network device.

The central network device determines the performance and number of nodes the network can handle.

If the central computer, hub, or switch fails, the entire network goes down and all computers are disconnected from the network

BUS:



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

Advantages of bus topology

It works well when you have a small network.

It's the easiest network topology for connecting computers or peripherals in a linear fashion.

It requires less cable length than a star topology.

Disadvantages of bus topology

It can be difficult to identify the problems if the whole network goes down.

It can be hard to troubleshoot individual device issues.

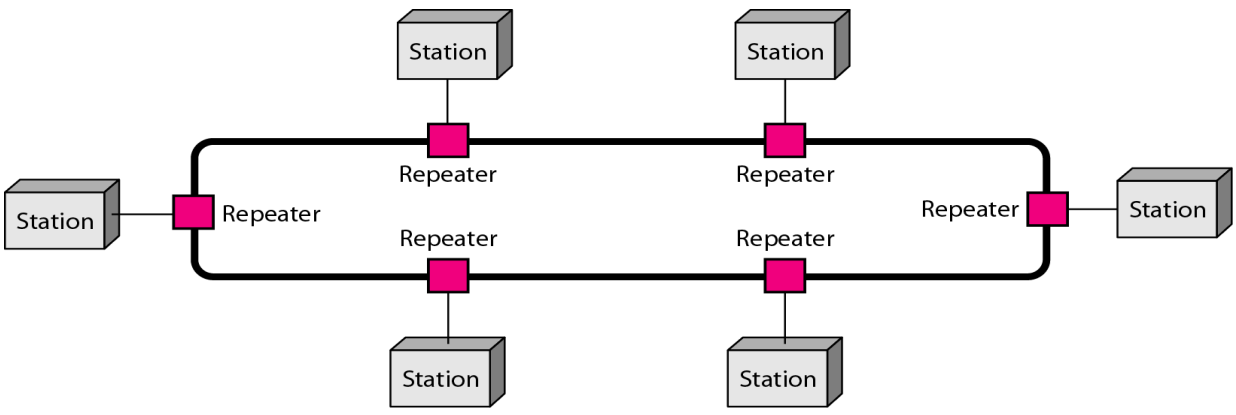
Bus topology is not great for large networks.

Terminators are required for both ends of the main cable.

Additional devices slow the network down.

If a main cable is damaged, the network fails or splits into two.

RING:



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

The major disadvantage of a ring topology is that if any individual connection in the ring is broken, the entire network is affected.

Ring topologies may be used in either local area networks (LANs) or wide area networks (WANs).

Advantages of ring topology

All data flows in one direction, reducing the chance of packet collisions.

A network server is not needed to control network connectivity between each workstation.

Data can transfer between workstations at high speeds.

Additional workstations can be added without impacting performance of the network.

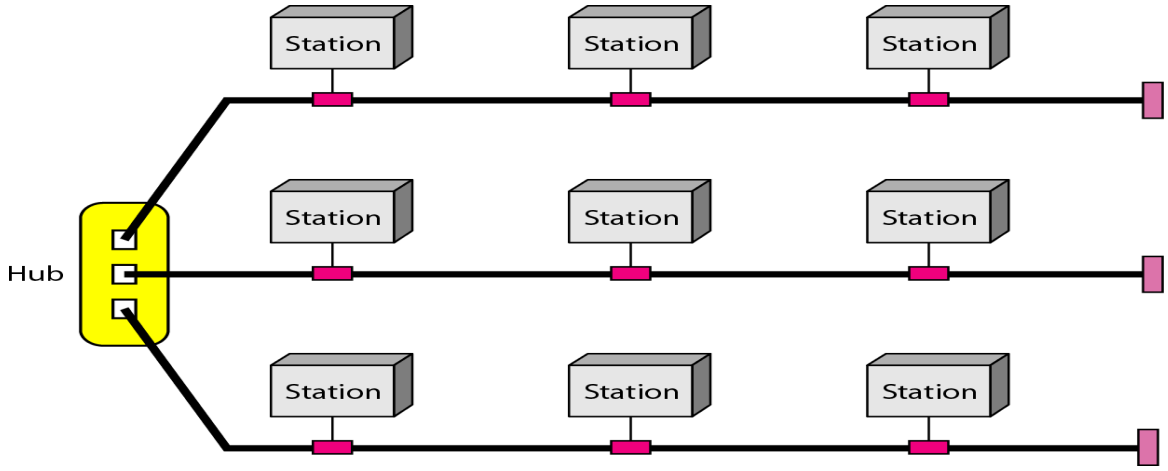
Disadvantages of ring topology

All data being transferred over the network must pass through each workstation on the network, which can make it slower than a star topology.

The entire network will be impacted if one workstation shuts down.

The hardware needed to connect each workstation to the network is more expensive than Ethernet cards and hubs/switches.

Hybrid Topology A network can be hybrid. For example, we can have a main star topology with each branch connecting several stations in a bus topology as shown in Figure



Types of Network based on size

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

Basic types

LAN (Local Area Network)

Group of interconnected computers within a small area. (room, building, campus)

Two or more pc's can from a LAN to share files, folders, printers, applications and other devices.

Coaxial or CAT 5 cables are normally used for connections. Due to short distances, errors and noise are minimum.

Data transfer rate is 10 to 100 mbps.

Example: A computer lab in a school. **MAN (Metropolitan Area Network)** Design to extend over a large area.

Connecting number of LAN's to form larger network, so that resources can be shared.

Networks can be up to 5 to 50 km. Owned by organization or individual. Data transfer rate is low compare to LAN.

Example: Organization with different branches located in the city.

WAN (Wide Area Network)

Are country and worldwide network.

Contains multiple LAN's and MAN's.

Distinguished in terms of geographical range. Uses satellites and microwave relays.

Data transfer rate depends upon the ISP provider and varies over the location. Best example is the internet.

Other types

WLAN (Wireless LAN)

A LAN that uses high frequency radio waves for communication. Provides short range connectivity with high speed data transmission. **PAN (Personal Area Network)**

Network organized by the individual user for its personal use.

SAN (Storage Area Network)

Connects servers to data storage devices via fiber-optic cables. E.g.: Used for daily backup of organization or a mirror copy

---

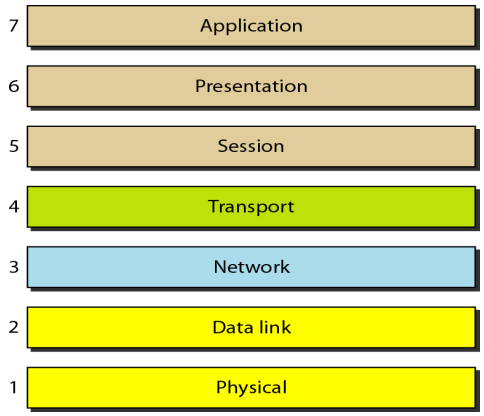
Reference models:

OSI

- OSI stands for Open Systems Interconnection
- Created by International Standards Organization (ISO)
- Was created as a framework and reference model to explain how different networking technologies work together and interact
- It is not a standard that networking protocols must follow

Each layer has specific functions it is responsible for

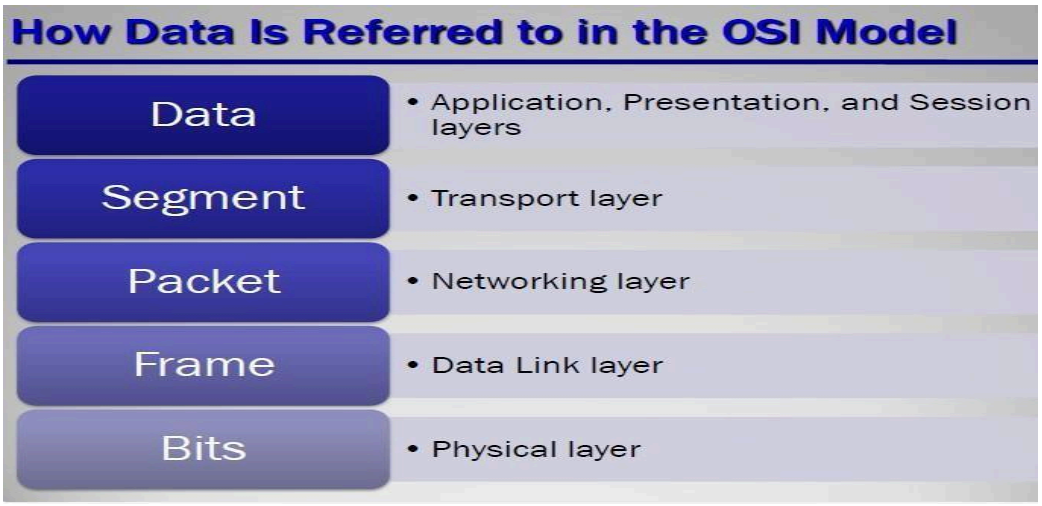
All layers work together in the correct order to move data around a network



Top to bottom

-All People Seem To Need Data Processing Bottom to top

-Please Do Not Throw Sausage Pizza Away



Physical Layer

Deals with all aspects of physically moving data from one computer to the next

Converts data from the upper layers into 1s and 0s for transmission over media

Defines how data is encoded onto the media to transmit the data

Defined on this layer: Cable standards, wireless standards, and fiber optic standards.

Copper wiring, fiber optic cable, radio frequencies, anything that can be used to transmit data is defined on the Physical layer of the OSI Model

Device example: Hub

Used to transmit data

Data Link Layer

Is responsible for moving frames from node to node or computer to computer

Can move frames from one adjacent computer to another, cannot move frames across routers

Encapsulation = frame

Requires MAC address or *physical address*

Protocols defined include Ethernet Protocol and Point-to-Point Protocol (PPP)

Device example: Switch

Two sublayers: Logical Link Control (LLC) and the Media Access Control (MAC)

- Logical Link Control (LLC)
  - –Data Link layer addressing, flow control, address notification, error control
- Media Access Control (MAC)
  - –Determines which computer has access to the network media at any given time
  - –Determines where one frame ends and the next one starts, called frame synchronization

Network Layer

Responsible for moving packets (data) from one end of the network to the other, called *end-to-end communications*

Requires *logical addresses* such as IP addresses

Device example: Router

–Routing is the ability of various network devices and their related software to move data packets from source to destination

Transport Layer

Takes data from higher levels of OSI Model and breaks it into segments that can be sent to lower-level layers for data transmission

Conversely, reassembles data segments into data that higher-level protocols and applications can use

Also puts segments in correct order (called sequencing) so they can be reassembled in correct order at destination

Concerned with the reliability of the transport of sent data

May use a *connection-oriented protocol* such as TCP to ensure destination received segments

May use a *connectionless protocol* such as UDP to send segments without assurance of delivery

Uses port addressing

Session Layer

Responsible for managing the dialog between networked devices

Establishes, manages, and terminates connections

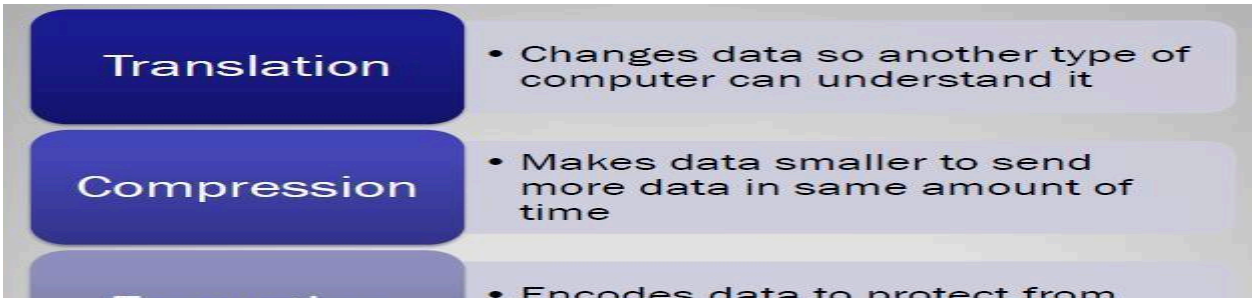
Provides duplex, half-duplex, or simplex communications between devices

Provides procedures for establishing checkpoints, adjournment, termination, and restart or recovery procedures

Presentation Layer

Concerned with how data is presented to the network

Handles three primary tasks: –Translation , –Compression , –Encryption

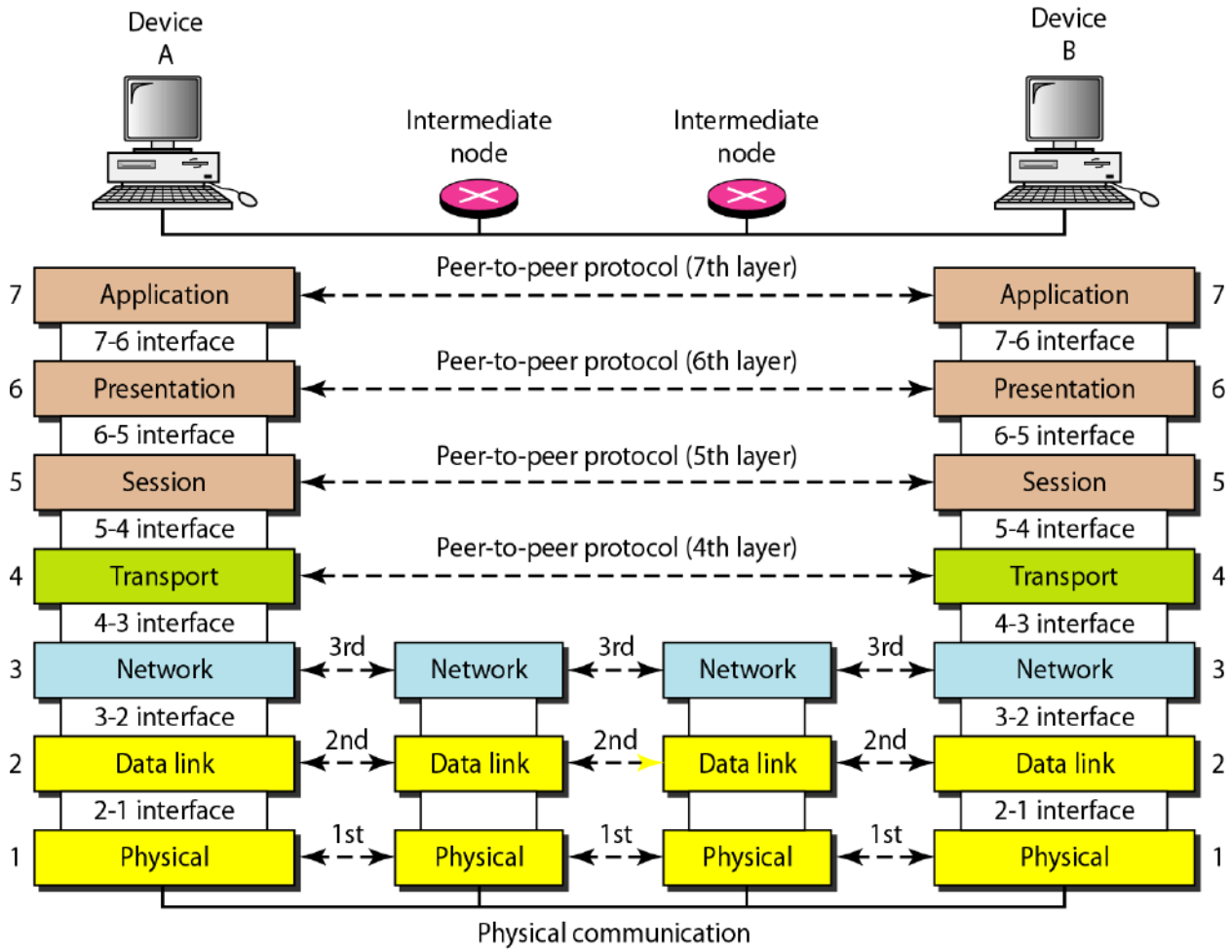


Application Layer

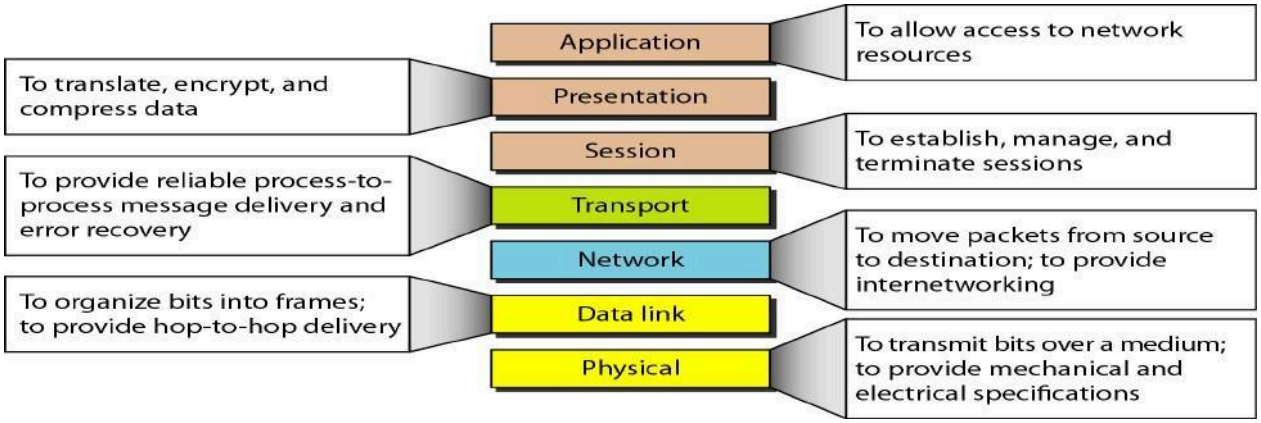
Contains all services or protocols needed by application software or operating system to communicate on the network

Examples

- -Firefox web browser uses HTTP (Hyper-Text Transport Protocol)
- -E-mail program may use POP3 (Post Office Protocol version 3) to read e-mails and SMTP (Simple Mail Transport Protocol) to send e-mails.
- ***The interaction between layers in the OSI model***

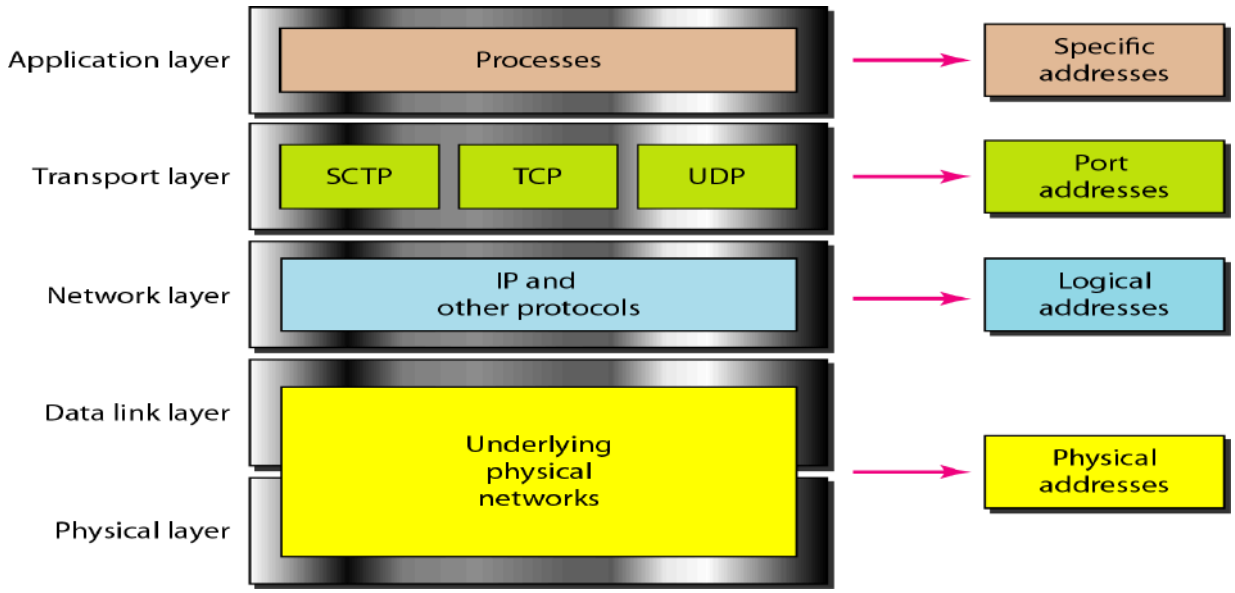


An exchange using the OSI model



TCP/IP Model (Transmission Control Protocol/Internet Protocol)

–A *protocol suite* is a large number of related protocols that work together to allow networked computers to communicate



Relationship of layers and addresses in TCP/IP

Application Layer

Application layer protocols define the rules when implementing specific network applications

Rely on the underlying layers to provide accurate and efficient data delivery

Typical protocols:

- FTP – File Transfer Protocol
  - For file transfer
- Telnet – Remote terminal protocol
  - For remote login on any other computer on the network
- SMTP – Simple Mail Transfer Protocol
  - For mail transfer
- HTTP – Hypertext Transfer Protocol
  - For Web browsing

Encompasses same functions as these OSI Model layers Application Presentation Session

Transport Layer

TCP is a connection-oriented protocol

- Does not mean it has a physical connection between sender and receiver
- TCP provides the function to allow a connection virtually exists – also called virtual circuit

UDP provides the functions:

- Dividing a chunk of data into segments
- Reassembly segments into the original chunk
- Provide further the functions such as reordering and data resend

Offering a reliable byte-stream delivery service

Functions the same as the Transport layer in OSI

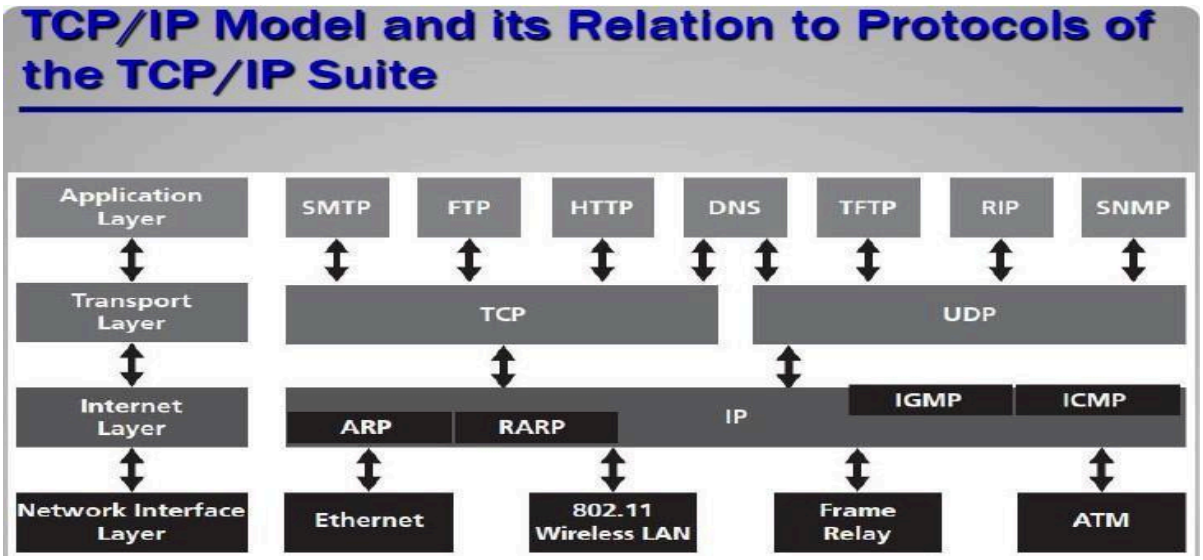
Synchronize source and destination computers to set up the session between the respective computers

Internet Layer

The network layer, also called the internet layer, deals with packets and connects independent networks to transport the packets across network boundaries. The network layer protocols are the IP and the Internet Control Message Protocol (ICMP), which is used for error reporting.

Host-to-network layer

The **Host-to-network layer** is the lowest **layer** of the **TCP/IP** reference model. It combines the link **layer** and the physical **layer** of the ISO/OSI model. At this **layer**, data is transferred between adjacent **network** nodes in a WAN or between nodes on the same LAN.



Differences b/w OSI AND TCP/IP REFERENCE MODEL:

OSI MODEL	TCP/IP MODEL
Contains 7 Layers	Contains 4 Layers
Uses Strict Layering resulting in vertical layers.	Uses Loose Layering resulting in horizontal layers.
Supports both connectionless & connection-oriented communication in the Network layer, but only connection-oriented communication in Transport Layer	Supports only connectionless communication in the Network layer, but both connectionless & connection-oriented communication in Transport Layer
It distinguishes between Service, Interface and Protocol.	Does not clearly distinguish between Service, Interface and Protocol.
Protocols are better hidden and can be replaced relatively easily as technology changes (No transparency)	Protocols are not hidden and thus cannot be replaced easily. (Transparency) Replacing IP by a substantially different protocol would be virtually impossible
OSI reference model was devised before the corresponding protocols were designed.	The protocols came first and the model was a description of the existing protocols

EXAMPLE NETWORKS

THE INTERNET

The Internet has revolutionized many aspects of our daily lives. It has affected the way we do business as well as the way we spend our leisure time. Count the ways you've used the Internet recently. Perhaps you've sent electronic mail (e-mail) to a business associate, paid a utility bill, read a newspaper from a distant city, or looked up a local movie schedule-all by using the Internet. Or maybe you researched a medical topic, booked a hotel reservation, chatted with a fellow Trekkie, or comparison-shopped for a car. The Internet is a communication system that has brought a wealth of information to our fingertips and organized it for our use.

A Brief History

A network is a group of connected communicating devices such as computers and printers. An internet (note the lowercase letter i) is two or more networks that can communicate with each other. The most notable internet is called the Internet (uppercase letter I), a collaboration of more than hundreds of thousands of interconnected networks. Private individuals as well as various organizations such as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet. Millions of people are users. Yet this extraordinary communication system only came into being in 1969.

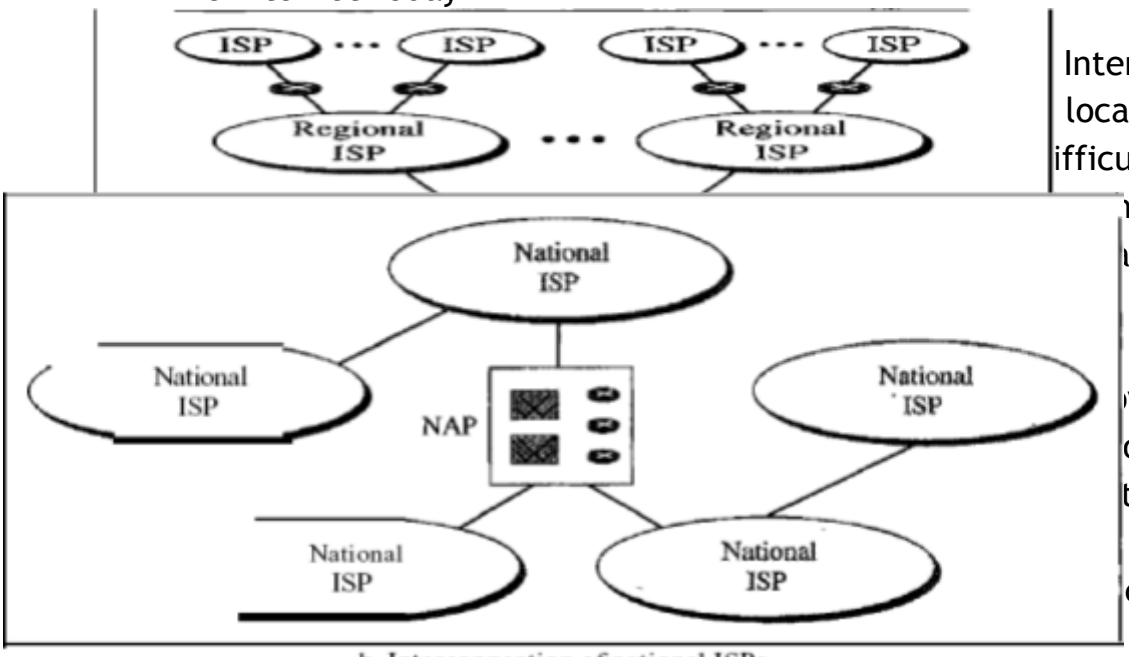
In the mid-1960s, mainframe computers in research organizations were standalone devices. Computers from different manufacturers were unable to communicate with one another. The Advanced Research Projects Agency

(ARPA) in the Department of Defense (DoD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for ARPANET, a small network of connected computers. The idea was that each host computer (not necessarily from the same manufacturer) would be attached to a specialized computer, called an *inteiface message processor* (IMP). The IMPs, in tum, would be connected to one another. Each IMP had to be able to communicate with other IMPs as well as with its own attached host. By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.

In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetting Projec1.* Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end- to-end delivery of packets. This paper on Transmission Control Protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. Shortly thereafter, authorities made a decision to split TCP into two protocols: Transmission Control Protocol (TCP) and Internetworking Protocol (lP). IP would handle datagram routing while TCP would be responsible for higher-level functions such as segmentation, reassembly, and error detection. The internetworking protocol became known as TCPIIP.

The Internet Today



b. Interconnection of national ISPs

Internet today is not a simple
local-area networks joined by
ifficult to give an accurate
anging-new networks are
nd networks of defunct
want Internet connection
are international service
viders, and local service
ot the government. Figure
t.

ce providers that connect

National Internet Service Providers:

The national Internet service providers are backbone networks created and maintained by specialized companies. There are many national ISPs operating in North America; some of the most well known are SprintLink, PSINet, UUNet Technology, AGIS,

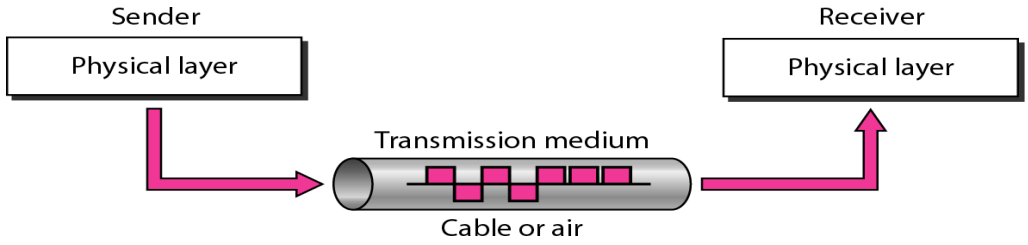
and internet Mel. To provide connectivity between the end users, these backbone networks are connected by complex switching stations (normally run by a third party) called network access points (NAPs). Some national ISP networks are also connected to one another by private switching stations called *peering points*. These normally operate at a high data rate (up to 600 Mbps).

Regional Internet Service Providers:

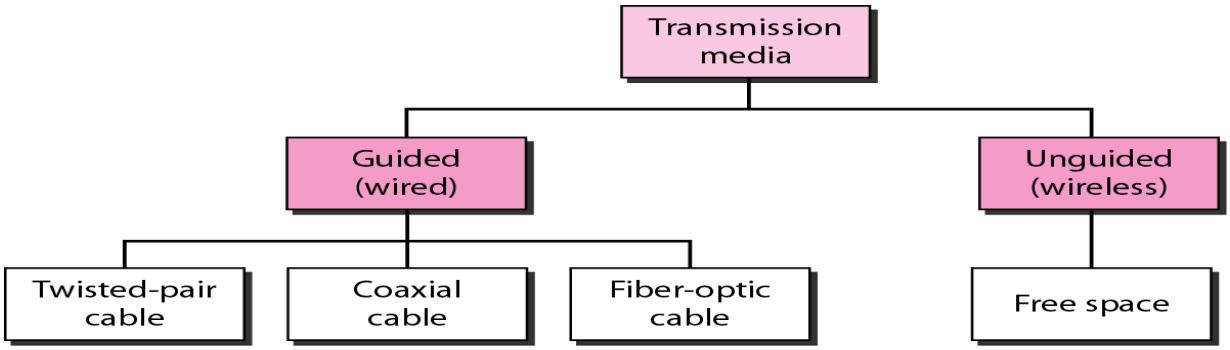
Regional internet service providers or regional ISPs are smaller ISPs that are connected to one or more national ISPs. They are at the third level of the hierarchy with a smaller data rate. ***Local Internet Service Providers:***

Local Internet service providers provide direct service to the end users. The local ISPs can be connected to regional ISPs or directly to national ISPs. Most end users are connected to the local ISPs. Note that in this sense, a local ISP can be a company that just provides Internet services, a corporation with a network that supplies services to its own employees, or a nonprofit organization, such as a college or a university, that runs its own network. Each of these local ISPs can be connected to a regional or national service provider.

A **transmission medium** can be broadly defined as anything that can carry information from a source to a destination.

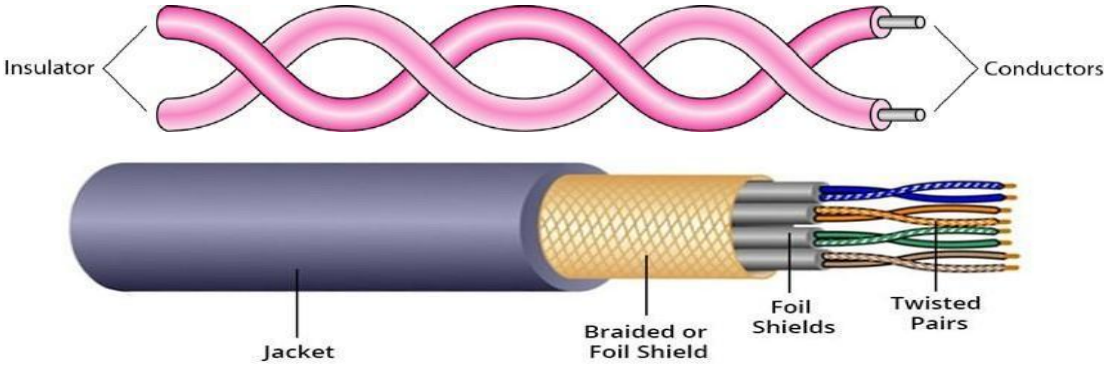


Classes of transmission media



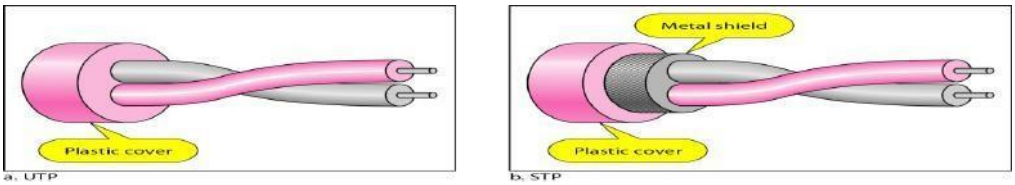
Guided Media: Guided media, which are those that provide a medium from one device to another, include twisted-pair cable, coaxial cable, and fiber-optic cable.

Twisted-Pair Cable: A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together. One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference.



Unshielded Versus Shielded Twisted-Pair Cable

The most common twisted-pair cable used in communications is referred to as unshielded twisted-pair (UTP). STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors. Although metal casing improves the quality of cable by preventing the penetration of noise or crosstalk, it is bulkier and more expensive.



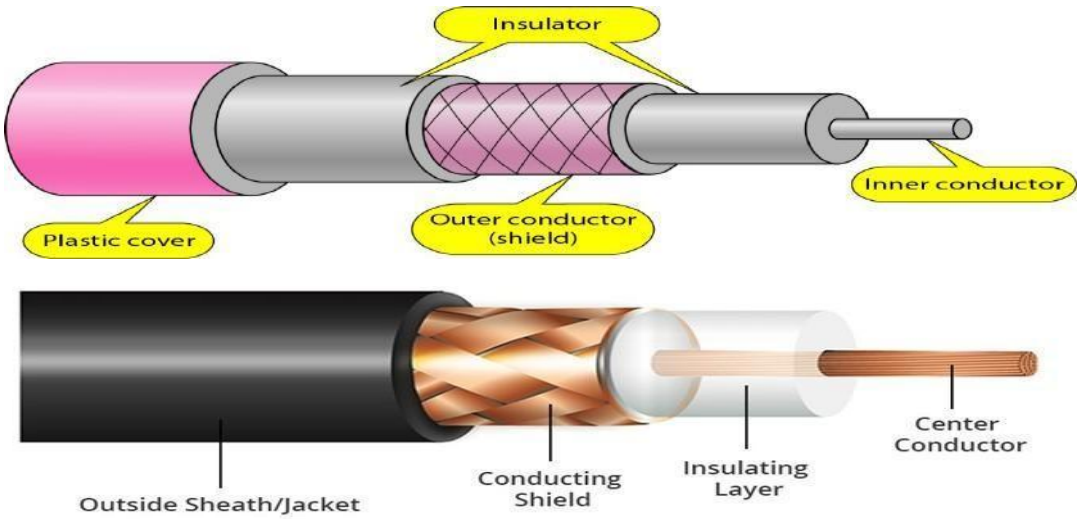
The most common UTP connector is RJ45 (RJ stands for registered jack)

Applications

Twisted-pair cables are used in telephone lines to provide voice and data channels. Local-area networks, such as l0Base-T and l00Base-T, also use twisted-pair cables.

Coaxial Cable

Coaxial cable (or *coax)* carries signals of higher frequency ranges than those in twisted pair cable. coax has a central core conductor of solid or stranded wire (usuallycopper) enclosed in an insulating sheath, which is, in turn, encased in an outer conductor of metal foil, braid, or a combination of the two. The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit.This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected by a plastic cover.



The most common type of connector used today is the Bayone-Neill-Concelman (BNe), connector.

Applications

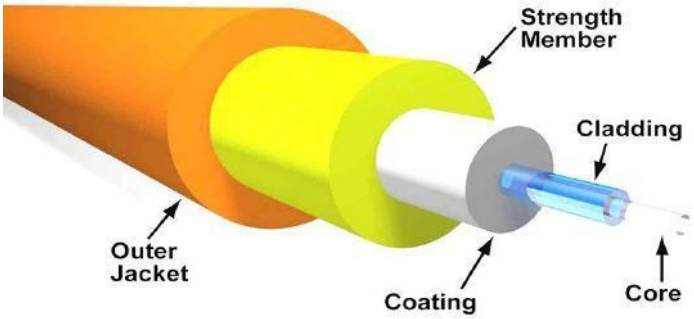
Coaxial cable was widely used in analog telephone networks,digital telephone networks Cable TV networks also use coaxial cables.

Another common application of coaxial cable is in traditional Ethernet LANs

Fiber-Optic Cable

A fiber-optic cable is made of glass or plastic and transmits signals in the form of light. Light travels in a straight line as long as it is moving through a single uniform substance.

Fiber Construction



The **subscriber channel** (SC) **connector,** The **straight-tip** (ST) **connector, MT-RJ(mechanical transfer registered jack)** is a connector

Applications

Fiber-optic cable is often found in backbone networks because its wide bandwidth is cost-effective..

Some cable TV companies use a combination of optical fiber and coaxial cable,thus creating a hybrid network.

Local-area networks such as 100Base-FX network (Fast Ethernet) and 1000Base-X also use fiber-optic cable

Advantages and Disadvantages of Optical Fiber

Advantages Fiber-optic cable has several advantages over metallic cable (twisted pair or coaxial).

1 Higher bandwidth.

2 Less signal attenuation. Fiber-optic transmission distance is significantly greaterthan that of other guided media. A signal can run for 50 km without requiring regeneration. We need repeaters every 5 km for coaxial or twisted- pair cable.

3 Immunity to electromagnetic interference. Electromagnetic noise cannot affect fiber-optic cables.

4 Resistance to corrosive materials. Glass is more resistant to corrosive materials than copper.

5 Light weight. Fiber-optic cables are much lighter than copper cables.

6 Greater immunity to tapping. Fiber-optic cables are more immune to tapping than copper cables. Copper cables create antenna effects that can easily be tapped.

Disadvantages There are some disadvantages in the use of optical fiber. 1Installation and

maintenance

2 Unidirectional light propagation. Propagation of light is unidirectional. If we need bidirectional communication, two fibers are needed.

3 Cost. The cable and the interfaces are relatively more expensive than those of other guided media. If the demand for bandwidth is not high, often the use of optical fiber cannot be justified.

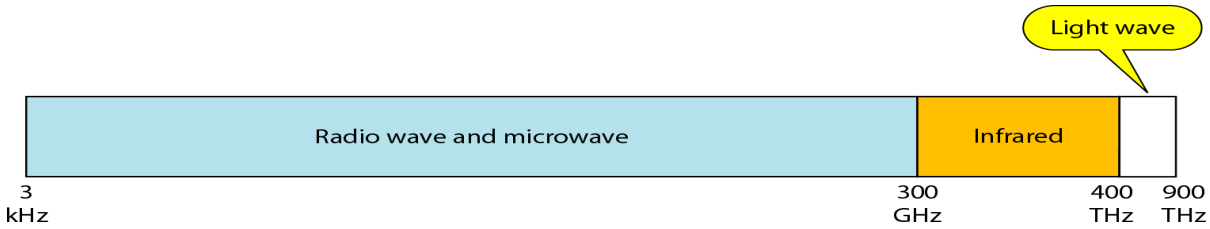
UNGUIDED MEDIA: WIRELESS

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

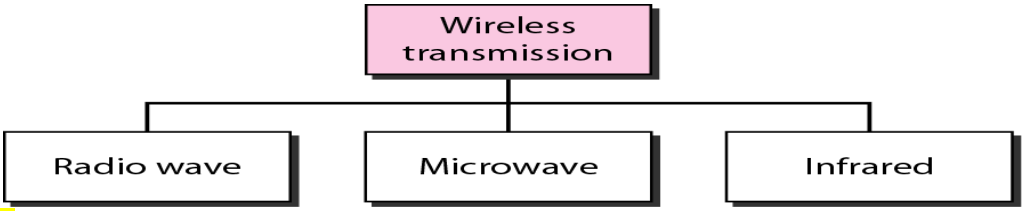
Radio Waves

Microwaves

Infrared



Unguided signals can travel from the source to destination in several ways: ground propagation, sky propagation, and line-of-sight propagation, as shown in Figure

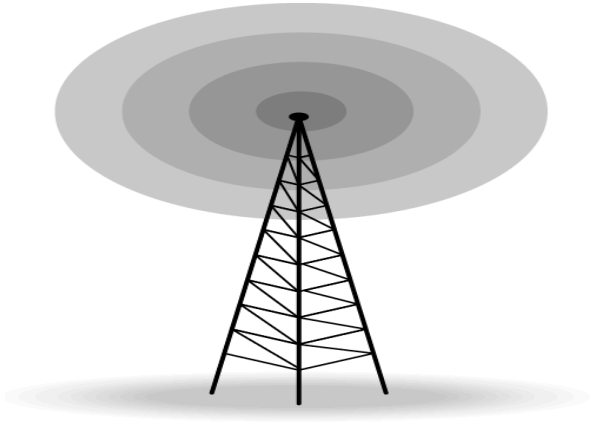


Radio Waves

Electromagnetic waves ranging in frequencies between 3 kHz and 1 GHz are normally called radio waves. Radio waves are omni directional. When an antenna transmits radio waves, they are propagated in all directions. This means that the sending and receiving antennas do not have to be aligned. A sending antenna sends waves that can be received by any receiving antenna. The omni directional property has a disadvantage, too. The radio waves transmitted by one antenna are susceptible to interference by another antenna that may send signals using the same frequency or band.

Omni directional Antenna

Radio waves use omnidirectional antennas that send out signals in all directions. Based on the wavelength, strength, and the purpose of transmission, we can have several types of antennas. Figure shows an omnidirectional antenna.



Applications

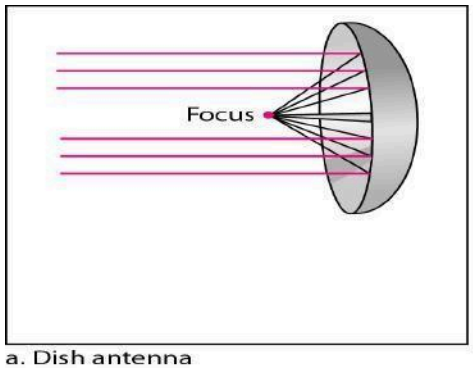
The Omni directional characteristics of radio waves make them useful for multicasting, in which there is one sender but many receivers. AM and FM radio, television, maritime radio, cordless phones, and paging are examples of multicasting.

Microwaves

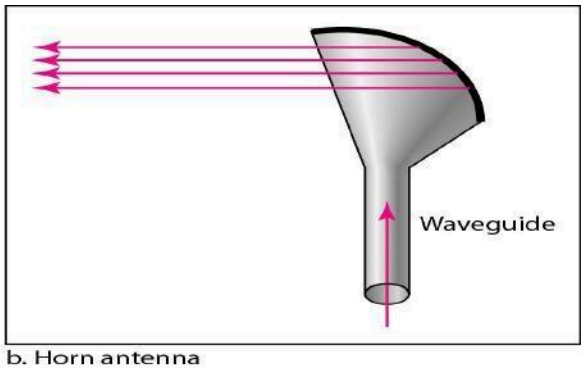
Electromagnetic waves having frequencies between 1 and 300 GHz are called microwaves. Microwaves are unidirectional. The sending and receiving antennas need to be aligned. The unidirectional property has an obvious advantage. A pair of antennas can be aligned without interfering with another pair of aligned antennas

Unidirectional Antenna

Microwaves need unidirectional antennas that send out signals in one direction. Two types of antennas are used for microwave communications: the parabolic dish and the horn



a. Dish antenna



b. Horn antenna

Applications:

Microwaves are used for unicast communication such as cellular telephones, satellite networks, and wireless LANs

Infrared

Infrared waves, with frequencies from 300 GHz to 400 THz (wavelengths from 1 mm to 770 nm), can be used for short-range 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.

When we use our infrared remote control, we do not interfere with the use of the remote by our neighbors. Infrared signals useless for long-range communication. In addition, we cannot use infrared waves outside a building because the sun's rays contain infrared waves that can interfere with the communication.

Applications:

Infrared signals can be used for short-range communication in a closed area using line-of-sight propagation.

UNIT -2 IMPORTANT QUESTIONS (LONG ANSWER) FOR END SEMESTER/FINAL EXAM

1) EXPLAIN DATA LINK LAYER FRAMING METHODS
2) EXPLAIN CRC MECHANISM WITH AN EXAMPLE
3) EXPLAIN PURE ALOHA,SLOTTED ALOHA PROTOCOLS WITH THEIR PERFORMANCE
4) EXPLAIN COLLISON FREE PROTOCOLS
5) EXPLAIN GOBACK N AND SELECTIVE REPEAT PROTOCOLS WITH EXAMPLE
6) EXPLAIN ROUTER,REPEATER,BRIDGE,GATEWAY

UNIT- II

DATA LINK LAYER FUNCTIONS (SERVICES)

1. **Providing services to the network layer:**

   1 Unacknowledged connectionless service.

   Appropriate for low error rate and real-time traffic. Ex: Ethernet

   2. Acknowledged connectionless service.

   Useful in unreliable channels, WiFi. Ack/Timer/Resend

   3. Acknowledged connection-oriented service.

   Guarantee frames are received exactly once and in the right order. Appropriate over long, unreliable links such as a satellite channel or a long- distance telephone circuit
2. **Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.
3. **Physical Addressing**: The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.
4. **Flow Control:** A receiving node can receive the frames at a faster rate than it can process the frame. Without flow control, the receiver's buffer can overflow, and frames can get lost. To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link. This prevents traffic jam at the receiver side.
5. **Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

   Error detection: Errors can be introduced by signal attenuation and noise. Data Link Layer protocol provides a mechanism to detect one or more errors. This is achieved by adding error detection bits in the frame and then receiving node can perform an error check.

   Error correction: Error correction is similar to the Error detection, except that receiving node not only detects the errors but also determine where the errors have occurred in the frame.
6. **Access Control**: Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link**.**
7. **Reliable delivery**: Data Link Layer provides a reliable delivery service, i.e., transmits the network layer datagram without any error. A reliable delivery service is accomplished with transmissions and acknowledgements. A data link layer

mainly provides the reliable delivery

service over the links as they have higher error rates and they can be corrected locally, link at which an error occurs rather than forcing to retransmit the data.

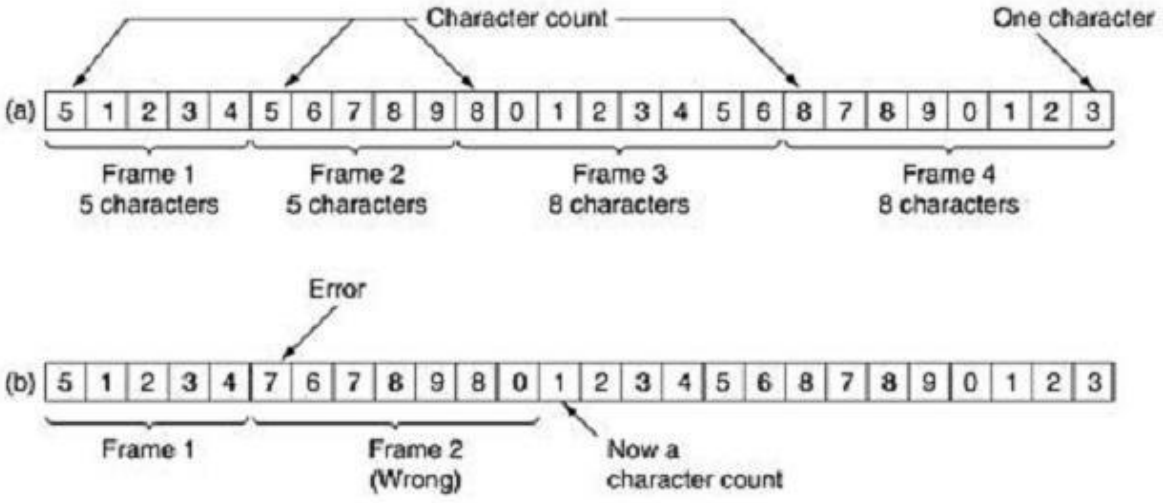
8. **Half-Duplex & Full-Duplex**: In a Full-Duplex mode, both the nodes can transmit the data at the same time. In a Half-Duplex mode, only one node can transmit the data at the same time.

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. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to **detect and, if necessary, correct errors**. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame **(framing**). When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report).We will look at four framing methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

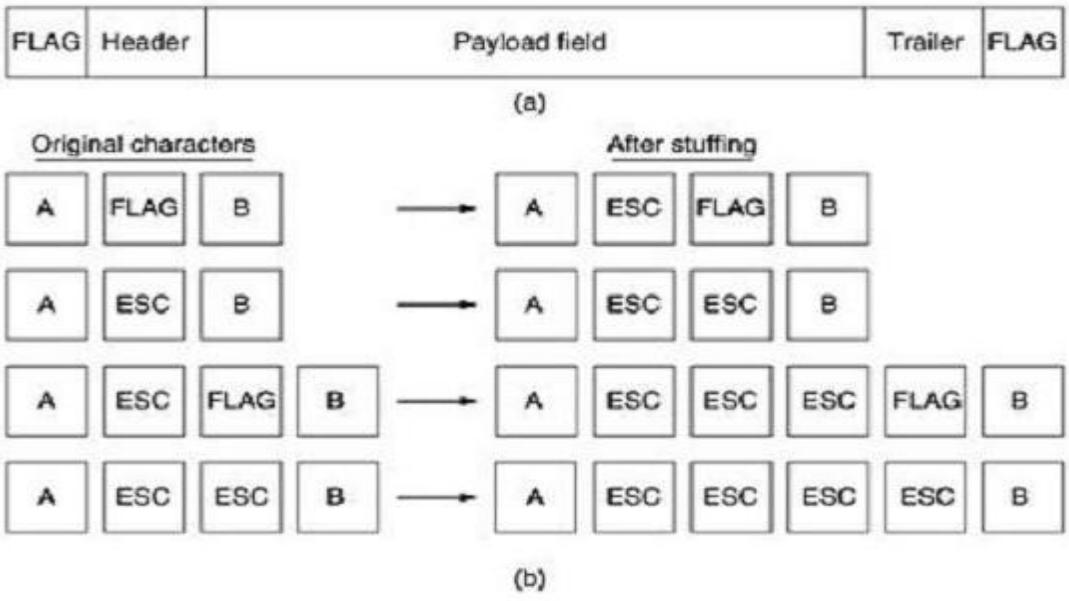
Character count method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig. (a) For four frames of sizes 5, 5, 8, and 8 characters, respectively.



A character stream. (a) Without errors. (b) With one error

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the character count of 5 in the second frame of Fig. (b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

Flag bytes with byte stuffing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in Fig. (a) as FLAG. In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.



(a) A frame delimited by flag bytes (b) Four examples of byte sequences before and after byte stuffing

It may easily happen that the flag byte's bit pattern occurs in the data. This situation will usually interfere with the framing. 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. The data link layer on the receiving end removes the escape byte before the data are given to the network layer. This technique is called byte stuffing or character stuffing.

Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

What happens if an escape byte occurs in the middle of the data? The answer is that, it too is stuffed with an escape byte. Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data. Some examples are shown in Fig. (b). In all cases, the byte sequence delivered after de stuffing is exactly the same as the original byte sequence.

A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters. Not all character codes use 8-bit characters. For example UNICODE uses 16-bit characters, so a new technique had to be developed to allow arbitrary sized characters

Starting and ending flags, with bit stuffing allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte). 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. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de- stuffs (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.

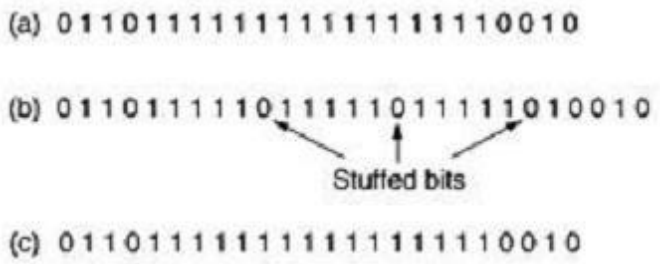
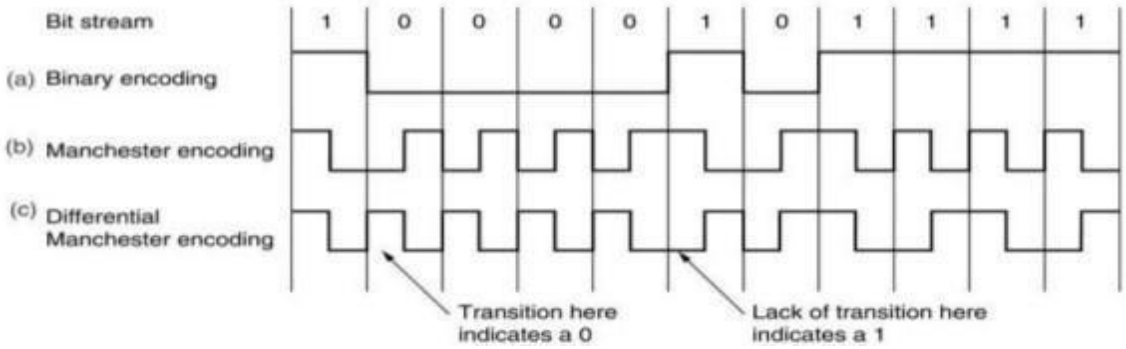


Fig:Bit stuffing. (a) The original data. (b) The data as they appear on the line.
(c) The data as they are stored in the receiver's memory after destuffing.

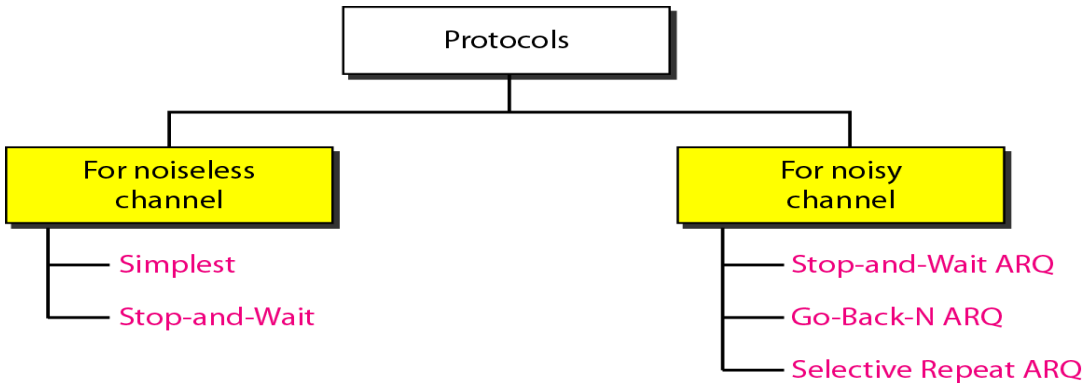
With bit stuffing, the boundary between two frames can be unambiguously recognized by the flag pattern. Thus, if the receiver loses track of where it is, all it has to do is scan the input for flag sequences, since they can only occur at frame boundaries and never within the data.

Physical layer coding violations method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-

high and low-low are not used for data but are used for delimiting frames in some protocols.

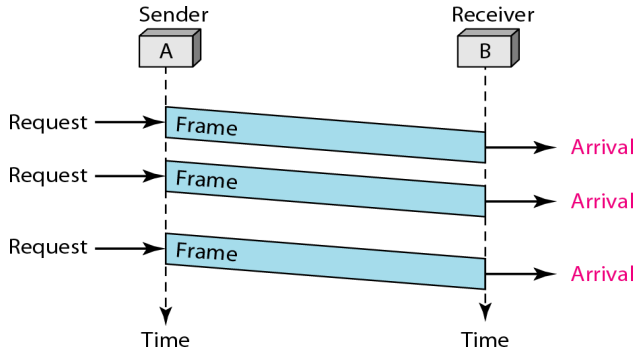


As a final note on framing, many data link protocols use combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter



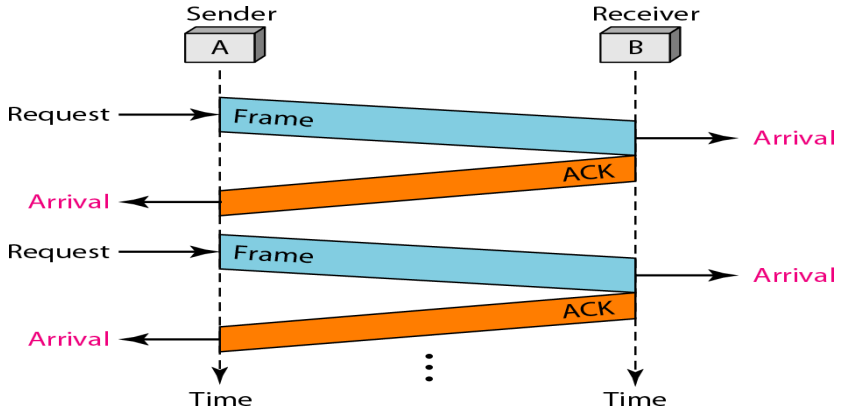
ELEMENTARY DATA LINK PROTOCOLS

Simplest Protocol



It is very simple. The sender sends a sequence of frames without even thinking about the receiver. Data are transmitted in one direction only. Both sender & receiver always ready. Processing time can be ignored. Infinite buffer space is available. And best of all, the communication channel between the data link layers never damages or loses frames. This thoroughly unrealistic protocol, which we will nickname "Utopia," .The utopia protocol is unrealistic because **it does not handle either flow control or error correction**

Stop-and-wait Protocol



It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame

It is Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens of acknowledgment) travel from the other direction. We add flow control to our previous protocol.

NOISY CHANNELS

Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor, noiseless channels are nonexistent. We can ignore the error (as we sometimes do), or we need to add error control to our protocols. We discuss three protocols in this section that use error control.

Sliding Window Protocols:

1 Stop-and-Wait Automatic Repeat Request

2 Go-Back-N Automatic Repeat Request

3 **Selective Repeat Automatic Repeat Request**

1 Stop-and-Wait Automatic Repeat Request

To detect and correct corrupted frames, we need to add redundancy bits to our data frame. When the frame arrives at the receiver site, it is checked and if it is corrupted, it is silently discarded. The detection of errors in this protocol is manifested by the silence of the receiver.

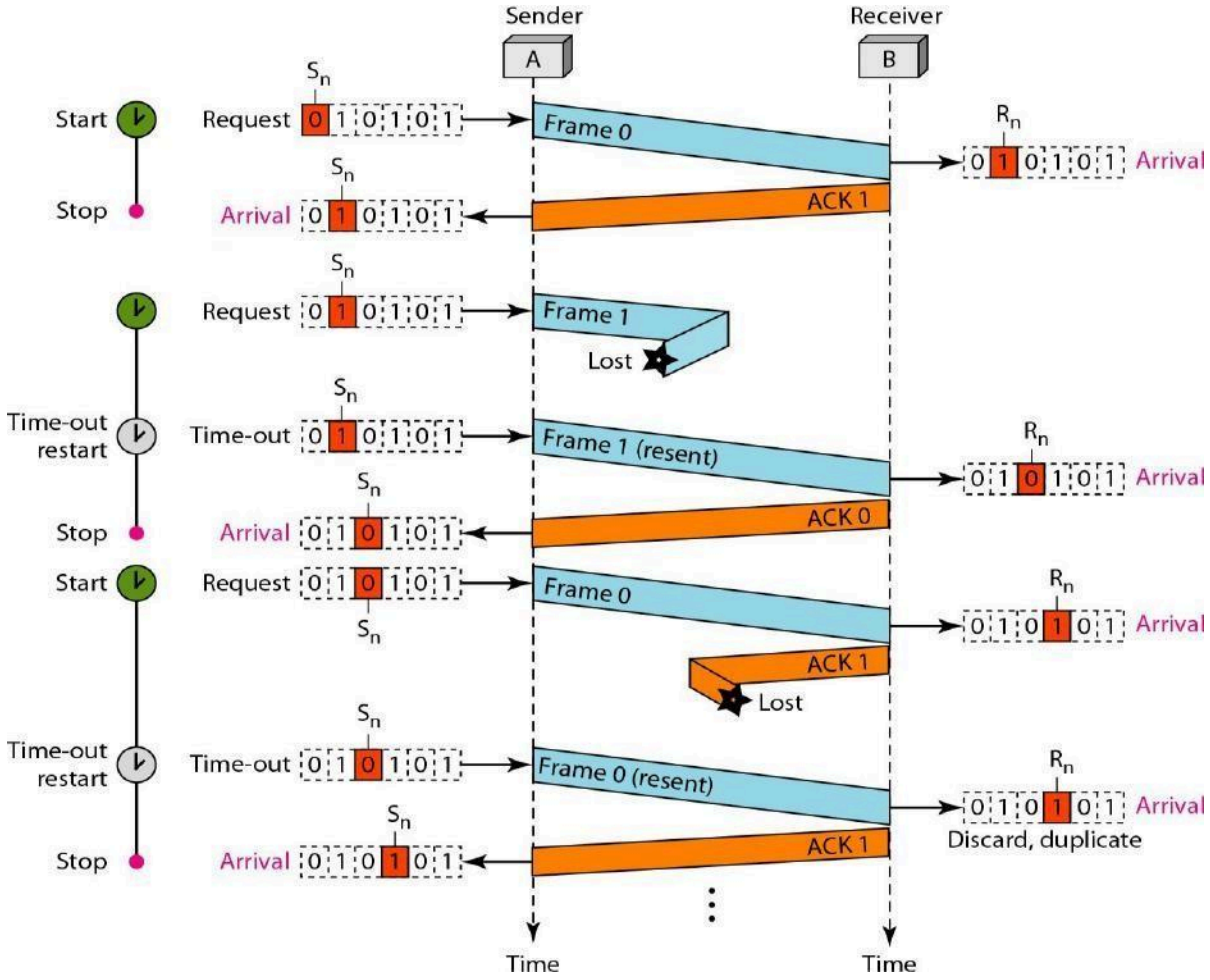
Lost frames are more difficult to handle than corrupted ones. In our previous protocols, there was no way to identify a frame. The received frame could be the correct one, or a duplicate, or a frame out of order. The solution is to number the frames. When the receiver receives a data frame that is out of order, this means that frames were either lost or duplicated

The lost frames need to be resent in this protocol. If the receiver does not respond when there is an error, how can the sender know which frame to resend? To remedy this problem, the sender keeps a copy of the sent frame. At the same time, it starts a timer. If the timer expires and there is no ACK for the sent frame, the frame is resent, the copy is held, and the timer is restarted. Since the protocol uses the stop-and-wait mechanism, there is only one specific frame that needs an ACK

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

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

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

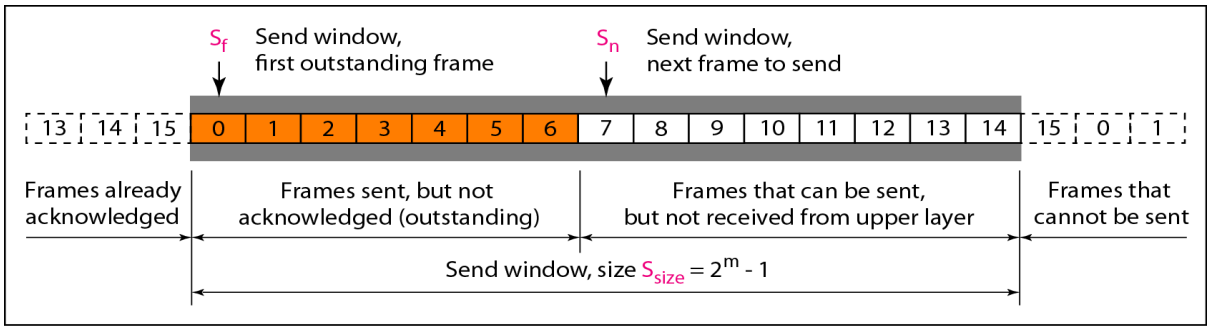


2. Go-Back-N Automatic Repeat Request

To improve the efficiency of transmission (filling the pipe), multiple frames must be in transition while waiting for acknowledgment. In other words, we need to let more than one frame be outstanding to keep the channel busy while the sender is waiting for acknowledgment.

The first is called Go-Back-N Automatic Repeat. In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledgments arrive.

In the Go-Back-N Protocol, the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits. The sequence numbers range from 0 to 2 *power m*- 1. For example, if *m* is 4, the only sequence numbers are 0 through 15 inclusive.



a. Send window before sliding

The **sender window** at any time divides the possible sequence numbers into four regions.

The first region, from the far left to the left wall of the window, defines the sequence numbers belonging to frames that are already acknowledged. The sender does not worry about these frames and keeps no copies of them.

The second region, colored in Figure (a), defines the range of sequence numbers belonging to the frames that are sent and have an unknown status. The sender needs to wait to find out if these frames have been received or were lost. We call these outstanding frames.

The third range, white in the figure, defines the range of sequence numbers for frames that can be sent; however, the corresponding data packets have not yet been received from the network layer.

Finally, the fourth region defines sequence numbers that cannot be used until the window slides

The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: S_f, S_n, and S_{size}. The variable *Sf* defines the sequence number of the first (oldest) outstanding frame. The variable *Sn* holds the sequence number that will be assigned to the next frame to be sent. Finally, the variable Ssize defines the size of the window.

Figure (b) shows how a send window can slide one or more slots to the right when an acknowledgment arrives from the other end. The acknowledgments in this protocol are cumulative, meaning that more than one frame can be acknowledged by an ACK frame. In Figure, frames 0, I, and 2 are

acknowledged, so the window has slide to the right three slots. Note that the value of *Sf* is 3 because frame 3 is now the first outstanding frame.**The send window can slide one or more slots when a valid acknowledgment arrives.**

Receiver window: variable *Rn* (receive window, next frame expected) .
The sequence numbers to the left of the window belong to the frames already received and acknowledged; the sequence numbers to the right of this window define the frames that cannot be received. Any received frame with a sequence number in these two regions is discarded. Only a frame with a sequence number matching the value of *Rn* is accepted and acknowledged. The receive window also slides, but only one slot at a time. When a correct frame is received (and a frame is received only one at a time), the window slides.(see below figure for receiving window)

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

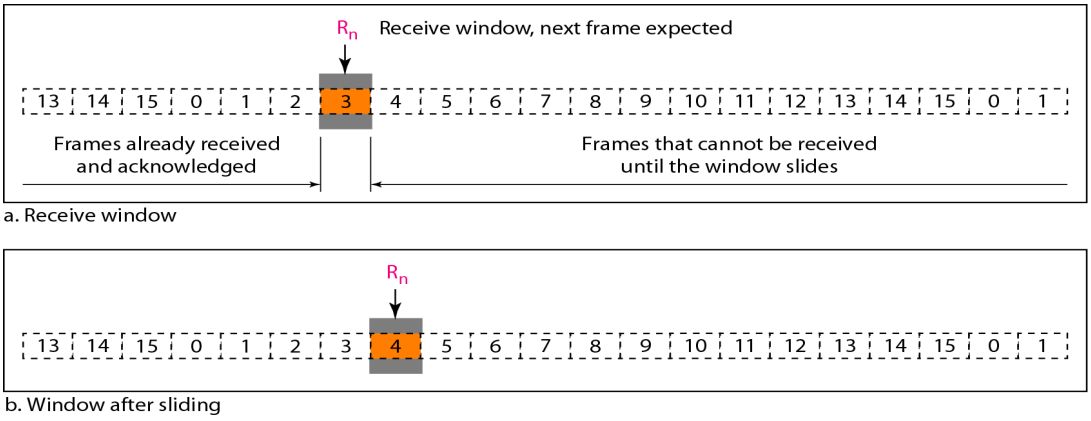


Fig: Receiver window (before sliding (a), After sliding (b))

Timers

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

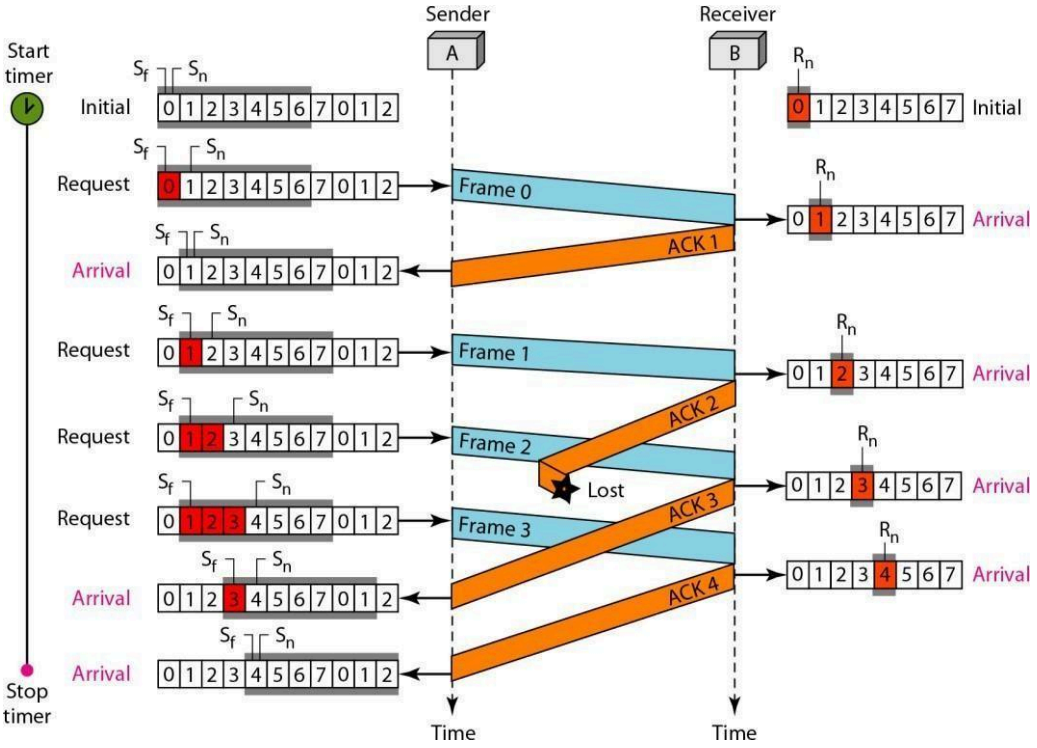
Acknowledgment

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

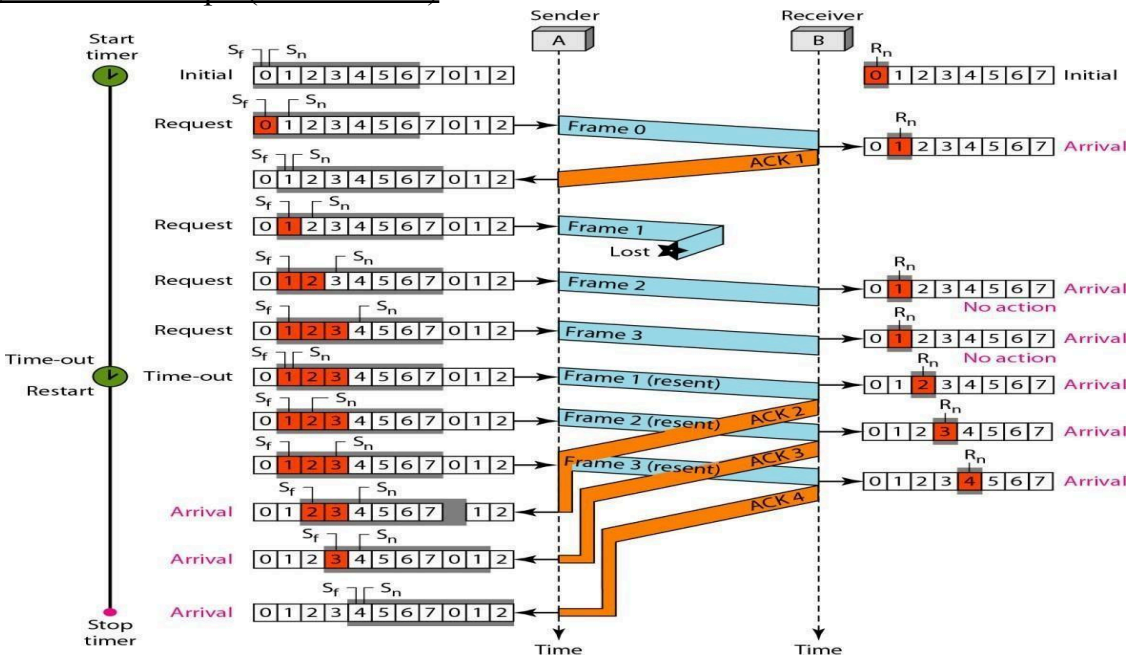
Resending a Frame

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

Below figure is an example(if ack lost) of a case where the forward channel is reliable, but the reverse is not. No data frames are lost, but some ACKs are delayed and one is lost. The example also shows how cumulative acknowledgments can help if acknowledgments are delayed or lost



Below figure is an example(if frame lost)



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

3 Selective Repeat Automatic Repeat Request

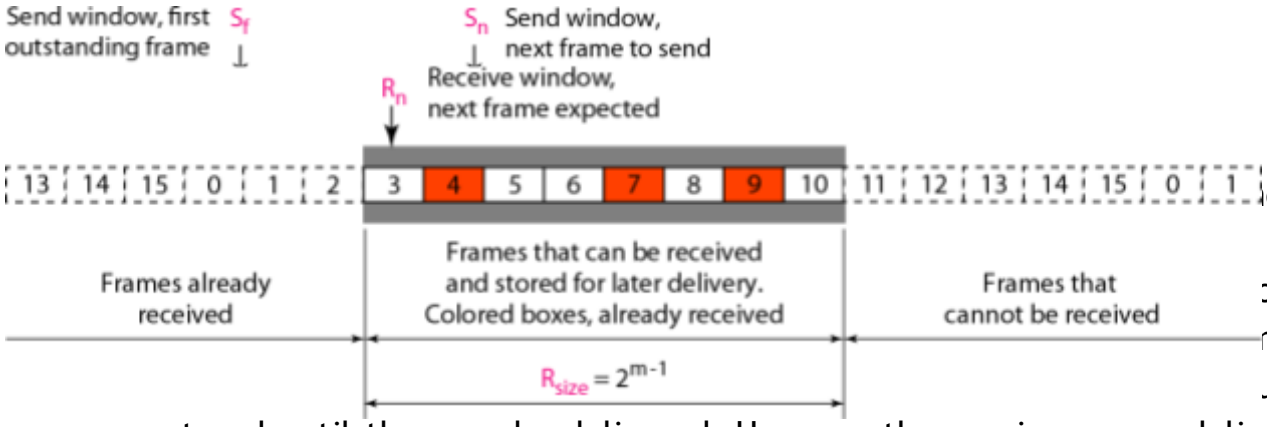
In Go-Back-N ARQ, The receiver keeps track of only one variable, and there is no need to buffer out-of- order frames; they are simply discarded. However, this protocol is very inefficient for a noisy link.

In a noisy link a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission.

For noisy links, there is another mechanism that does not resend *N* frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called Selective Repeat ARQ.

It is more efficient for noisy links, but the processing at the receiver is more complex.

Sender Window (explain go-back N sender window concept (before & after sliding.) The only difference in sender window between Go-back N and Selective Repeat is Window size)

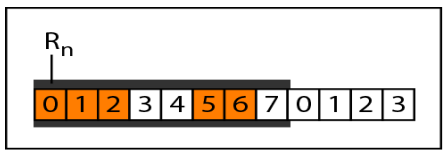


one in Go Back-N.
d window *(2^{m-1})*.
of the receiver
order frames to be
ndow and receive
ut of order and be

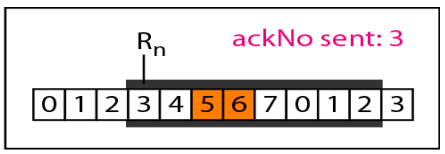
stored until they can be delivered. However the receiver never delivers packets out of order to the network layer. Above Figure shows the receive window. Those slots inside the window that are colored define frames that have arrived out of order and are waiting for their neighbors to arrive before delivery to the network layer.

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

Delivery of Data in Selective Repeat ARQ:

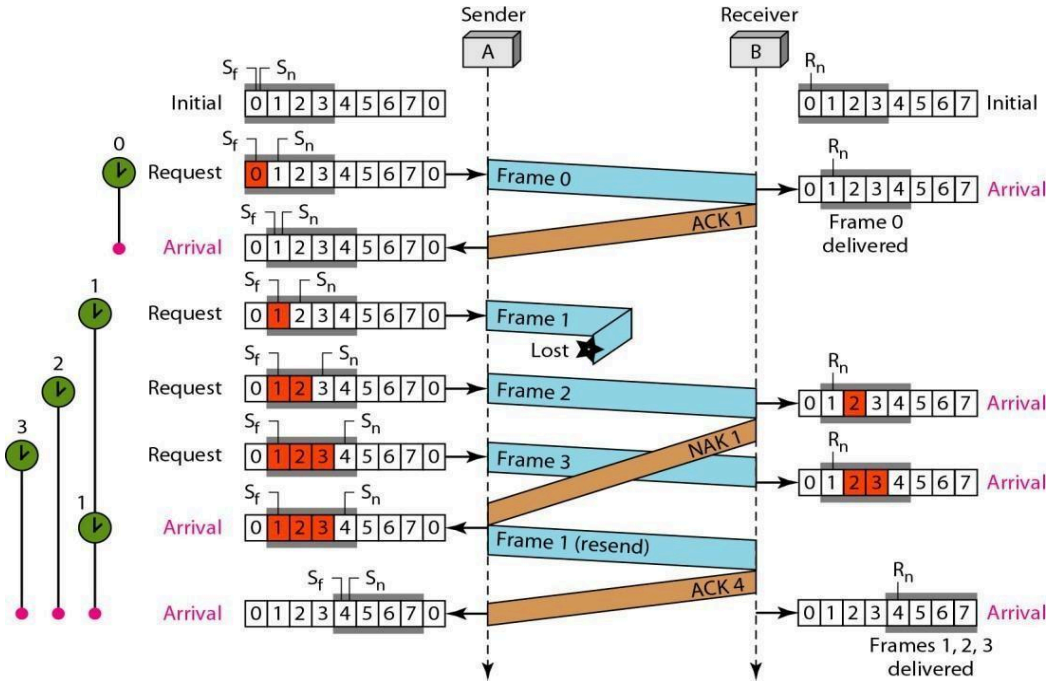


a. Before delivery



b. After delivery

Flow Diagram



Differences between Go-Back N & Selective Repeat

One main difference is the number of timers. Here, each frame sent or resent needs a timer, which means that the timers need to be numbered (0, 1,2, and 3). The timer for frame 0 starts at the first request, but stops when the ACK for this frame arrives.

There are two conditions for the delivery of frames to the network layer: First, a set of consecutive frames must have arrived. Second, the set starts from the beginning of the window. After the first arrival, there was only one frame and it started from the beginning of the window. After the last arrival, there are three frames and the first one starts from the beginning of the window.

Another important point is that a NAK is sent.

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

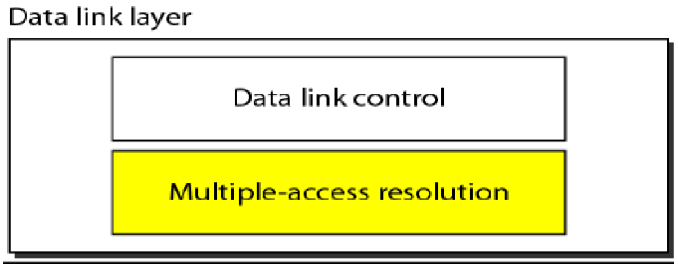
Piggybacking

A technique called **piggybacking** is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

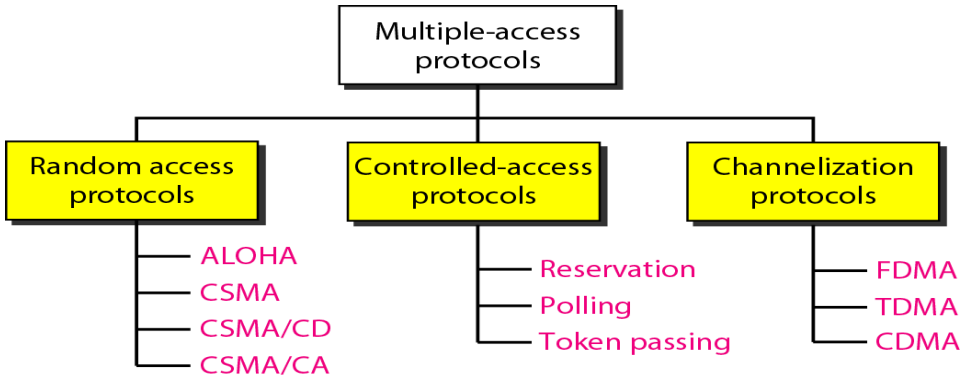
RANDOM ACCESS PROTOCOLS

We can consider the data link layer as two sub layers. The upper sub layer is responsible for data link control, and the lower sub layer is responsible for resolving access to the

shared media



The upper sub layer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sub layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer. When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link.



Taxonomy of multiple-access protocols

RANDOM ACCESS

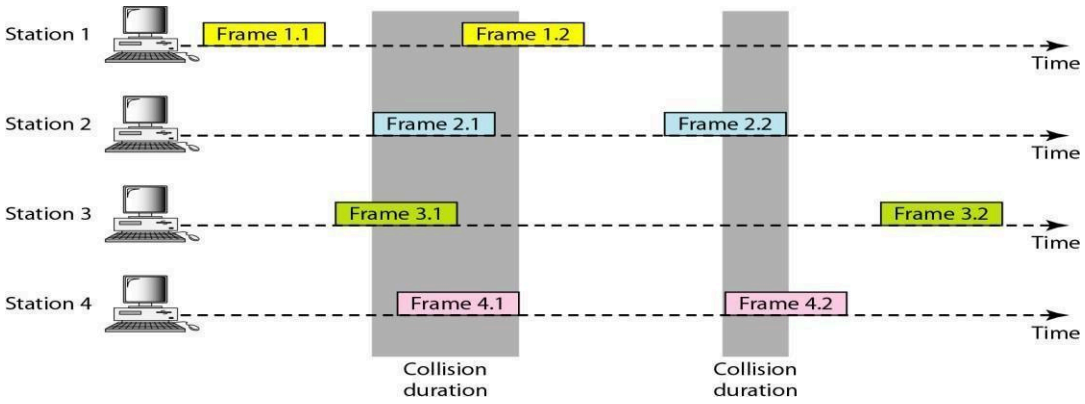
In random access or contention methods, no station is superior to another station and none is assigned the control over another.

Two features give this method its name. First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access*. Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.

ALOHA

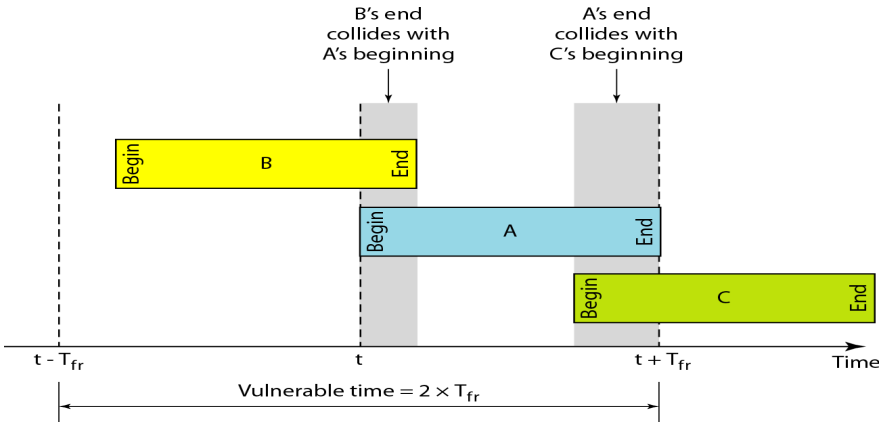
1 *Pure ALOHA*

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



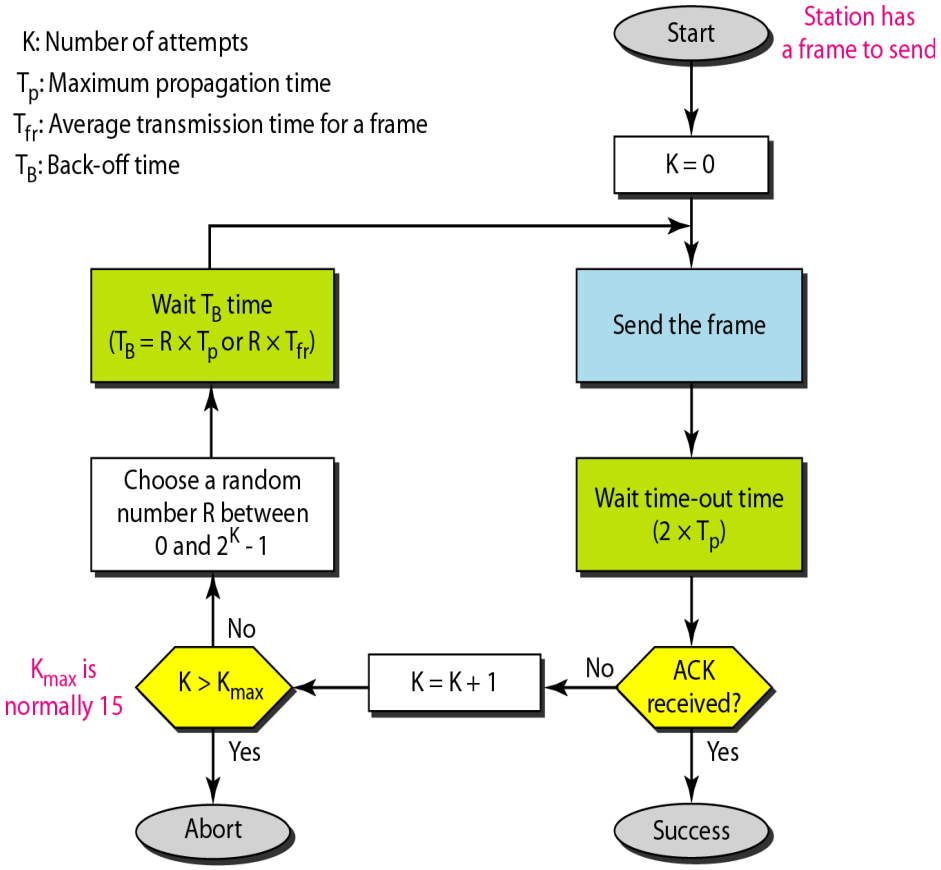
Frames in a pure ALOHA network

In pure ALOHA, the stations transmit frames whenever they have data to send.
When two or more stations transmit simultaneously, there is collision and the frames are destroyed.
In pure ALOHA, whenever any station transmits a frame, it expects the acknowledgement from the receiver.
If acknowledgement is not received within specified time, the station assumes that the frame (or acknowledgement) has been destroyed.
If the frame is destroyed because of collision the station waits for a random amount of time and sends it again. This waiting time must be random otherwise same frames will collide again and again.
Therefore pure ALOHA dictates that when time-out period passes, each station must wait for a random amount of time before resending its frame. This randomness will help avoid more collisions.
Vulnerable time Let us find the length of time, the **vulnerable time,** in which there is a possibility of collision. We assume that the stations send fixed- length frames with each frame taking *Tfr* S to send. Below Figure shows the vulnerable time for station A.



Station A sends a frame at time *t*. Now imagine station B has already sent a frame between *t* - *T*fr and *t*. This leads to a collision between the frames from station A and station B. The end of B's frame collides with the beginning of A's frame. On the other hand, suppose that station C sends a frame between *t* and *t* + *Tfr* . Here, there is a collision between frames from station A and station C. The beginning of C's frame collides with the end of A's frame

Looking at Figure, we see that the vulnerable time, during which a collision may occur in pure ALOHA, is 2 times the frame transmission time. Pure ALOHA vulnerable time = 2 x Tfr



Procedure for pure ALOHA protocol

2 *Slotted ALOHA*

Pure ALOHA has a vulnerable time of 2 x *Tfr* . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA.

In slotted ALOHA we divide the time into slots of Tfr s and force the station to send only at the beginning of the time slot. Figure 3 shows an example of frame collisions in slotted ALOHA

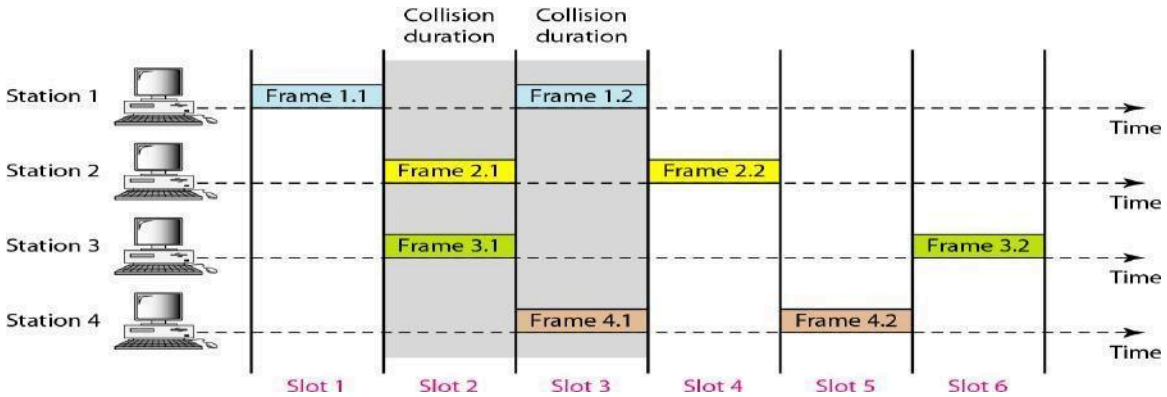
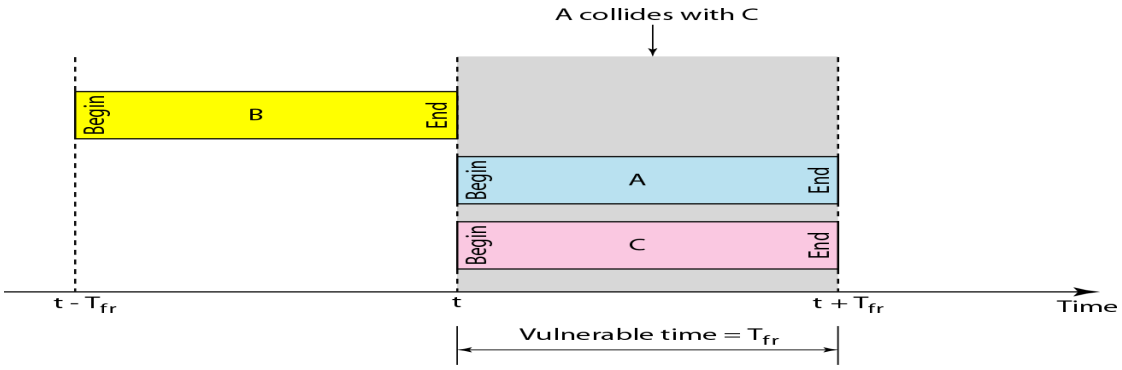


FIG:3

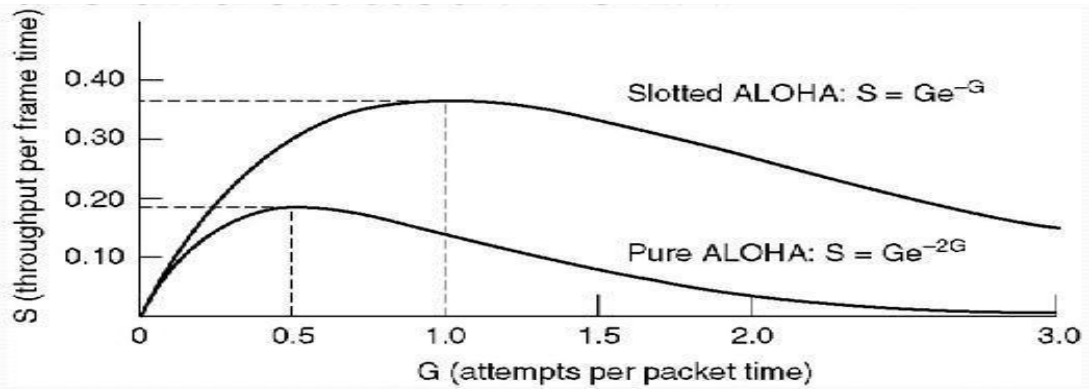
Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to *Tfr* Figure 4 shows the situation

Below fig shows that the vulnerable time for slotted ALOHA is one-half that of pure ALOHA. Slotted ALOHA vulnerable time = Tfr



The throughput for slotted ALOHA is S = G × e−G . *The maximum throughput Smax = 0.368 when G = 1.*

Comparison between Pure Aloha & Slotted Aloha

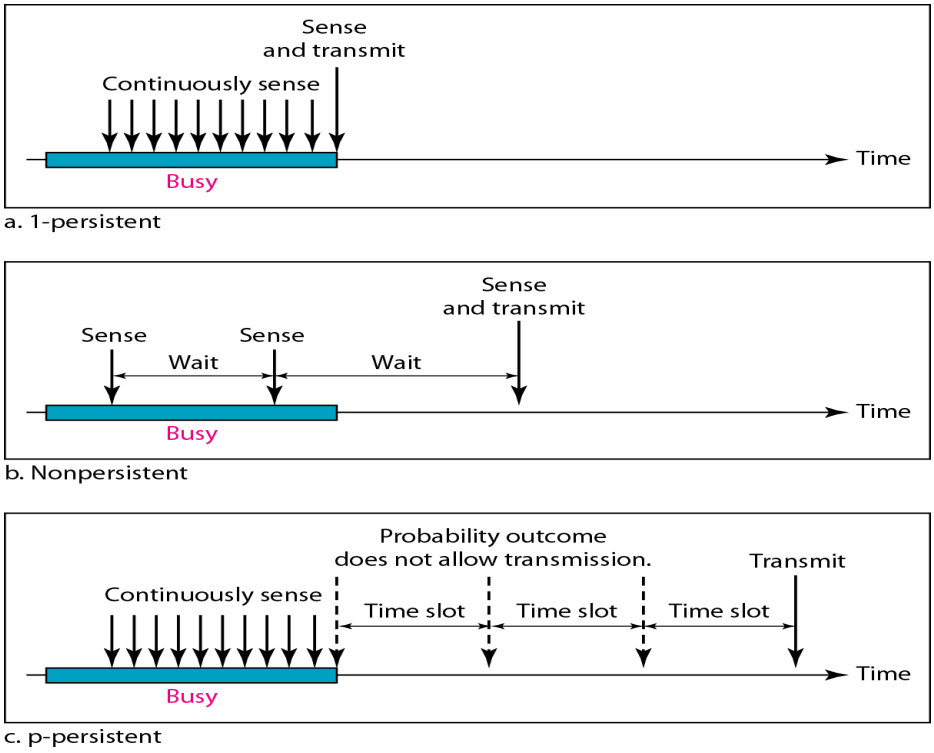


Carrier Sense Multiple Access (CSMA)

To minimize the chance of collision and, therefore, increase the performance, the CSMA method was developed. The chance of collision can be reduced if a station senses the medium before trying to use it. Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending. In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."

Persistence Methods

What should a station do if the channel is busy? What should a station do if the channel is idle? Three methods have been devised to answer these questions: the 1-persistent method, the non-persistent method, and the p-persistent method



1-Persistent: In this method, after the station finds the line idle, it sends its frame immediately (with probability 1). This method has the highest chance of collision because two or more stations may find the line idle and send their frames immediately.

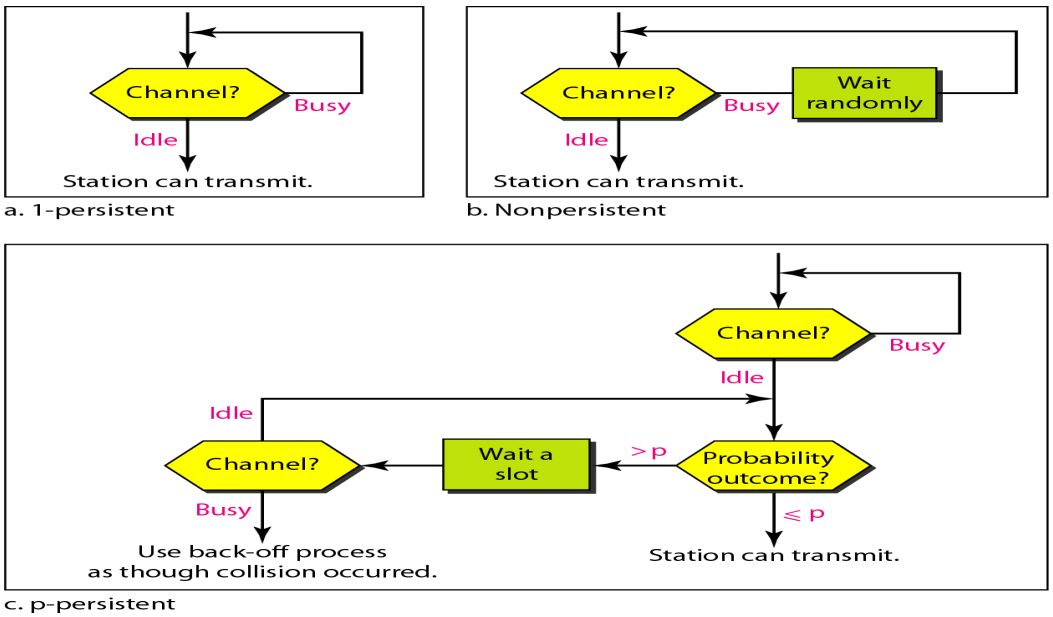
Non-persistent: a station that has a frame to send senses the line. If the line is idle, it sends immediately. If the line is not idle, it waits a random amount of time and then senses the line again. This approach reduces the chance of collision because it is unlikely that two or more stations will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

p-Persistent: This is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time. The p-persistent approach combines the advantages of the other two strategies. It reduces the chance of collision and improves

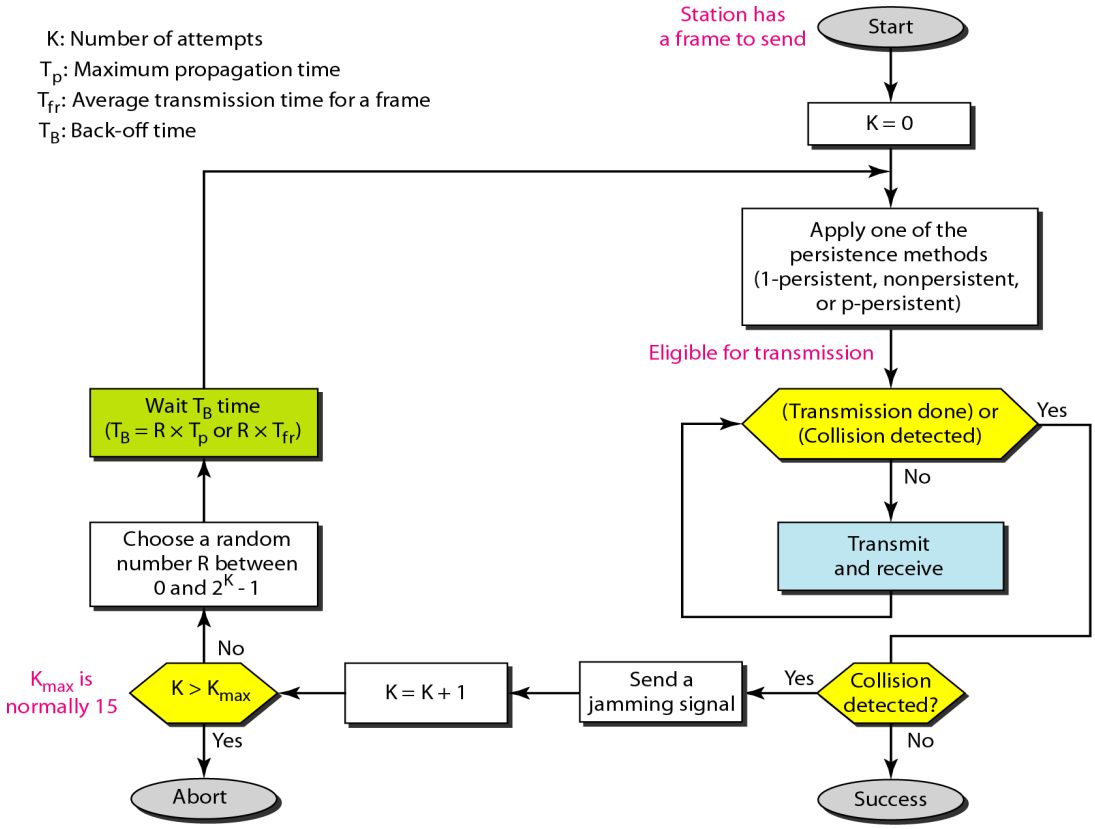
efficiency.

In this method, after the station finds the line idle it follows these steps:

1. With probability *p*, the station sends its frame.
2. With probability *q* = 1 - *p*, the station waits for the beginning of the next time slot and checks the line again.
   a. If the line is idle, it goes to step 1.
   b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.



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



Flow diagram for the CSMA/CD

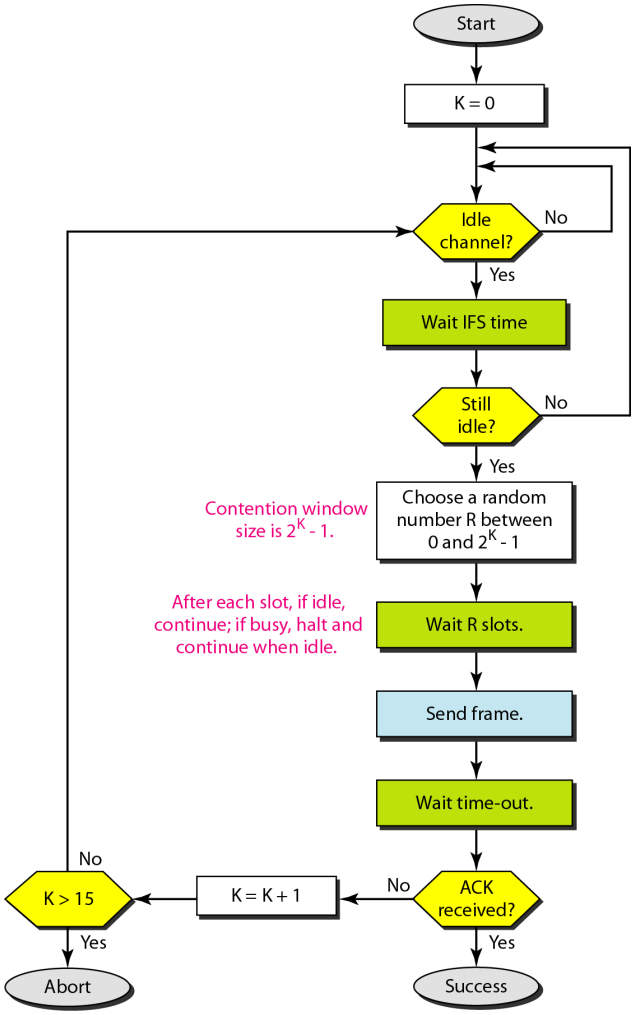
DIFFERENCES BETWEEN ALOHA & CSMA/CD

The first difference is the addition of the persistence process. We need to sense the channel before we start sending the frame by using one of the persistence processes

The second difference is the frame transmission. In ALOHA, we first transmit the entire frame and then wait for an acknowledgment. In *CSMA/CD*, transmission and collision detection is a continuous process. We do not send the entire frame and then look for a collision. The station transmits and receives continuously and simultaneously

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

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



This is the CSMA protocol with collision avoidance.

- The station ready to transmit, senses the line by using one of the persistent strategies.
- As soon as it finds the line to be idle, the station waits for an IFS (Inter frame space) amount of time.
- If then waits for some random time and sends the frame.
- After sending the frame, it sets a timer and waits for the acknowledgement from the receiver.
- If the acknowledgement is received before expiry of the timer, then the transmission is successful.
- But if the transmitting station does not receive the expected acknowledgement before the timer expiry then it increments the back off

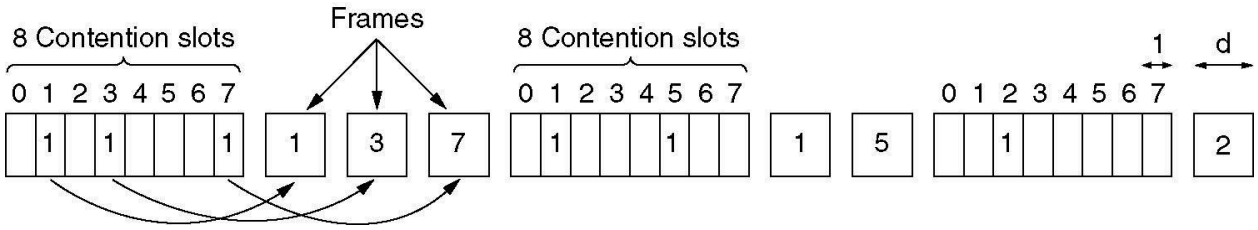
parameter, waits for the back off time and re senses the line

Collision-Free Protocols

- we assume that there are exactly *N* stations, each with a unique address from 0 to *N* - 1 "wired" into it.
- It does not matter that some stations may be inactive part of the time. We also assume that propagation delay is negligible.
- the **basic bit-map method**, each contention period consists of exactly *N* slots. If station 0 has a frame to send, it transmits a 1 bit during the zeroth slot. No other station is allowed to transmit during this slot. Regardless of what station 0 does,
- station 1 gets the opportunity to transmit a 1 during slot 1, but only if it has a frame queued. In general, station *j* may announce that it has a frame to send by inserting a 1 bit into slot *j*.
- After all *N* slots have passed by, each station has complete knowledge of which stations wish to transmit. At that point, they begin transmitting in numerical order .

a) Since everyone agrees on who goes next, there will never be any collisions.

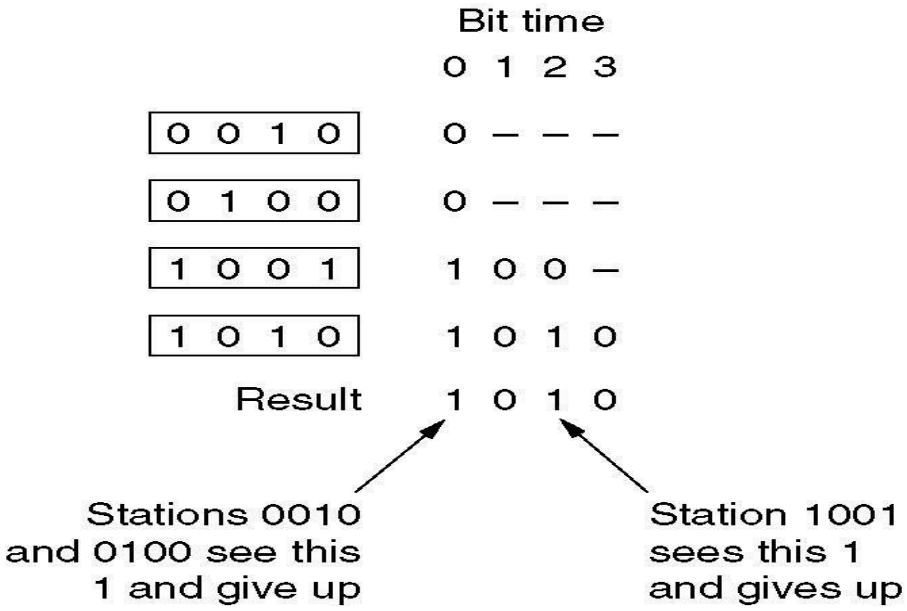
The basic bit-map protocol.



- If a station becomes ready just after its bit slot has passed by, it is out of luck and must remain silent until every station has had a chance and the bit map has come around again.
- Protocols like this in which the desire to transmit is broadcast before the actual transmission are called reservation protocols.

The binary countdown protocol

- A problem with the basic bit-map protocol is that the overhead is 1 bit per station, so it does not scale well to networks with thousands of stations.
- A station wanting to use the channel now broadcasts its address as a binary bit string, starting with the high-order bit. All addresses are assumed to be the same length. The bits in each address position from different stations are BOOLEAN ORed together. We will call this protocol **binary countdown**.
- To avoid conflicts, an arbitration rule must be applied: as soon as a station sees that a high-order bit position that is 0 in its address has been overwritten with a 1, it gives up.
- For example, if stations 0010, 0100, 1001, and 1010 are all trying to get the channel, in the first bit time the stations transmit 0, 0, 1, and 1, respectively.
- These are ORed together to form a 1. Stations 0010 and 0100 see the 1 and know that a higher-numbered station is competing for the channel, so they give up for the current round. Stations 1001 and 1010 continue.
- The next bit is 0, and both stations continue. The next bit is 1, so station 1001 gives up. The winner is station 1010 because it has the highest address. After winning the bidding, it may now transmit a frame, after which another bidding cycle starts.



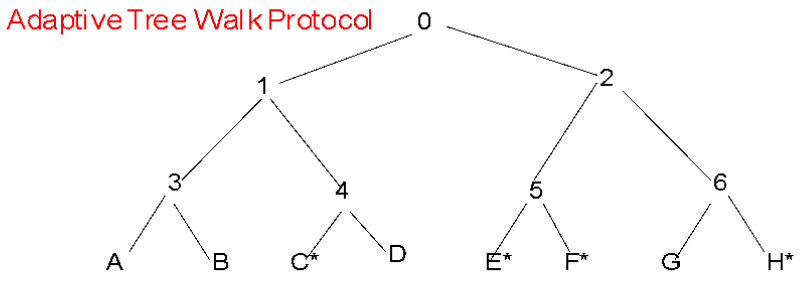
- ***A dash indicates silence.***
- It has the property that higher-numbered stations have a higher priority than lower-numbered stations, which may be either good or bad, depending on the context.

Limited-Contention Protocols

- collision based protocols (ALOHA,CSMA/CD) are good when the network load is low.
  - collision free protocols (bit map, binary countdown) are good when load is high.
  - How about combining their advantages -- limited contention protocols.
    - Behave like the ALOHA scheme under light load
    - Behave like the bitmap scheme under heavy load.

Adaptive tree walk protocol
trick: partition the group of stations and limit the contention for each slot.

- under light load, every one can try for each slot like aloha
- under heavy load, only a small group can try for each slot
- how do we do it
  - » treat stations as the leaf of a binary tree.
  - » first slot (after successful transmission), all stations (under the root node) can try to get the slot.
  - » if no conflict, fine.
  - » if conflict, only nodes under a subtree get to try for the next one. (depth first search)



The tree for eight stations. And 4-stations are participating.

- Slot 0: C*, E*, F*, H* (all nodes under node 0 can try), conflict
- slot 1: C* (all nodes under node 1 can try), C sends
- slot 2: E*, F*, H*(all nodes under node 2 can try), conflict
- slot 3: E*, F* (all nodes under node 5 can try), conflict
- slot 4: E* (all nodes under E can try), E sends
- slot 5: F* (all nodes under F can try), F sends
- slot 6: H* (all nodes under node 6 can try), H sends.

Error Detection

Error

A condition when the receiver's information does not matches with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0. **Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model)** Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

Basic approach used for error detection is the use of redundancy bits, where

additional bits are added to facilitate detection of errors. Some popular techniques for error detection are:

1. Simple Parity check

2. Two-dimensional Parity check

3. Checksum

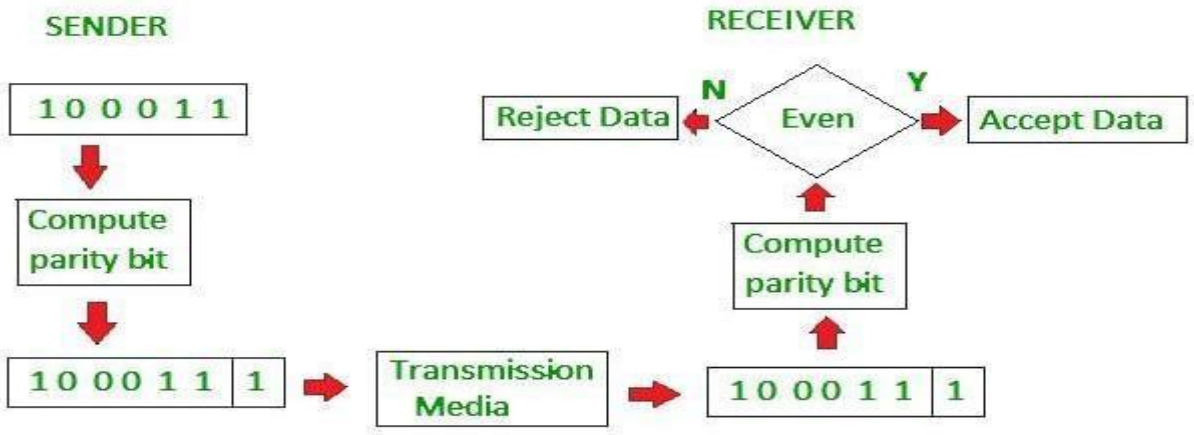
4. Cyclic redundancy check

Simple Parity check

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of : 1 is added to the block if it contains odd number of 1's, and

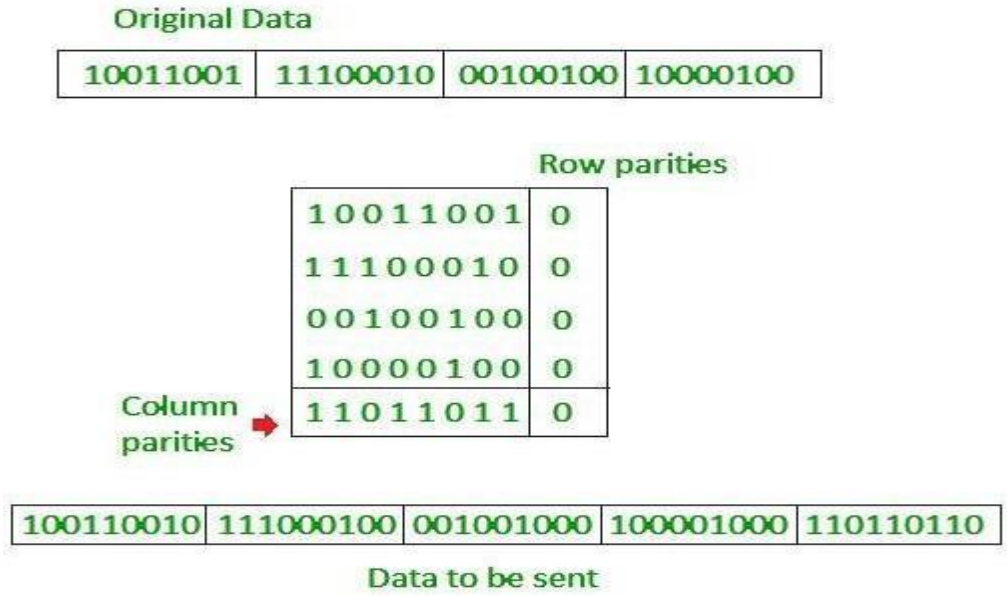
0 is added if it contains even number of 1's

This scheme makes the total number of 1's even, that is why it is called even parity checking.



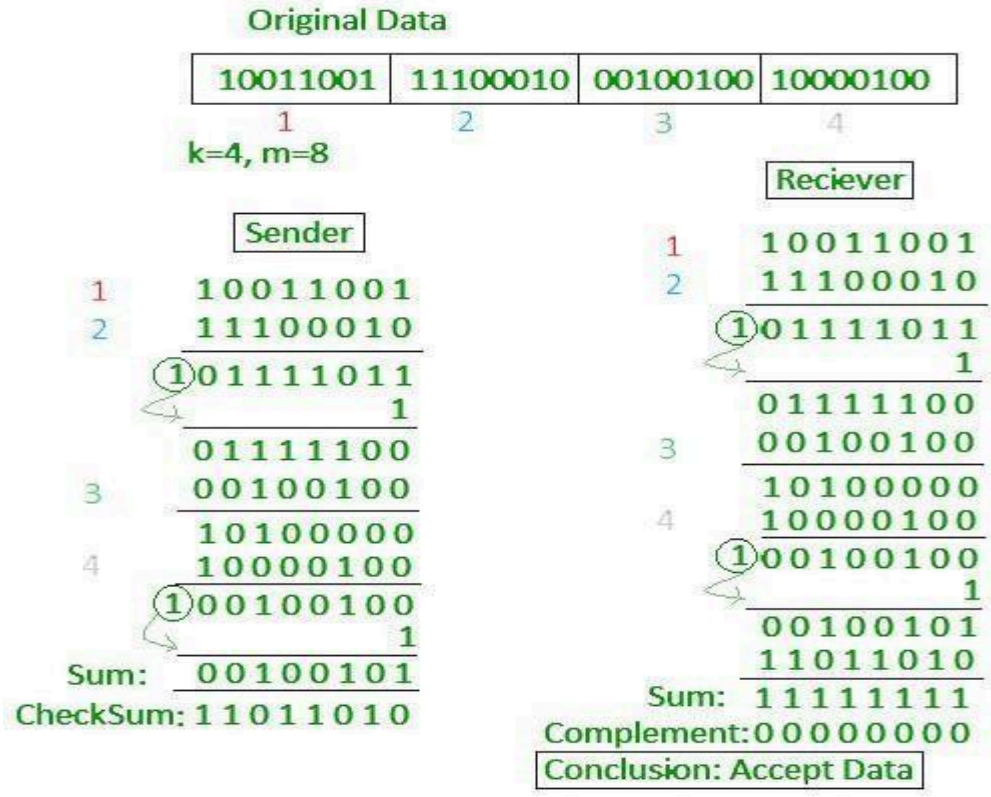
Two-dimensional Parity check

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

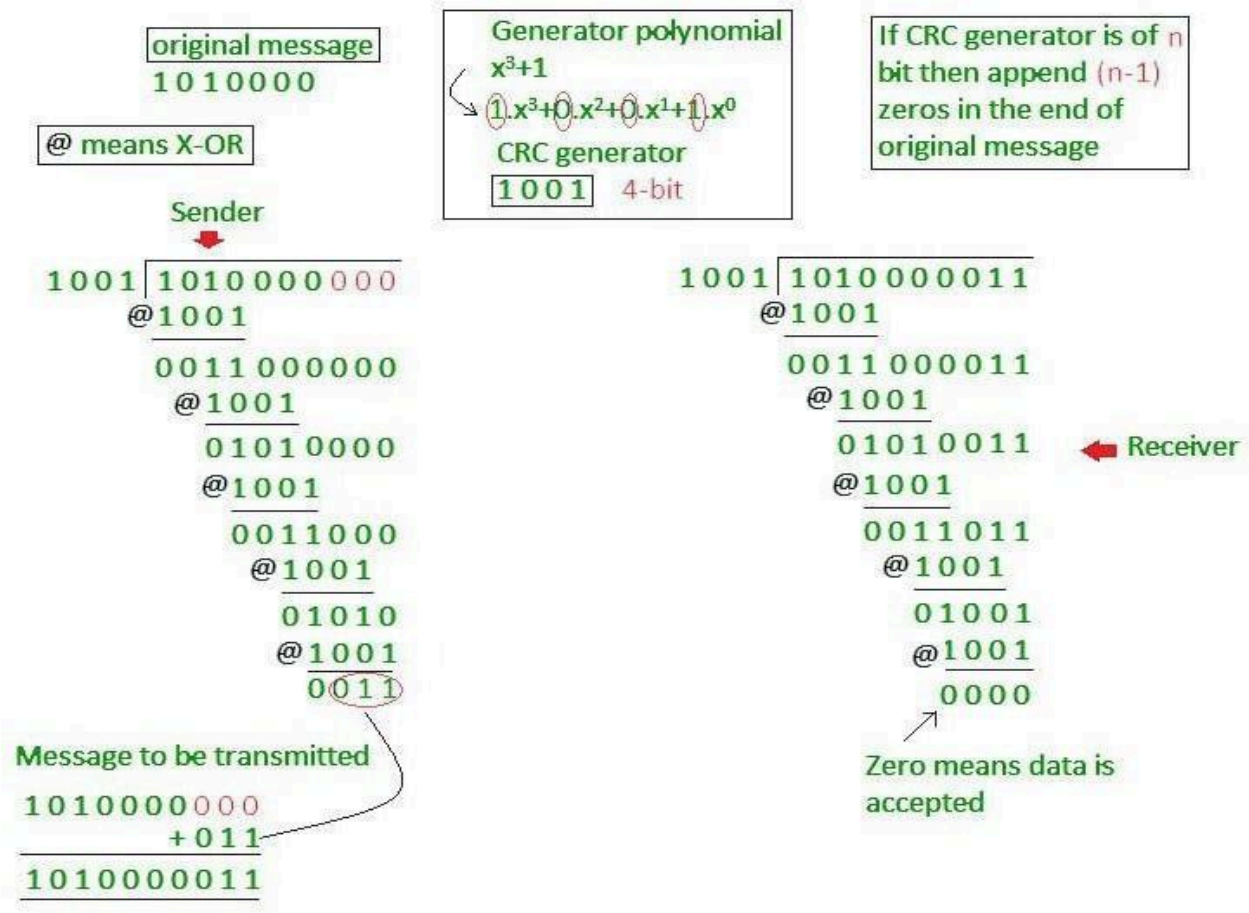


Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.



Cyclic redundancy check (CRC)



- Unlike 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 data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.

Error Correction

Error Correction codes are used to detect and correct the errors when data is transmitted from the sender to the receiver.

Error Correction can be handled in two ways:
Backward error correction: Once the error is discovered, the receiver requests the sender to retransmit the entire data unit.
Forward error correction: In this case, the receiver uses the error-correcting

code which automatically corrects the errors.
A single additional bit can detect the error, but cannot correct it.

For correcting the errors, one has to know the exact position of the error. For example, If we want to calculate a single-bit error, the error correction code will determine which one of seven bits is in error. To achieve this, we have to add some additional redundant bits.

Suppose r is the number of redundant bits and d is the total number of the data bits. The number of redundant bits r can be calculated by using the formula:

$2^r >= d+r+1$

The value of r is calculated by using the above formula. For example, if the value of d is 4, then the possible smallest value that satisfies the above relation would be 3.

To determine the position of the bit which is in error, a technique developed by R.W Hamming is Hamming code which can be applied to any length of the data unit and uses the relationship between data units and redundant units.

Hamming Code

Parity bits: The bit which is appended to the original data of binary bits so that the total number of 1s is even or odd.

Even parity: To check for even parity, if the total number of 1s is even, then the value of the parity bit is 0. If the total number of 1s occurrences is odd, then the value of the parity bit is 1.

Odd Parity: To check for odd parity, if the total number of 1s is even, then the value of parity bit is 1. If the total number of 1s is odd, then the value of parity bit is 0.

Algorithm of Hamming code:

An information of 'd' bits are added to the redundant bits 'r' to form d+r. The location of each of the (d+r) digits is assigned a decimal value.

The 'r' bits are placed in the positions 1,2, 2^{k-1}

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 : $2^r >= d+r+1$

$2^r >= 4+r+1$

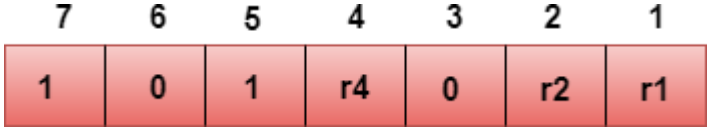
Therefore, the value of r is 3 that satisfies the above relation. Total number of bits = d+r = 4+3 = 7;

Determining the position of the redundant bits

The number of redundant bits is 3. The three bits are represented by r1, r2, r4. 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.

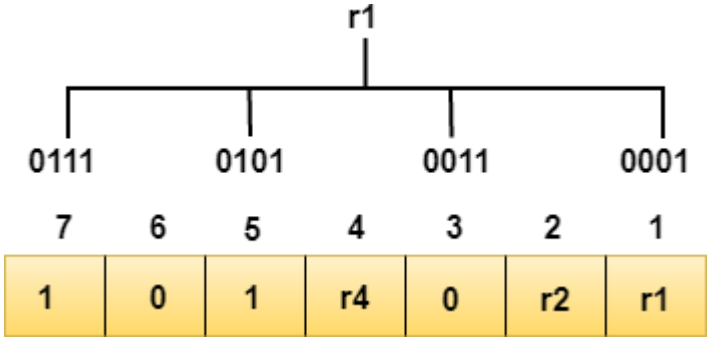
The position of r1 = 1, The position of r2 = 2 , The position of r4 = 4

Representation of Data on the addition of parity bits:



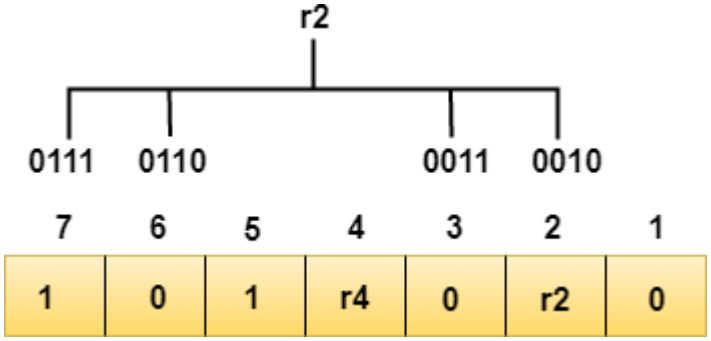
Determining the Parity bits

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 position 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.

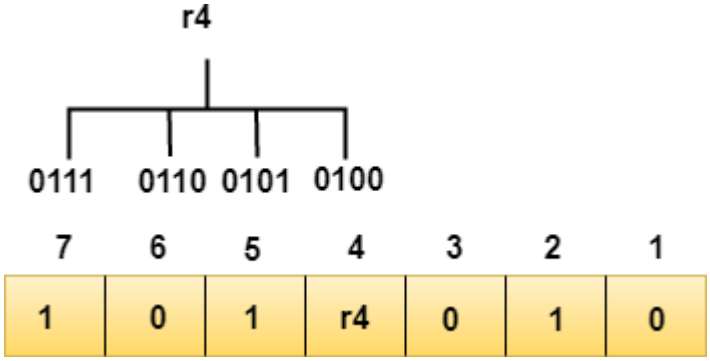
Determining r2 bit: The r2 bit is calculated by performing a parity check on the bit positions whose binary representation includes 1 in the second position



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

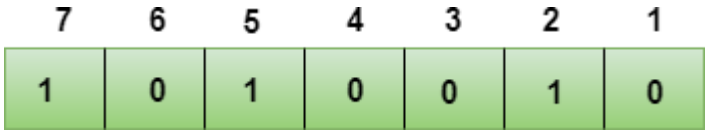
bit positions. The total number of 1 at these bit positions corresponding to r2 is odd, therefore, the value of the r2 bit is 1.

Determining 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 includes 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.

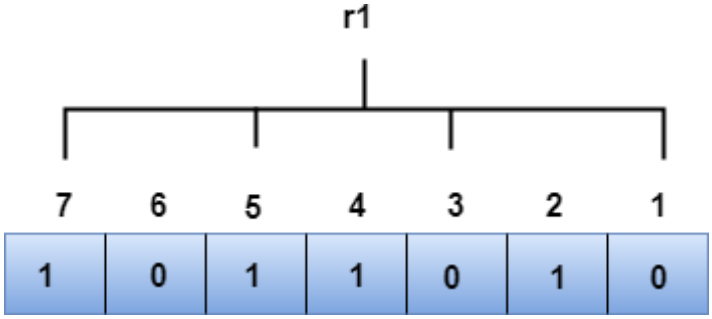
Data transferred is given below:



Suppose the 4th bit is changed from 0 to 1 at the receiving end, then parity bits are recalculated.

R1 bit

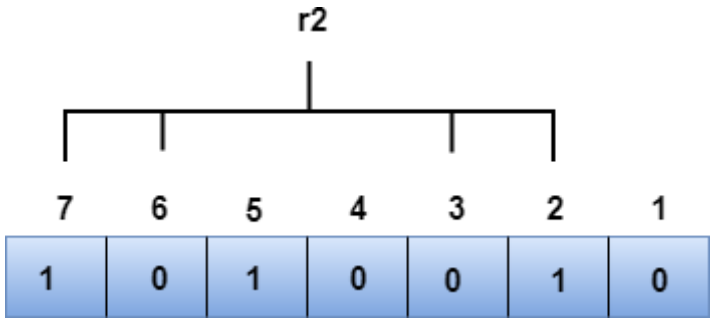
The bit positions of the r1 bit are 1,3,5,7



We observe from the above figure that the binary representation of r1 is 1100. Now, we perform the even-parity check, the total number of 1s appearing in the r1 bit is an even number. Therefore, the value of r1 is 0.

R2 bit

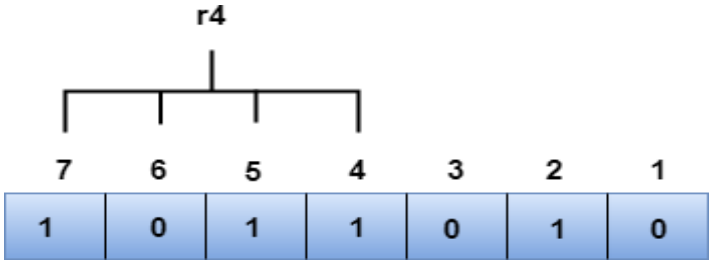
The bit positions of r2 bit are 2,3,6,7.



We observe from the above figure that the binary representation of r2 is 1001. Now, we perform the even-parity check, the total number of 1s appearing in the r2 bit is an even number. Therefore, the value of r2 is 0.

R4 bit

The bit positions of r4 bit are 4,5,6,7.



We observe from the above figure that the binary representation of r4 is 1011. Now, we perform the even-parity check, the total number of 1s appearing in the r4 bit is an odd number. Therefore, the value of r4 is 1.

The binary representation of redundant bits, i.e., r4r2r1 is 100, and its corresponding decimal value is 4. Therefore, the error occurs in a 4th bit position. The bit value must be changed from 1 to 0 to correct the error.

UNIT -3 IMPORTANT QUESTIONS (LONG ANSWER) FOR END SEMESTER/FINAL EXAM

1) EXPLAIN DESIGN ISSUES OF NETWORK LAYER
2) EXPLAIN DISTANCE VECTOR ROUTING ALGORITHM ALONG WITH COUNT TO INFINITY PROBLEM
3) EXPLAIN HIERARCHICAL ROUTING ALGORITHM
4) EXPLAIN DIJKSTRA'S SHORTESH PATH ALGORITHM WITH AN EXAMPLE
5) EXPLAIN VARIOUS CONGESTION CONTROL POLICIES
6) EXPLAIN LEAKY BUCKET AND TOKEN BUCKET ALGORITHMS

SHORT ANSWER QUESTIONS(2M OR 3M)

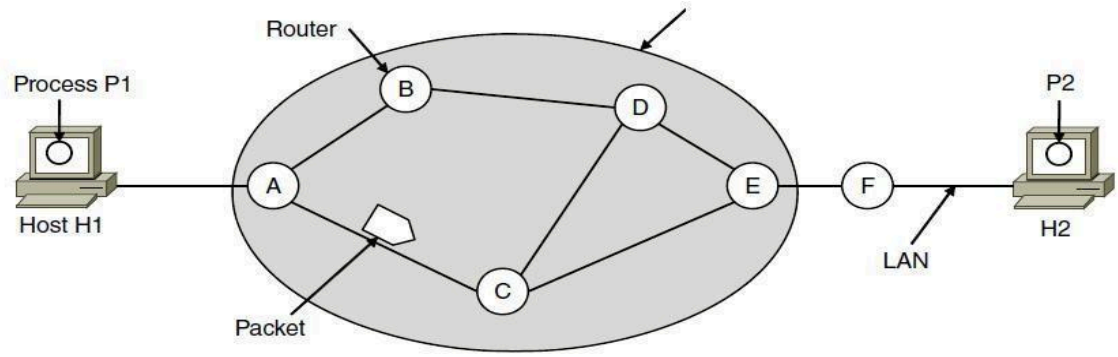
1) **Tunneling**
2) **ARP**
3) **ICMP Messages**
4) **IPv4 header format**
5) **Fragmentation**

UNIT-III

Network Layer Design Issues(VERY VERY IMPORTANT)

1. Store-and-forward packet switching
2. Services provided to transport layer
3. Implementation of connectionless service
4. Implementation of connection-oriented service
5. Comparison of virtual-circuit and datagram networks

1 Store-and-forward packet switching



A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the ISP. The packet is stored there until it has fully arrived and the link has finished its processing by verifying the checksum. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching.

2 Services provided to transport layer

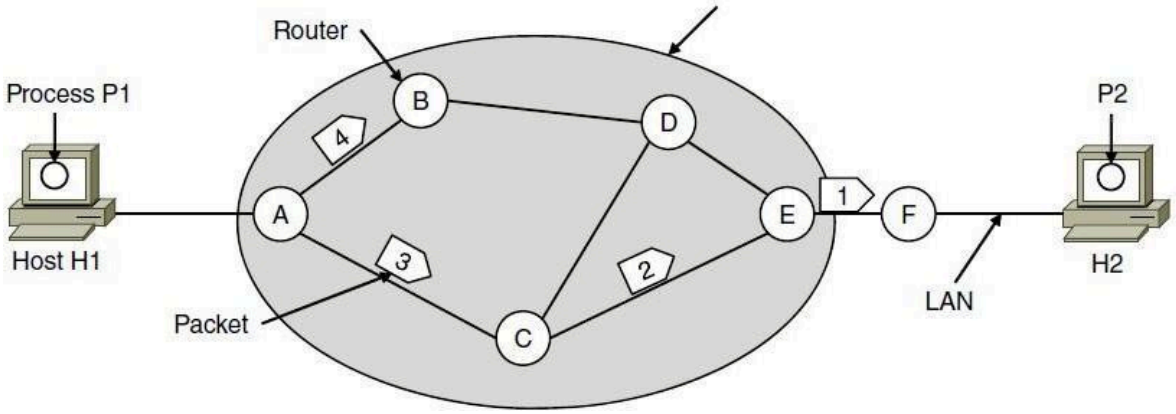
The network layer provides services to the transport layer at the network layer/transport layer interface. The services need to be carefully designed with the following goals in mind:

1. Services independent of router technology.
2. Transport layer shielded from number, type, topology of routers.
3. Network addresses available to transport layer use uniform numbering plan
   - even across LANs and WANs

3 Implementation of 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** (in analogy with telegrams) and the network is called a **datagram network**.



A's table (initially)

Dest.	Line
A	☒
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	☒
B	B
C	C
D	B
E	D
F	D

C's Table

A	A
B	A
C	☒
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	☒
F	F

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.*

Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair(destination and the outgoing line). Only directly connected lines can be used.

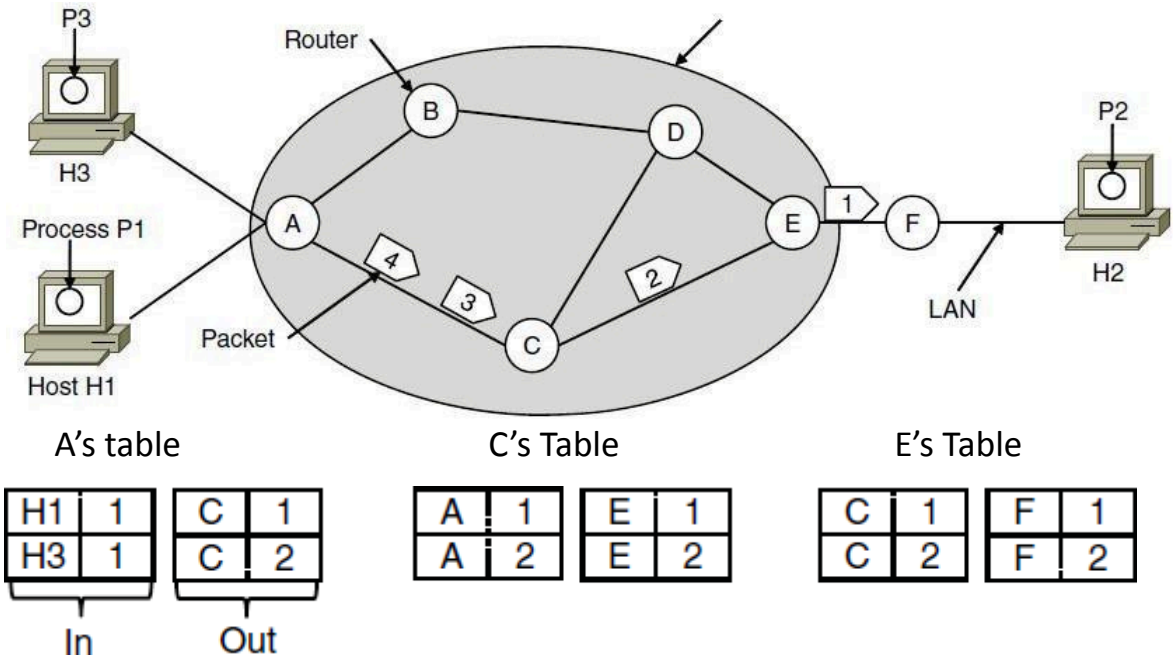
A's initial routing table is shown in the figure under the label ''initially.''

At *A*, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link. 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*.

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 (traffic jam along ACE path), *A* decided to send packet 4 via a different route than that of the first three packets. Router A 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**.

4 Implementation of 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**), and the network is called a **virtual-circuit network**

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 Figure. 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**. An example of a connection-oriented network service is **MPLS** (**Multi Protocol Label Switching**).

5 **Comparison of virtual-circuit and datagram networks**

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

The main function of NL (Network Layer) is routing packets from the source machine to the destination machine.
There are two processes inside router:

a) One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing table. This process is forwarding.
b) The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play. This process is routing.

Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm **correctness, simplicity, robustness, stability, fairness, optimality**

Routing algorithms can be grouped into two major classes:

1) nonadaptive (Static Routing)
2) adaptive. (Dynamic Routing)

Nonadaptive algorithm do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J is computed in advance, off line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

Adaptive algorithm, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well.

Adaptive algorithms differ in

1) Where they get their information (e.g., locally, from adjacent routers, or from all routers),
2) When they change the routes (e.g., every ΔT sec, when the load changes or when the topology changes), and
3) What metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

This procedure is called dynamic routing

Different Routing Algorithms

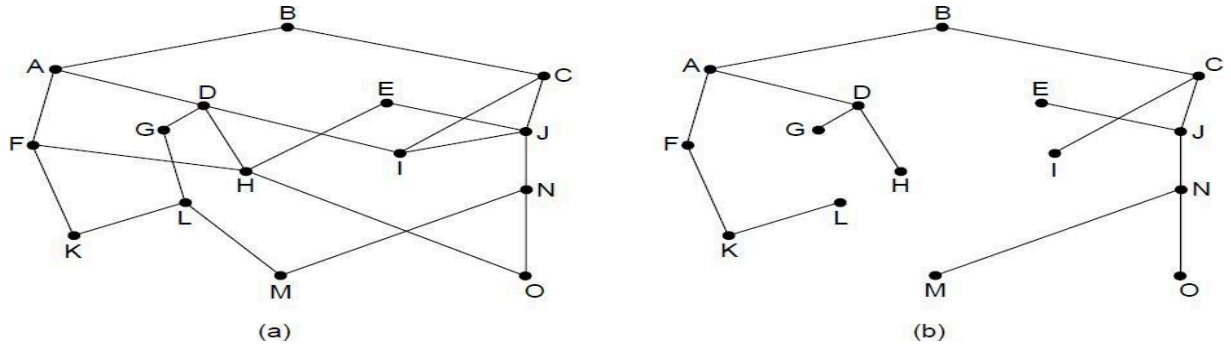
- Optimality principle
- Shortest path algorithm
- Flooding
- Distance vector routing
- Link state routing
- Hierarchical Routing

The Optimality Principle

One can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle.

It states that if 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

As a direct consequence of the optimality principle, we can see that the set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree**. The goal of all routing algorithms is to discover and use the sink trees for all routers



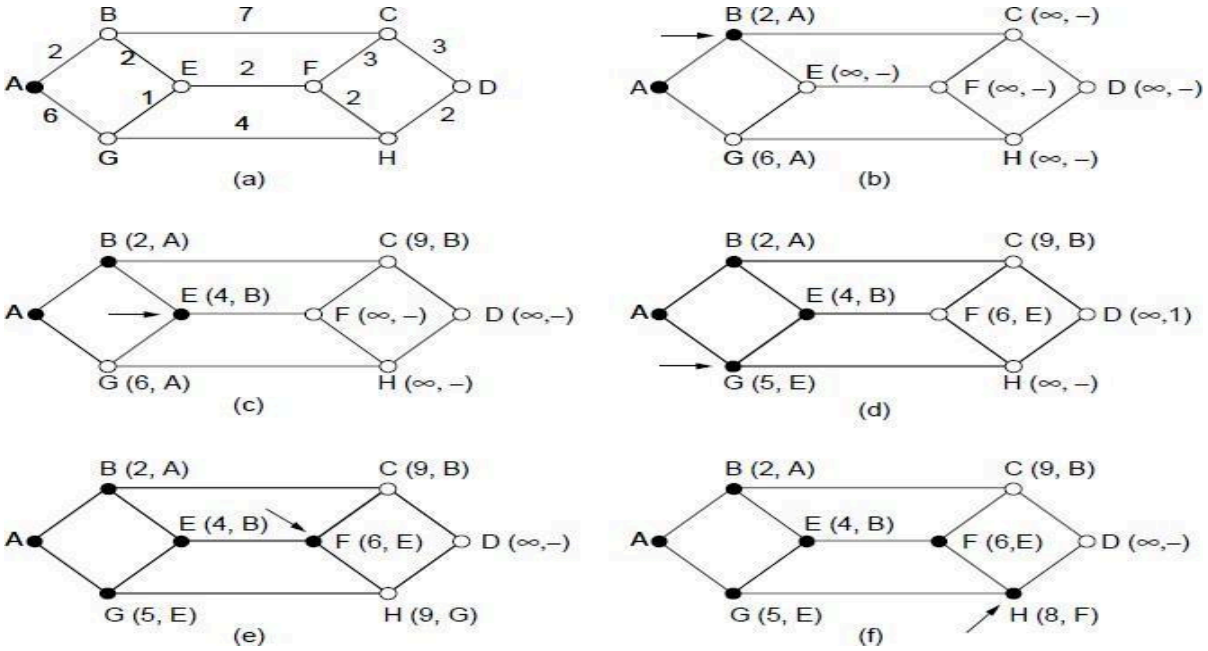
(a) A network. (b) A sink tree for router *B.*

Shortest Path Routing (Dijkstra's)(very very important)

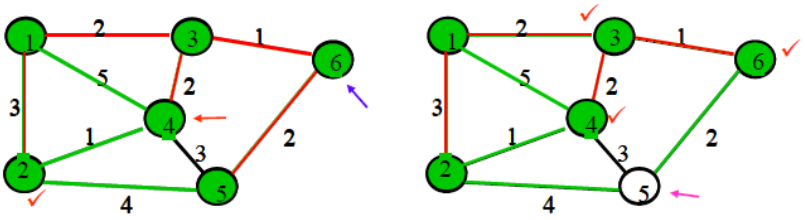
The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line or link.

To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph

1. Start with the local node (router) as the root of the tree. Assign a cost of 0 to this node and make it the first permanent node.
2. Examine each neighbor of the node that was the last permanent node.
3. Assign a cumulative cost to each node and make it tentative
4. Among the list of tentative nodes
    a. Find the node with the smallest cost and make it Permanent
    b. If a node can be reached from more than one route then select the route with the shortest cumulative cost.
5. Repeat steps 2 to 4 until every node becomes permanent



Execution of Dijkstra's algorithm



Iteration	Permanent	tentative	D_2	D_3	D_4	D_5	D_6
Initial	{1}	{2,3,4}	3	2 ✓	5	∝	∝
1	{1,3}	{2,4,6}	3 ✓	2	4	∝	3
2	{1,2,3}	{4,6,5}	3	2	4	7	3 ✓
3	{1,2,3,6}	{4,5}	3	2	4 ✓	5	3
4	{1,2,3,4,6}	{5}	3	2	4	5 ✓	3
5	{1,2,3,4,5,6}	{}	3	2	4	5	3

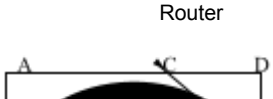
Flooding

- Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on.
- Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.
- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination.
- A variation of flooding that is slightly more practical is **selective flooding**. In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction.
- Flooding is not practical in most applications.

Distance Vector Routing:(very very important)

A **distance vector routing** algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there. These tables are updated by exchanging information with the neighbors. In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the network. This entry has two parts: the preferred outgoing line to use for that destination and an estimate of the distance to that destination. The distance might be measured as the number of hops or using another metric.

This updating process is illustrated in Fig. 5-9. Part (a) shows a network. The first four columns of part (b) show the delay vectors received from the neighbors of router *J*. *A* claims to have a 12-msec delay to *B*, a 25-msec delay to *C*, a 40- msec delay to *D*, etc. Suppose that *J* has measured or estimated its delay to its neighbors, *A*, *I*, *H*, and *K*, as 8, 10, 12, and 6 msec, respectively.



E H

I J K L

Newestimateddelayfromj

0
12
25
40
14
23
18
17
21
9
24
29

24
36
18
27
7
20
31
20
0
11
22
33

20
31
19
8
30
19
6
0
14
7
22
9

21
28
36
24
22
40
31

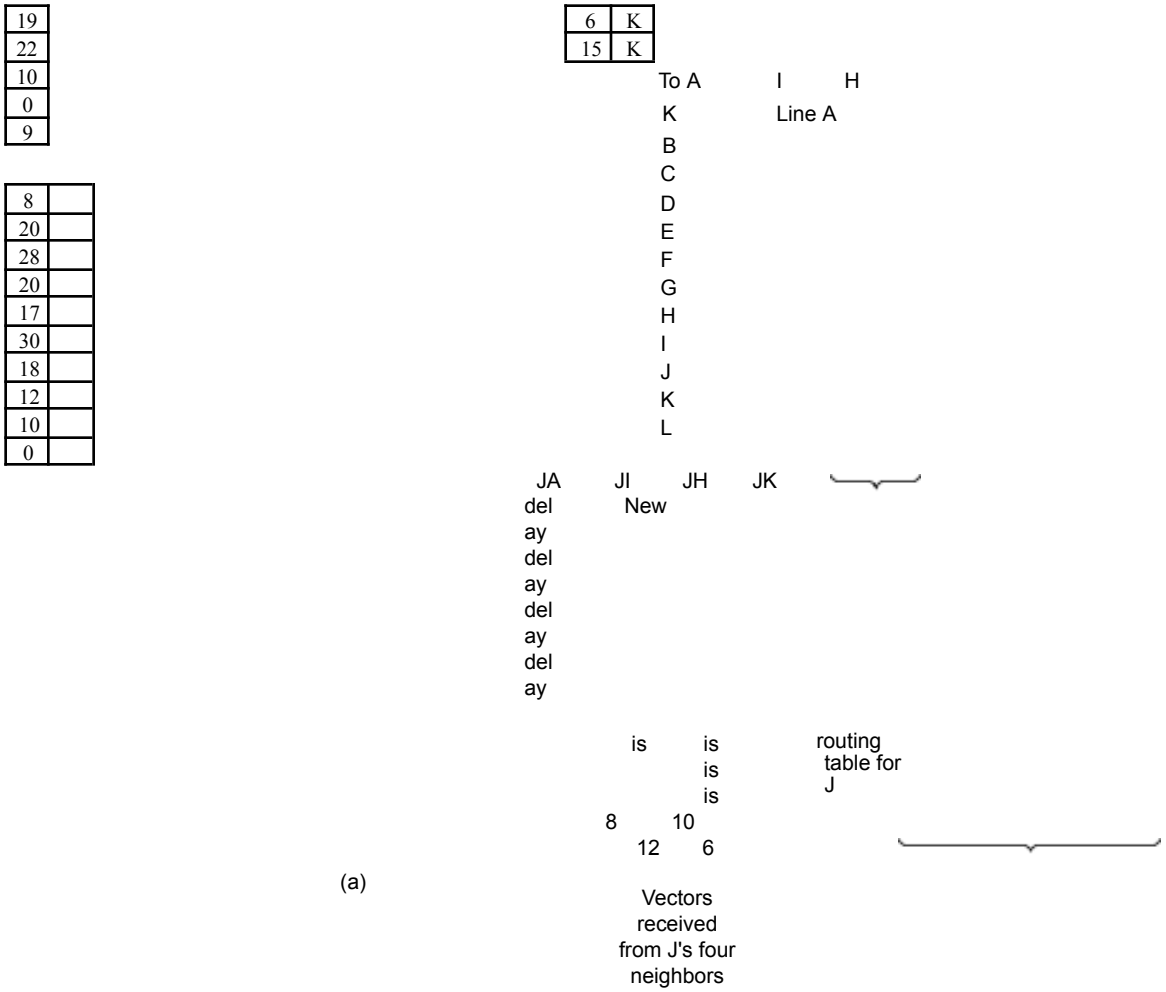


Figure 5-9. (a) A network. (b) Input from *A*, *I*, *H*, *K*, and the new routing table for *J*.

Consider how *J* computes its new route to router *G*. It knows that it can get to *A* in 8 msec, and furthermore *A* claims to be able to get to *G* in 18 msec, so *J* knows it can count on a delay of 26 msec to *G* if it forwards packets bound for *G*

to *A*. Similarly, it computes the delay to *G* via *I*, *H*, and *K* as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) msec, respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to *G* is 18 msec and that the route to use is via *H*. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.

The Count-to-Infinity Problem

The settling of routes to best paths across the network is called **convergence**. Distance vector routing is useful as a simple technique by which routers can col- lectively compute shortest paths, but it has a serious drawback in practice: al- though it converges to the correct answer, it may do so slowly. In particular, it reacts rapidly to good news, but leisurely to bad news. Consider a router whose best route to destination *X* is long. If, on the next exchange, neighbor *A* suddenly reports a short delay to *X*, the router just switches over to using the line to *A* to send traffic to *X*. In one vector exchange, the good news is processed.

A	B	C	D	E	
	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

(a)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		.			
	•	•	•	•	

(b)

To see how fast good news propagates, consider the five-node (linear) net- work of Fig. 5-10, where the delay metric is the number of hops. Suppose *A* is down initially and all the other routers know this. In other words, they have all recorded the delay to *A* as infinity.

Figure 5-10. The count-to-infinity problem.

When *A* comes up, the other routers learn about it via the vector exchanges. For simplicity, we will assume that there is a gigantic gong somewhere that is struck periodically to initiate a vector exchange at all routers simultaneously. At the time of the first exchange, *B* learns that its left-hand neighbor has zero delay to *A*. *B* now makes an entry in its routing table indicating that *A* is one hop away to the left. All the other routers still think that *A* is down. At this point, the rout- ing table entries for *A* are as shown in the second row of Fig.

5-10(a). On the next exchange, *C* learns that *B* has a path of length 1 to *A*, so it updates its routing table to indicate a path of length 2, but *D* and *E* do not hear the good news until later. Clearly, the good news is spreading at the rate of one hop per exchange. In a net- work whose longest path is of length *N* hops, within *N* exchanges everyone will know about newly revived links and routers.

Now let us consider the situation of Fig. 5-10(b), in which all the links and routers are initially up. Routers *B*, *C*, *D*, and *E* have distances to *A* of 1, 2, 3, and 4 hops, respectively. Suddenly, either *A* goes down or the link between *A* and *B* is cut (which is effectively the same thing from *B*'s point of view).

At the first packet exchange, *B* does not hear anything from *A*. Fortunately, *C* says ''Do not worry; I have a path to *A* of length 2.'' Little does *B* suspect that *C*'s path runs through *B* itself. For all *B* knows, *C* might have ten links all with sepa- rate paths to *A* of length 2. As a result, *B* thinks it can reach *A* via *C*, with a path length of 3. *D* and *E* do not update their entries for *A* on the first exchange.

On the second exchange, *C* notices that each of its neighbors claims to have a path to *A* of length 3. It picks one of them at random and makes its new distance to *A* 4, as shown in the third row of Fig. 5-10(b). Subsequent exchanges produce the history shown in the rest of Fig. 5-10(b).

From this figure, it should be clear why bad news travels slowly: no router ever has a value more than one higher than the minimum of all its neighbors. Gradually, all routers work their way up to infinity, but the number of exchanges required depends on the numerical value used for infinity. For this reason, it is wise to set infinity to the longest path plus 1.

Not entirely surprisingly, this problem is known as the **count-to-infinity** prob- lem.

Hierarchical Routing: 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, as it is in the telephone network.

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 dest- inations within its own region but knows nothing about the internal structure of other regions. When different networks are interconnected, it is natural to regard each one as a separate region to free the routers in one network from having to know the topological structure of the other ones.For huge networks, a two-level hierarchy may be insufficient; it may be nec- essary to group the regions into clusters, the clusters into zones, the zones into groups, and so on.

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 en- tries 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 rout- ers per region grows, the savings in table space increase.

Full table for 1A

Dest. 1A	Line	Hops
Region 1		
Region 2		
–	–	
1B		
1C		
1B		2
1B		3
1B		3
1B		4
1C		3

Hierarchical table for 1A

Dest. 1A	Line	Hops

1C	2
1C	3
1C	4
1C	4
1C	4
1C	5
1B	5
1C	6
1C	5

–	–
1B	1
1C	1
1B	2
1C	2
1C	3
1C	4

1B 2A 2B 1B 1B

1A 1C 1C

2C 2D 2A 2

1C 2B 3

2C 4

2D 5

3A 4A

3B 4B 5B

5C

3A

5A

3B

4C

5D

4A

4B

Region 3 Region 4 Region 5 4C

5A

5B

5C

5D

5E

(a) (b) (a)

Figure 5-14. Hierarchical routing.

Broadcast Routing

In some applications, hosts need to send messages to many or all other hosts. Sending a packet to all destinations simultaneously is called broadcasting. Various methods have been proposed for doing it.

One broadcasting method that requires no special features from the network is for the source to simply send a distinct packet to each destination.

multidestination routing, in which each packet contains either a list of destinations or a bit map indicating the desired destinations. When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed.

flooding. When implemented with a sequence number per source, flooding uses links efficiently with a decision rule at routers that is relatively simple.

reverse path forwarding: When a broadcast packet ar- rives at a router, the router checks to see if the packet arrived on the link that is normally used for sending packets *toward* the source of the broadcast. If so, there is an excellent chance that the broadcast packet itself followed the best route from the router.

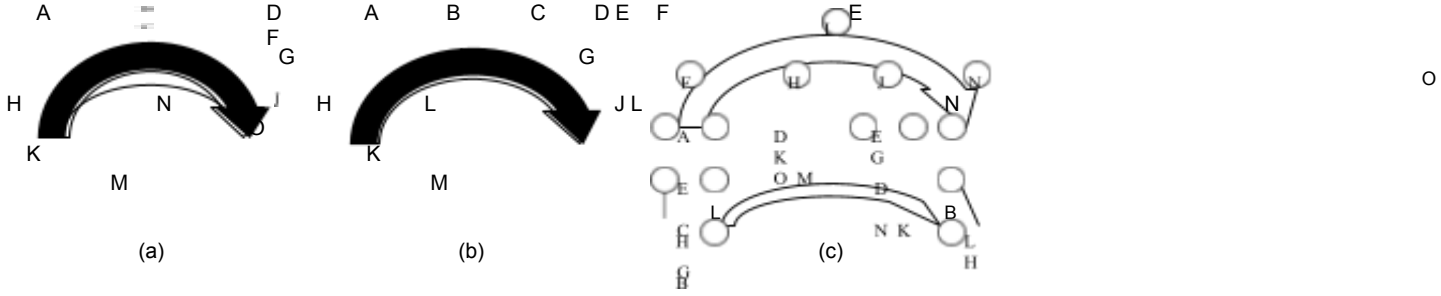


Figure 5-15. Reverse path forwarding. (a) A network. (b) A sink tree. (c) The tree built by reverse path forwarding.

An example of reverse path forwarding is shown in Fig. 5-15. Part (a) shows a network, part (b) shows a sink tree for router *I* of that network, and part (c) shows how the reverse path algorithm works. On the first hop, *I* sends packets to *F*, *H*, *J*, and *N*, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to *I* (assuming that the preferred path falls along the sink tree) and is so indicated by a circle around the letter. On the second hop,

eight packets are generated, two by each of the routers that received a packet on the first hop. As it turns out, all eight of these arrive at previously unvisited rout- ers, and five of these arrive along the preferred line. Of the six packets generated on the third hop, only three arrive on the preferred path (at *C*, *E*, and *K*); the oth- ers are duplicates. After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly.

A **spanning tree** is a subset of the network that includes all the routers but contains no loops. Sink trees are spanning trees. If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on.

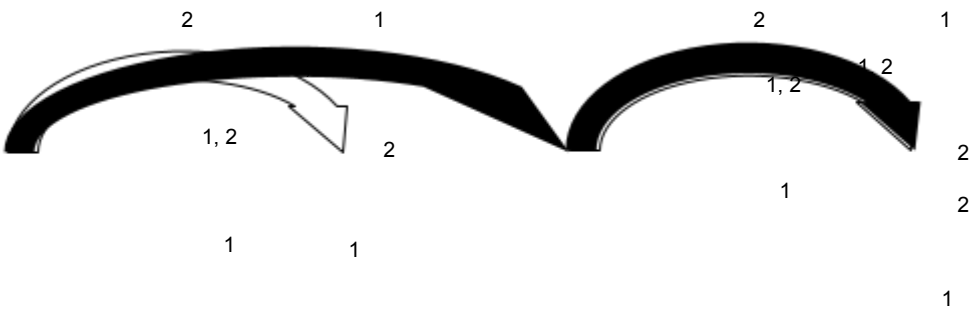
Multicast Routing

Some applications, such as a multiplayer game or live video of a sports event streamed to many viewing locations, send packets to multiple receivers.

Sending a message to such a group is called **multicasting**, and the routing al- gorithm used is called **multicast routing**. All multicasting schemes require some way to create and destroy groups and to identify which routers are members of a group.

As an example, consider the two groups, 1 and 2, in the network shown in Fig. 5-16(a). Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure. A spanning tree for the leftmost router is shown in Fig. 5-16(b). This tree can be used for broadcast but is overkill for mu- lticast, as can be seen from the two pruned versions that are shown next. In Fig. 5-16(c), all the links that do not lead to hosts that are members of group 1 have been removed. The result is the multicast spanning tree for the leftmost router to send to group 1. Packets are forwarded only along this spanning tree, which is more efficient than the broadcast tree because there are 7 links instead of

10. Fig. 5-16(d) shows the multicast spanning tree after pruning for group 2. It is efficient too, with only five links this time. It also shows that different multicast groups have different spanning trees.


2
1
2
1
1, 2
1, 2
2
2
1
2
1
1
1

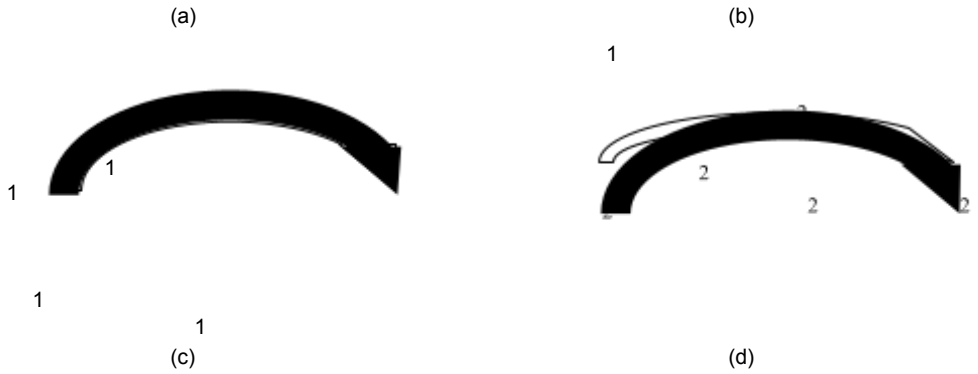


Figure 5-16. (a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.

CONGESTION AND QUALITY OF SERVICE(QOS) IN A NETWORK

Congestion: It occurs in a network when load of a network is greater than capacity of the network.

Congestion Control Techniques or Policies: (VERY VERY IMPORTANT)

These techniques or policies are used prevent the congestion before it happens or remove the congestion after it happens.
These techniques are two types

1)Open loop congestion control techniques
2)Closed loop congestion control techniques

Open loop congestion control techniques:

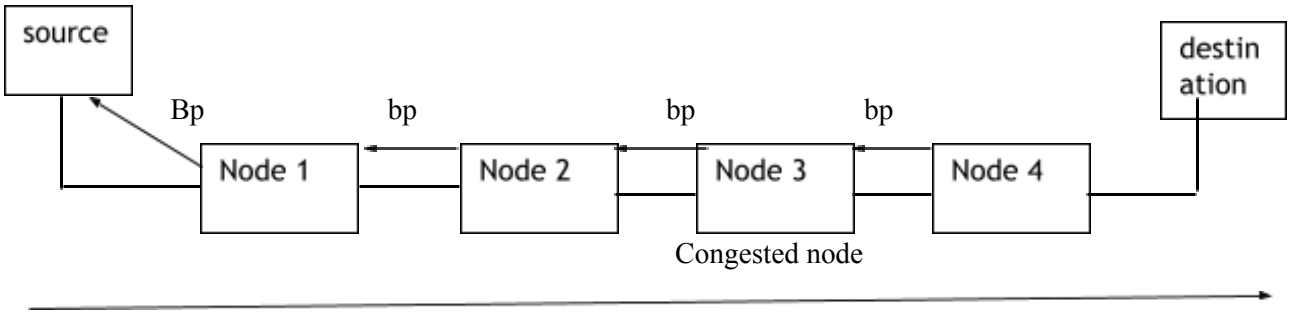
These techniques or policies are used prevent the congestion before it happens
The policies are
1)Retransmission policy
2)window policy
3) Acknowledgement policy
4)Discarding policy

1)Retransmision policy: packet or data or acknowledgement is sent when it is lost or damaged.Generaaly it leads to more load or congestion on a network.but good transmission policy available in TCP reduces the congestion.
2)Window policy: Selective repeat protocol has good window policy than GOBACKN.In GOBACK N if one frame is lost all the subsequent successful frames need to be transmitted again where as in the selective repeat only the lost frame retransmitted.
3)Acknowledgement Policy: good acknowledgement policy prevents the congestion.This is possible in 2 ways.
a)send acknowledgement for all the frames atba time with single message.
b)send the acknowledgement only when receiver has the data to send.
4.Discarding Policy:
Good discarding policy also prevents the congestion.In audio transmission applications sensitive or corrupted packets will be discarded by the routers with out affecting the quality of sound.

Closed loop congestion control techniques:

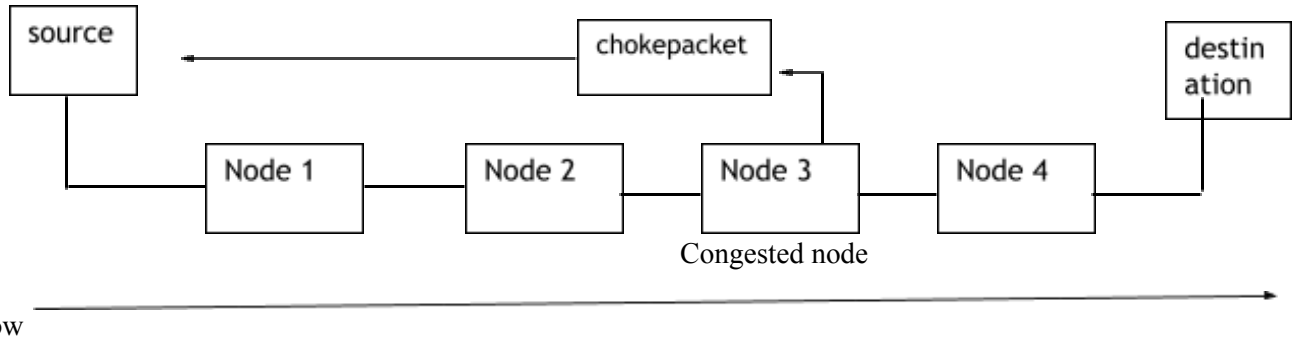
These techniques or policies are used remove the congestion after it happens
The policies are
1) Back pressure
2) Choke packet
3) Implicit Signaling
4) Explicit Signaling

1)Back Pressure(BP):
Assume that data is flowing from source to destination.



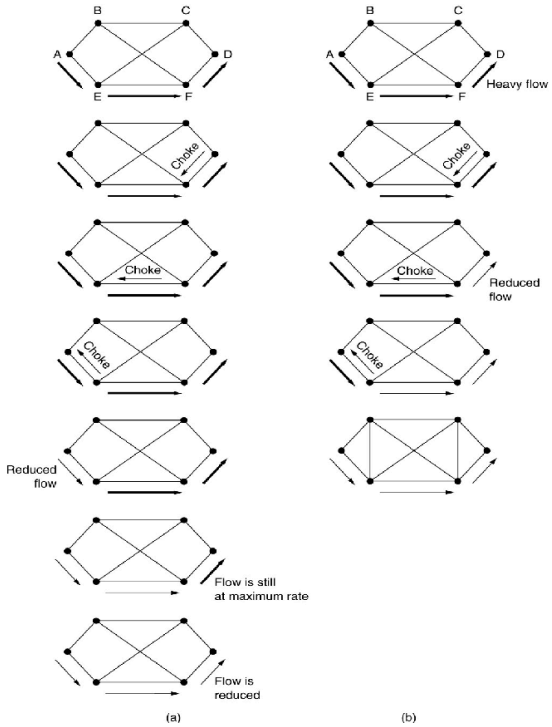
Suppose node3 is congested then it stops receiving the data from its immediate upstream node.so node 2 is congested ,it also follow the same till source node is congested.finally source node slows down the sending data to remove the congestion.
In backpressure nodes between congested node and source node affected due to congestion.

3)choke packet:
Assume that data is flowing from source to destination.



Suppose node3 is congested then it sends a special packet called choke packet directly to source node. node 2 is simply forwards the choke packet,similary other nodes alos.This process will be continued till it reaches to l source node .Finally source node slows down the sending data to remove the congestion.
In choke packet nodes between congested node and source node not affected due to congestion.

Hop-by-Hop Choke Packets



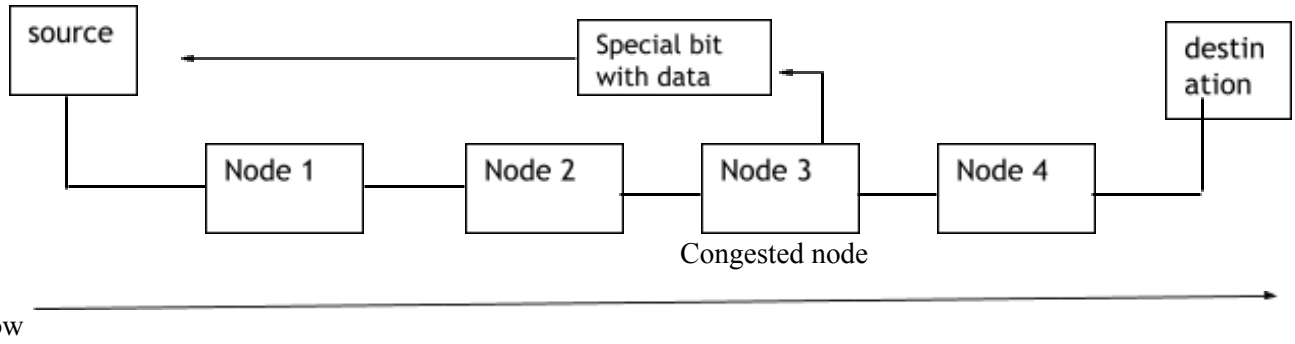
(a) A choke packet that affects only the source.

(b) A choke packet that affects each hop it passes through.

3)Implicit signaling:
It is also called LOAD SHEDDING.
In this no communication between source and congestion node. Source node guesses that congestion occurred in the network based on some symptoms.
For example sender sends the data and wait for acknowledgement for a while.If it does not receive the acknowledgement with in the time simply it slows down sending the data or packets by assuming congestion occurred.

4)Explicit Signaling:

Assume that data is flowing from source to destination.



Suppose node3 is congested then it sends a special signal in the form of bit directly to source node.This sepecial bit will be sent along with the data packet. Node 2 is simply forwards the choke packet,similary other nodes alos.This process will be continued till it reaches to l source node .Finally source node slows down the sending data to remove the congestion.
In choke packet nodes between congested node and source node not affected due to congestion.

Quality of Service:
Four characteristics determine the quality of service of a data.They are
1)Reliability: network should transfer the data or acknowledgement without lost.

2)delay: application like audio ,video transmissions require minimum delay than applications like file transfer and e mail applications.
3)Jitter: variation in packet's delay is called jitter or jitter control.
For example if four packets depart at times 0,1,2,3 and arrive at times 20,21,22,23 then delay respectively 20-0,21-1,22-2,23-3.
Suppose the arrival time is 25,28,21,29 then delay respectively 25-0,28-21,21-2,29-3. The variation in packet's delay is called jitter.
Min jitter is required for audio,video applications.

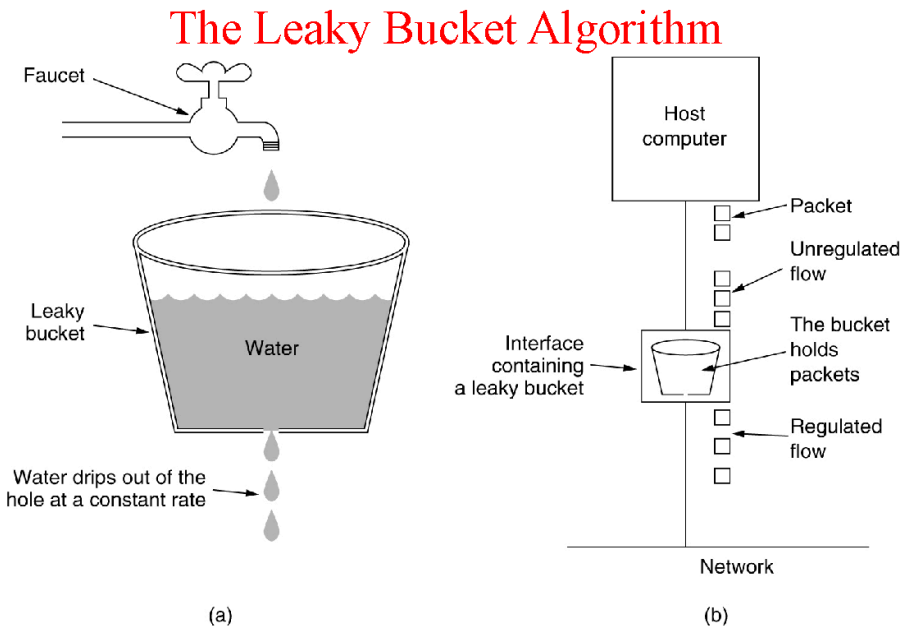
4)Bandwidth:
High bandwidth required for applications like audio,video transmission to transfer millions of bits per second to refresh the screen.
Low bandwidth required for email transmission.

Traffic Shaping:(very very important)

- **traffic shaping**, is a technique, which smooths out the traffic on the server side, rather than on the client side.
- Traffic shaping controls the *rate* at which packets are sent .
- Two traffic shaping algorithms are:
  a)Leaky Bucket
  b)Token Bucket

The Leaky Bucket Algorithm:

- Conceptually, each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded.
- The **Leaky Bucket Algorithm** used to control rate in a network.



(a) A leaky bucket with water. (b) a leaky bucket with packets.

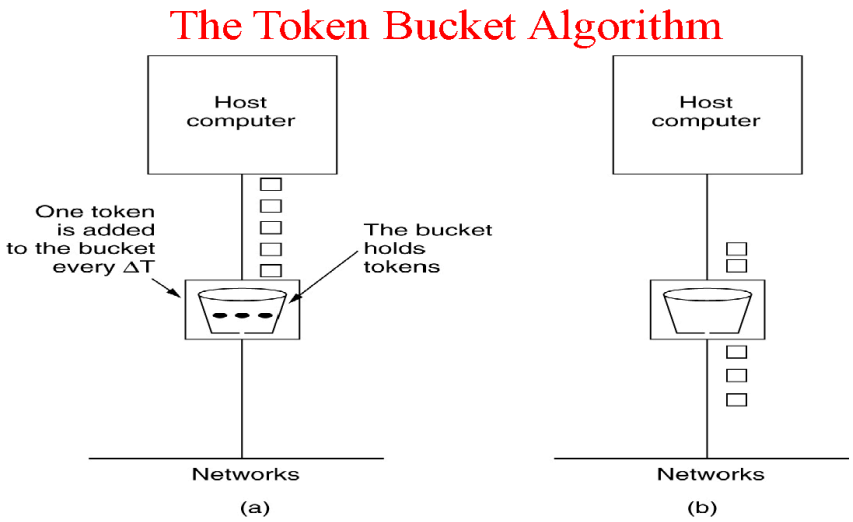
The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input

- The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.
- Implementing the original leaky bucket algorithm is easy. The leaky bucket consists of a finite queue. When a packet arrives, if there is room on the queue it is appended to the queue; otherwise, it is discarded. At every clock tick, one packet is transmitted (unless the queue is empty).

Token Bucket Algorithm:

- In contrast to the LB, the Token Bucket Algorithm, allows the output rate to vary, depending on the size of the burst.
- In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.

- Tokens are generated by a clock at the rate of one token every Δt sec.
- Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.



(a) Before. (b) After.

Internetworking Devices:

1. Repeater – A repeater operates at the physical layer. Its job is to regenerate the signal over the same network before the signal becomes too weak or corrupted so as to extend the length to which the signal can be transmitted over the same network.

2. Hub – A hub is basically a multiport repeater. A hub connects multiple wires coming from different branches, for example, the ***connector in star topology*** which connects different stations. Hubs cannot filter data, so data packets are sent to all connected devices. They do not have intelligence to find out best path for data packets which leads to inefficiencies and wastage.

3. Bridge – A bridge operates at data link layer. A bridge is a repeater, with add on functionality of filtering content by reading the MAC addresses of source and destination. It is also used for interconnecting two LANs working on the same protocol. It has a single input and single output port.

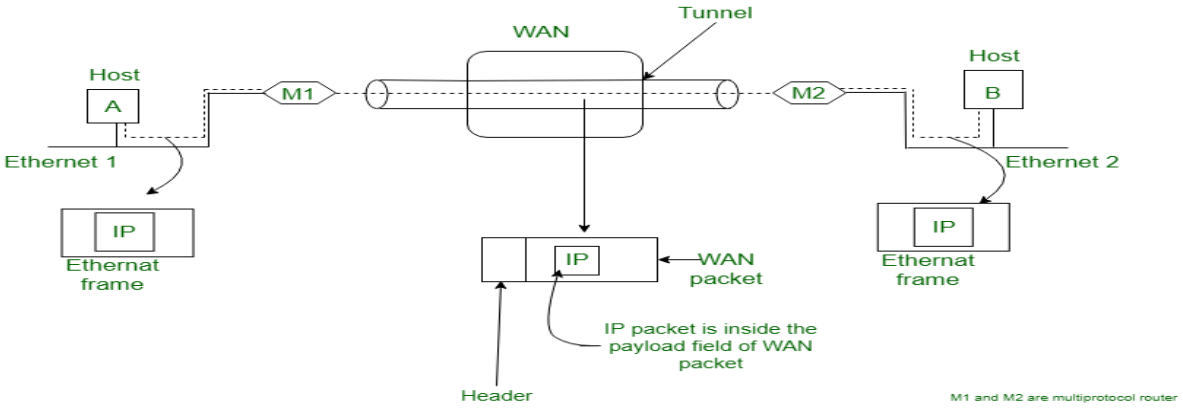
4. Switch – A switch is a multi port bridge with a buffer and a design that can boost its efficiency(large number of ports imply less traffic) and performance. Switch is data link layer device. Switch can perform error checking before forwarding data, that makes it very efficient as it does not forward packets that have errors and forward good packets selectively to correct port only.

5. Routers – A router is a device like a switch that routes data packets based on their IP addresses. Router is mainly a Network Layer device. Routers normally connect LANs and WANs together and have a dynamically updating routing table based on which they make decisions on routing the data packets.

6. Gateway – A gateway, as the name suggests, is a passage to connect two networks together that may work upon different networking models. They basically works as the messenger agents that take data from one system, interpret it, and transfer it to another system. Gateways are also called protocol converters and can operate at any network layer. Gateways are generally more complex than switch or router.

Tunneling:

A technique of internetworking called **Tunneling** is used when source and destination networks of same type are to be connected through a network of different type. For example, let us consider an Ethernet to be connected to another Ethernet through a WAN as:



Tunneling

The task is sent on an IP packet from host A of Ethernet-1 to the host B of ethernet-2 via a WAN.

Sequence of events:

1. Host A construct a packet which contains the IP address of Host B.
2. It then inserts this IP packet into an Ethernet frame and this frame is addressed to the multiprotocol router M1
3. Host A then puts this frame on Ethernet.
4. When M1 receives this frame, it removes the IP packet, inserts it in the payload packet of the WAN network layer packet and addresses the WAN packet to M2. The multiprotocol router M2 removes the IP packet and send it to host B in an Ethernet frame.

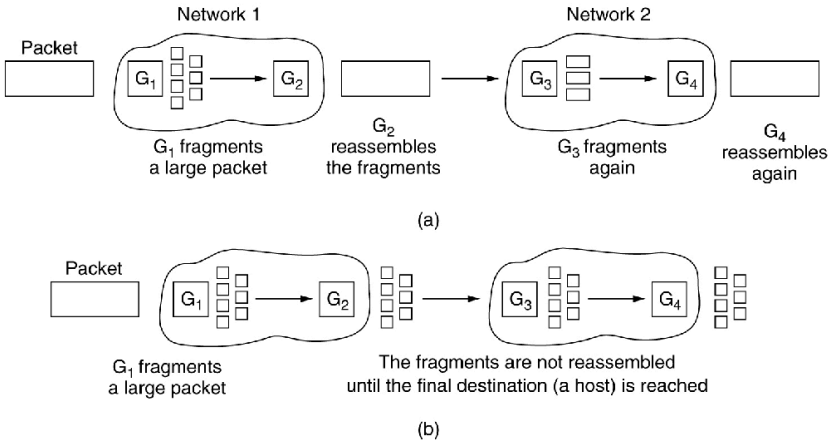
Fragmentation:

Different networks have different capacity of processing the packets. Small capacity networks cannot process large packets. Divide the large packets into small packets is called fragmentation. Two strategies used for fragmentation.

1)Transparent Fragmentation: In this large packet is divided in to small packets at the entry gateway of the network and reassembled or merged or combined at the exit gateway of the network. The same process is continued in all networks.

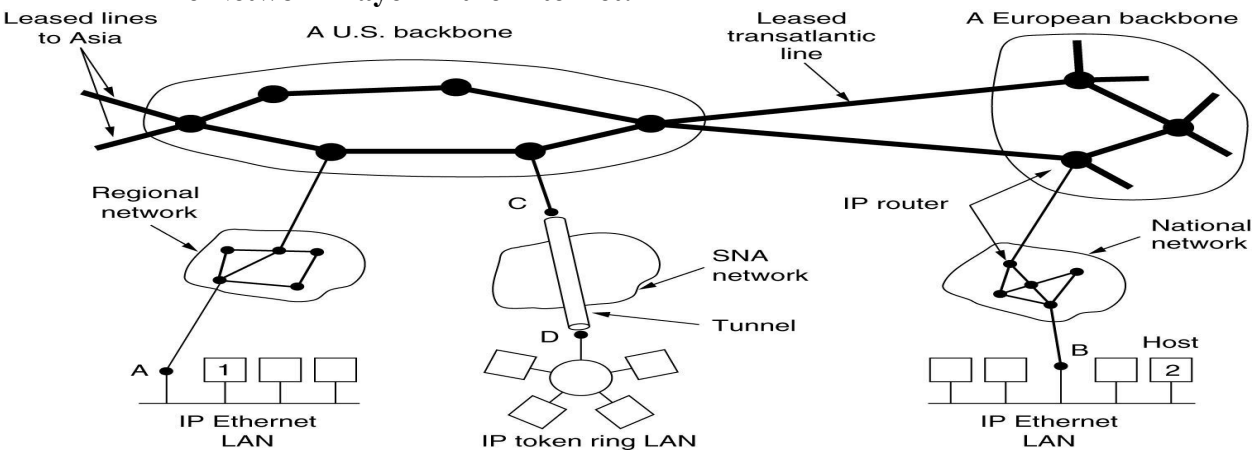
2)Non Transparent Fragmentation: In this large packet is divided in to small packets *only at the entry gateway of the first network* and reassembled or merged or combined only at the *destination*. The same process is continued in all networks.

Fragmentation



(a) Transparent fragmentation. (b) Nontransparent fragmentation.

The Network Layer in the Internet:

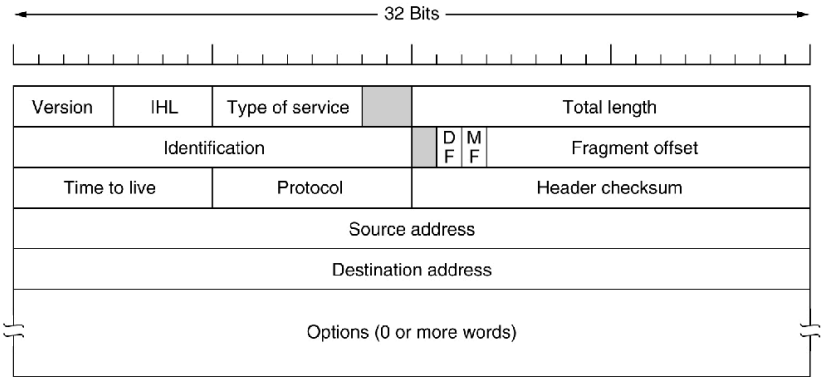


The Internet is an interconnected collection of many networks.
The network layer uses IP protocol to transmit the data grams .IP is unreliable,connectionless protocol.
There are 2 versions of IP.IP v4,IP v6

IP (internet protocol):

This contains min 20 bytes or max 60 bytes of header and variable size of data

The Internet Protocol(IP)

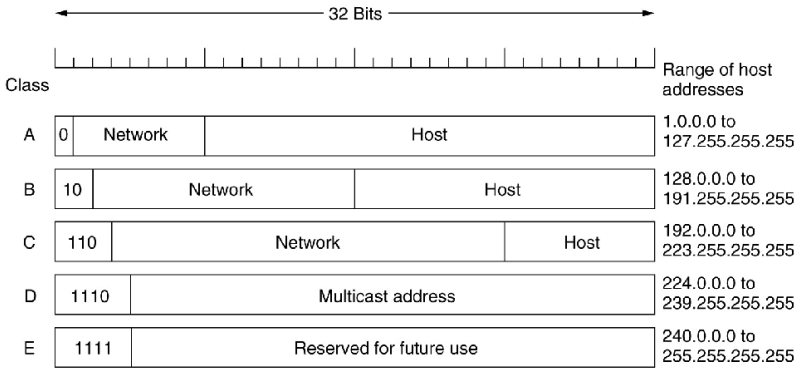


The IPv4 (Internet Protocol) header.

- The *Version* field keeps track of which version of the protocol
- *IHL*, is provided to tell how long the header is, in 32-bit words along with options
- The *Type of service* field is to distinguish between different classes of service.
- The *Total length* includes everything in the datagram—both header and data.
- The *Identification* field is needed. All the fragments of a datagram contain the same *Identification* value.
- *DF* stands for Don't Fragment. It is an order to the routers not to fragment the datagram because the destination is incapable of putting the pieces back together again.
- *MF* stands for More Fragments. All fragments except the last one have this bit set.
- The *Fragment offset* tells where in the current datagram this fragment belongs.
- The *Time to live* field is a counter used to limit packet lifetimes.
- The *Protocol* field tells it which transport process to give it to. TCP is one possibility, but so are UDP and some others.
- The *Header checksum* verifies the header only. Such a checksum is useful for detecting errors.
- The *Source address* and *Destination address* indicate the network number and host number.
- The options are variable length.

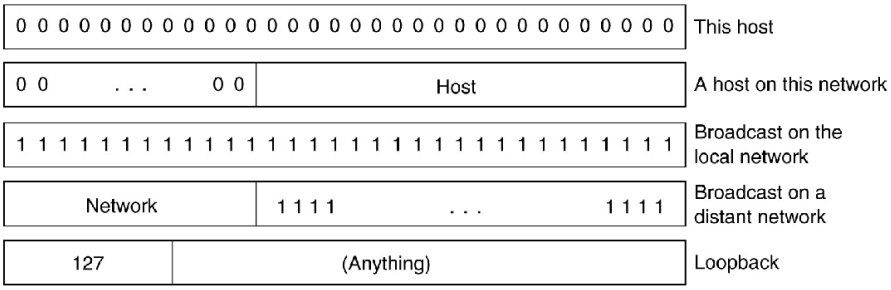
IP addresses:

IP Addresses



IP address formats.

IP Addresses ….



Special IP addresses.

ICMP(INTERNET CONTROLMESSAGE PROTOCOL):

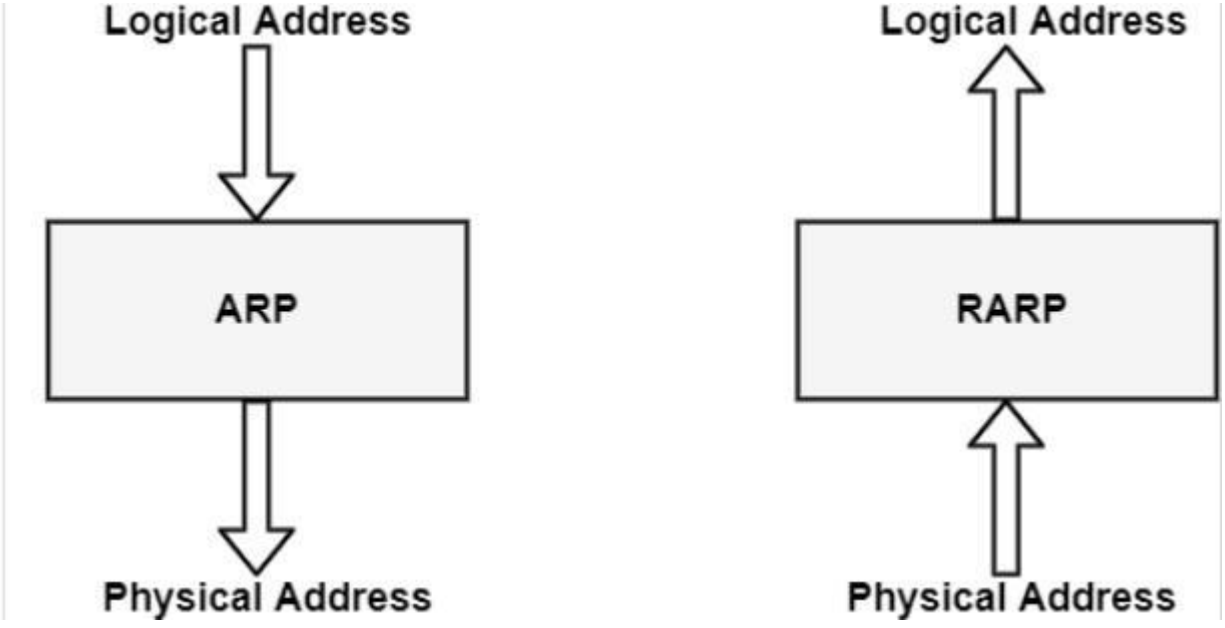
Internet Control Message Protocol

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

The principal ICMP message types.

ARP:

Address Resolution Protocol is important for changing the higher-level protocol address (IP addresses) to physical network addresses. It is described in RFC 826.



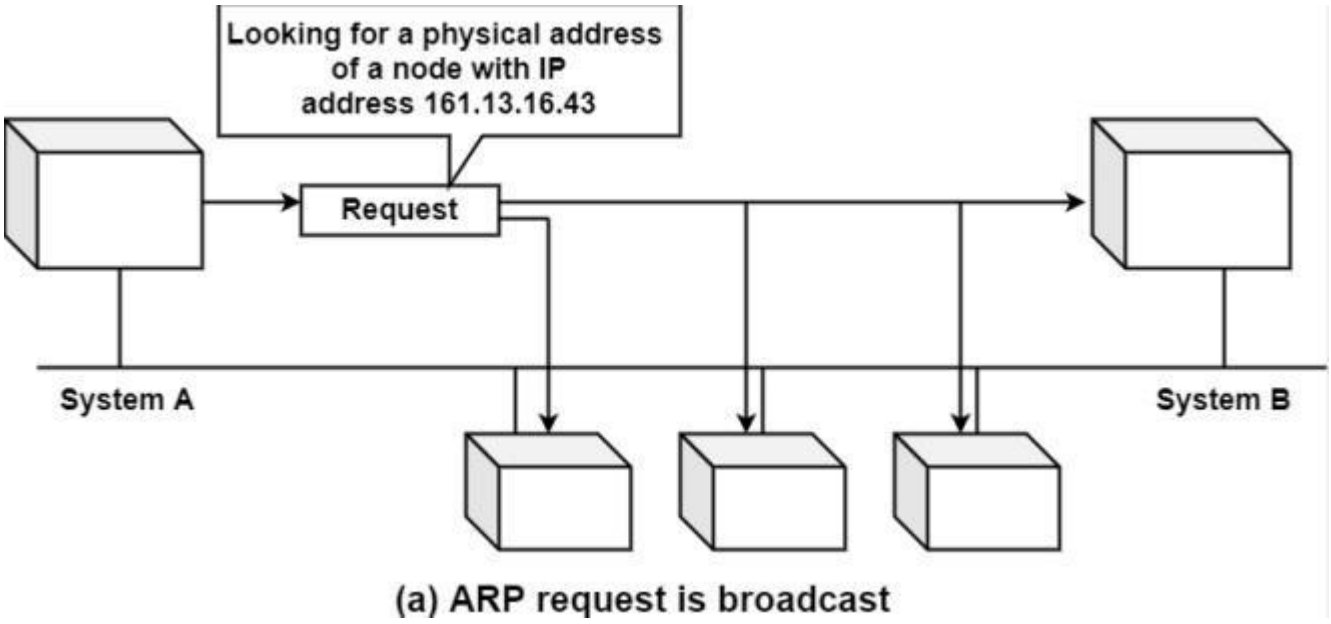
ARP relates an IP address with the physical address. On a typical physical network such as LAN, each device on a link is identified by a physical address, usually printed on the network interface card (NIC). A physical address can be changed easily when NIC on a particular machine fails.

When one host wants to communicate with another host on the network, it needs to resolve the IP address of each host to the host's hardware address.

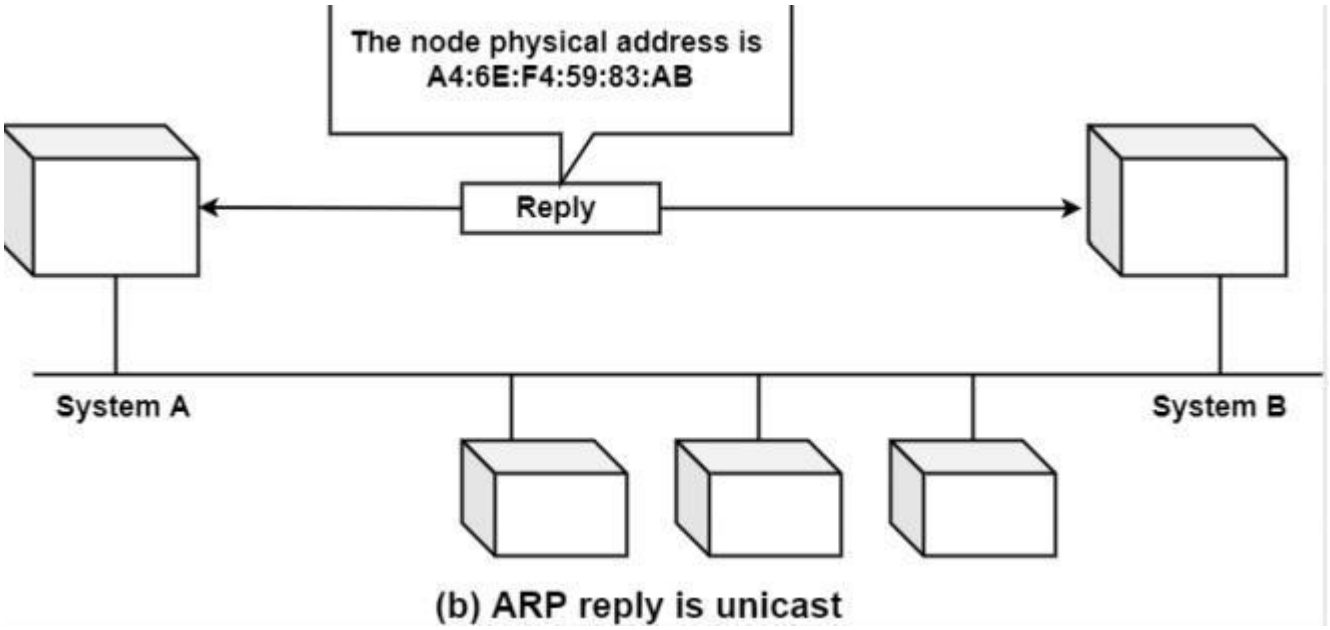
This process is as follows−

- When a host tries to interact with another host, an ARP request is initiated. If the IP address is for the local network, the source host checks its ARP cache to find out the hardware address of the destination computer.
- If the correspondence hardware address is not found, ARP broadcasts the request to all the local hosts.
- All hosts receive the broadcast and check their own IP address. If no match is discovered, the request is ignored.

- The destination host that finds the matching IP address sends an ARP reply to the source host along with its hardware address, thus establishing the communication. The ARP cache is then updated with the hardware address of the destination host.



(a) ARP request is broadcast



(b) ARP reply is unicast

RARP(REVERSE ADDRESS RESOLUTION PROTOCOL):

This protocol used to find IP Address of a host when MAC address is known.This is used to resolve IP address to MAC address .

It works similar to ARP but not in reverse process.

Subnetting:

If an organization has a large network,it is better to divide the network.The process of dividing the large network into small networks is called subnetting.

Advantages of subnetting:

1) Reduces the network traffic
2) Increases the network performance

3) Increases the security in the network

4) Maintenance of the network is easy.

Allow a single network address to span multiple physical networks is called subnet addressing.

For subnet address scheme to work, one should know which part of id address is used as subnet address.To accomplish this

Subnet mask is used.

The network administrator creates 32 bit subnet mask which contains 1's and 0's.1's indicate network address or subnet address and 0's indicate host address.

class B IP address ,the format is

Bits 8 8 8 8

Network id	Networked	Host id	Host id

Means

11111111	11111111	00000000	00000000

So the subnet mask address for CLASS B is 255.255.0.0

The default subnet mask of IP addresses

Class	IP Address format	Default subnet mask address
A	Networkid.hostid.hostid.hostid	11111111.00000000.00000000.00000000 (255.0.0)
B	Networkid.networkidid.hostid.hostid	11111111.1111111.00000000.00000000 (255.255.0.0)
C	Networkid.networkidid.networkid.hostid	11111111.1111111.11111111.00000000 (255.255.255.0)

Masking:

The process of extracting subnet address from IP address is called Masking.

To find the subnet address ,two methods are used

a) Boundary level masking

b) Non boundary level masking

   a) In this mask no is either 0 or 255.

      If mask no is 255 in the mask address then IP address is repeated in the subnet address.

      If mask no is 0(zero) in the mask address then 0 (zero) is repeated in the subnet address.

   b) In this mask no is >0 & <255 but not either 0(zero) or 255.

      For this masking ,BITWISE AND operation is performed b/w mask no and IP address.

Ex:

Find out subnet address for the following

IP Address mask address

a) 141.181.14.16 255.255.224.0

b) 200.34.22.156 255.255.255.240

c) 125.35.12.57 255.255.0.0

Solution:

a) IP Address : 141.181.14.16

Mask: 255.255.224.0

_____

Subnet address: 141.181.14&224.0

14&224

Binary equivalents of 14 ,224

14: 0000 1110

224 : 01110 0000

_____

14&224: 0000 0000

So the subnet address is 141.181.0.0

b) IP Address : 200.34.22.156

Mask: 255.255.255.240

_____

Subnet address: 200.34.22.156&240

Bitwise And operation b/w 156 &240

Binary equivalents of 156 ,240

156: 1001 1100

240 : 1111 0000

_____

156&240: 1001 0000 □ 144

So the subnet address is 200.34.22.144

c)

IP Address : 125.35.12.57

Mask: 255.255.0.0

_____

Subnet address: 125.35.0.0