DATA COMMUNICATION & COMPUTER NETWORKS (4IT4-07)

ACADEMIC SESSION (2019-20) (EVEN)

UNIT-1

PREPARED BY: Ms.Kusum Yadav

Vision of the Department

Vision:

To establish outcome based excellence in teaching, learning and commitment to support IT Industry.

Mission of the Department

Mission:

M1: To provide outcome based education.
M2: To provide fundamental & Intellectual knowledge with essential skills to meet current and future need of IT Industry across the globe.
M3: To inculcate the philosophy of continues learning, ethical values & Social Responsibility.

RTU SYLLABUS

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

SYLLABUS

II Year- IV Semester: B.Tech. (Information Technology)

4IT4-07: Data Communication and Computer Networks

Credit: 3 **Max. Marks: 150(IA:30, ETE:120)**

3L+0T+0P **End Term Exam: 3 Hours**

SN	Contents	Hours
1	**Introduction:** Objective, scope and outcome of the course.	1
2	Introductory Concepts: Network hardware, Network software, topologies, Protocols and standards, OSI model, TCP model, TCP/IP model, Physical Layer:Digital and Analog Signals, Periodic Analog Signals, Signal Transmission, Limitations of Data Rate, Digital Data Transmission, Performance Measures, Line Coding, Digital Modulation, Media and Digital Transmission System.	7
3	Data Link Layer:Error Detection and Correction,Types of Errors, Two dimensional parity check, Detection verses correction, Block Coding, Linear Block Coding, Cyclic Codes, Checksum, Standardized Polynomial Code, Error Correction Methods, Forward Error Correction, Protocols: Stop and wait, Go-back-N ARQ, Selective Repeat ARQ, Sliding window, Piggy backing, Pure ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA	8
4	Network Layer:Design issues, Routing algorithms: IPV4, IPV6, Address mapping: ARQ, RARQ, Congestion control, Unicast, Multicast, Broadcast routing protocols, Quality of Service, Internetworking.	8
5	Transport Layer: Transport service, Elements of transport protocols, User Datagram Protocol, Transmission Control Protocol, Quality of service, Leaky Bucket and Token Bucket algorithm.	8
6	Application Layer: WWW, DNS, Multimedia, Electronic mail, FTP, HTTP, SMTP, Introduction to network security	8
	Total	**40**

PROGRAM OUTCOMES

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems in IT.
2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences in IT.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations using IT.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions using IT.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations in IT.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice using IT.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development in IT.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice using IT.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings in IT.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage IT projects and in multidisciplinary environments.
12. **Life –long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological changes needed in IT.

DATA COMMUNICATION & COMPUTER NETWORKS (4IT4-07)
Academic Session (2019-20)

COURSE OUTCOMES (Cos):

Subject and Code: Data communication & Computer networks &(4IT4-07) [L/T/P - 3/0/0] Semester: IV (A+B) Faculty: Ms.Kusum Yadav

Class	:	B. Tech. – 4th Semester
External marks	:	120
Internal marks	:	30
Total marks	:	150

Course Outcomes (COs):
Graduates will be able:

CO1 Understand & explain the concept of data communication & networks ,layered architecture & their applications along with data link layer flow control techniques.

CO2 Evaluate data communication link considering elementary concepts of data link layer protocols for error detection & correction .Also Analyze of different Medium access control protocols.

CO3-Evalute different Network layer protocols and Transport layer Protocols.

CO4-Understand and design application layer protocols and internet applications such Http, SMTP, Ftp, network security, Email and DNS.

2. CO-PO mapping: MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES:

SEM	**SUBJECT WITH CODE**	**L/ P/ T**	**CO**	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
IV	**Data communication & Computer networks (4IT4-07)**	3/ 0/ 0	CO1 Understand & explain the concept of data communication & networks ,layered architecture & their applications along with data link layer flow control techniques.	H	H	M	L	L			L	L	M	L
			CO2 Evaluate data communication link considering elementary concepts of data link layer protocols for error detection & correction .Also Analyze of different Medium access control protocols.	H	H	H	L	L					M	L
			CO3-Evalute different Network layer protocols and Transport layer Protocols	H	H	H	L	L					M	L
			CO4-Understand and design application layer protocols and internet applications such Http, SMTP, Ftp, network security, Email and DNS	H	H	L	M	M			L	H	M	H

Data Communications

The fundamental purpose of a communication system is the exchange of data between two communicable devices. So to make the communication possible, the devices need to be connected through some form of transmission medium such as a wire cable.

For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs).

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

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

2. **Accuracy:** The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

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

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

Components of a data communications system

A data communications system has five components:

Message: The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

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

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

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.

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.

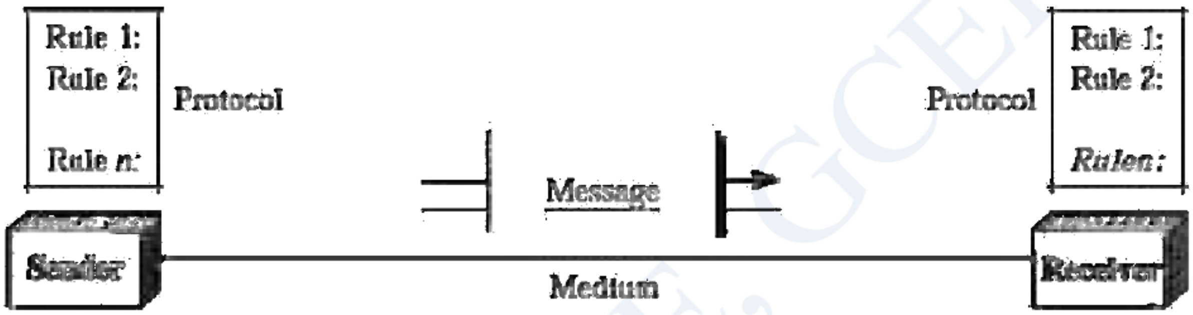


Fig. 1.1 components of data communication

Data Flow

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

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.

Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.

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.

The half-duplex mode is like a one-lane road with traffic allowed in both directions. When cars are traveling in one direction, cars going the other way must wait**.**
In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. **Walkie-talkies and CB(citizens band) radios** are both half-duplex systems.

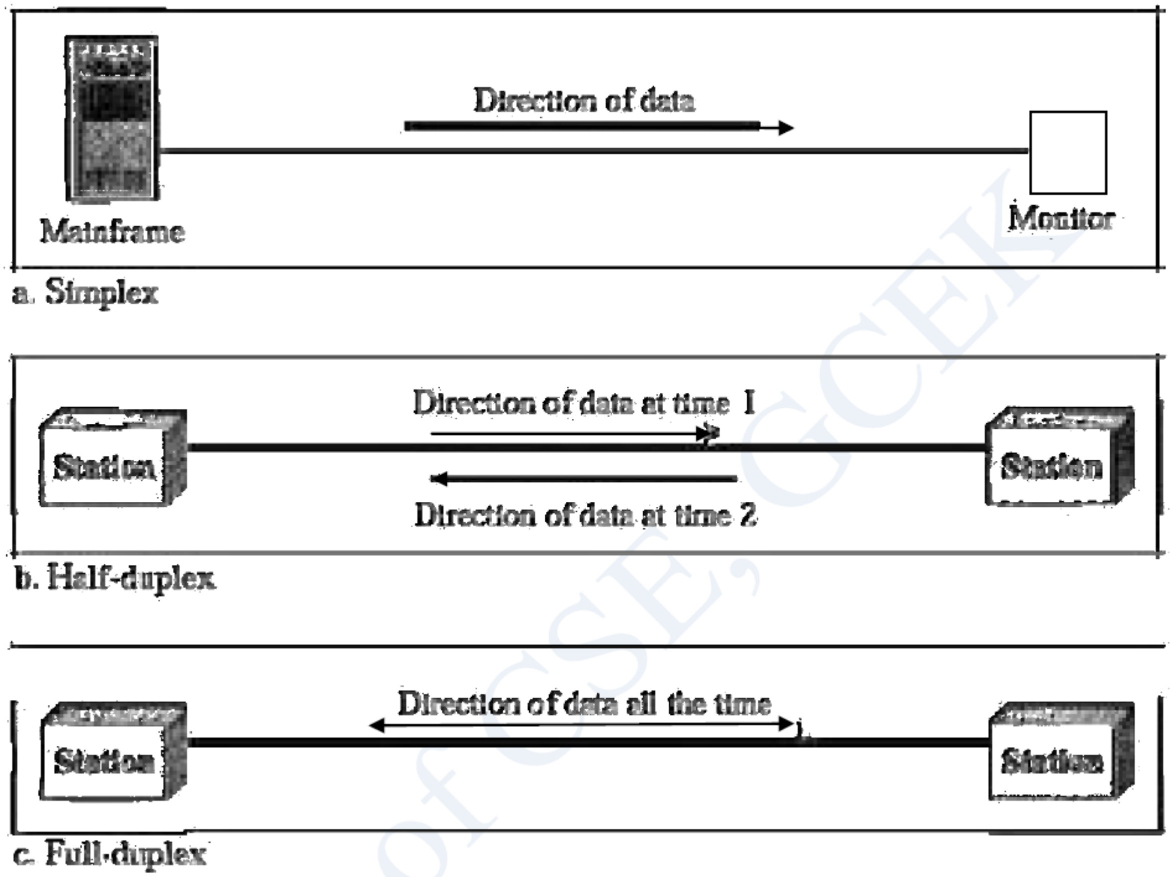


Fig. 1.2 Data flow (simplex,half-duplex, and full-duplex)

Full Duplex

In full-duplex mode, both stations can transmit and receive simultaneously.

The full-duplex mode is like a two way street with traffic flowing in both directions at the same time.

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.

Network

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.

Distributed Processing

Most networks use distributed processing, in which a task is divided among multiple computers. Instead of one single large machine being responsible for all aspects of a process, separate computer (usually a personal computer or workstation) handle a subset.

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 transmit time and response time.

Transmit 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 throughputs and less delay.

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.

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.

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. For visualization purposes, it is simplest to imagine any link as a line drawn between two points.

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.

Ex: changing television channels by infrared remote control.

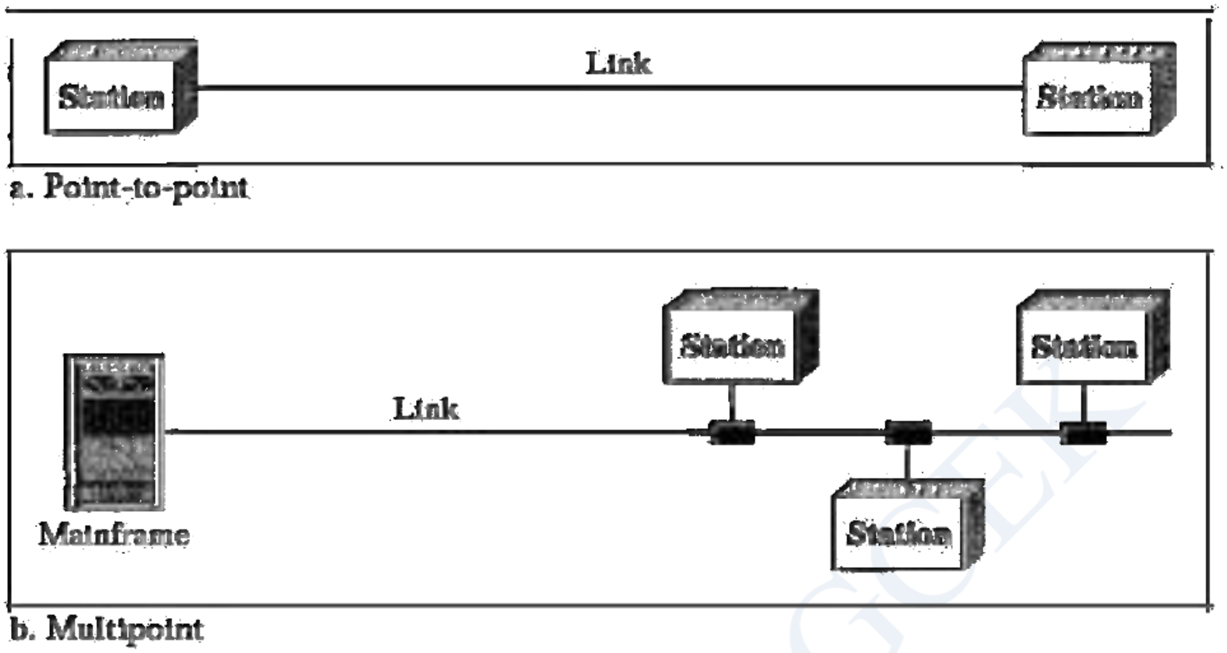


Fig. 1.3 Types of connections: point-to-point and multipoint

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.

NETWORK HARDWARE:

Hub:

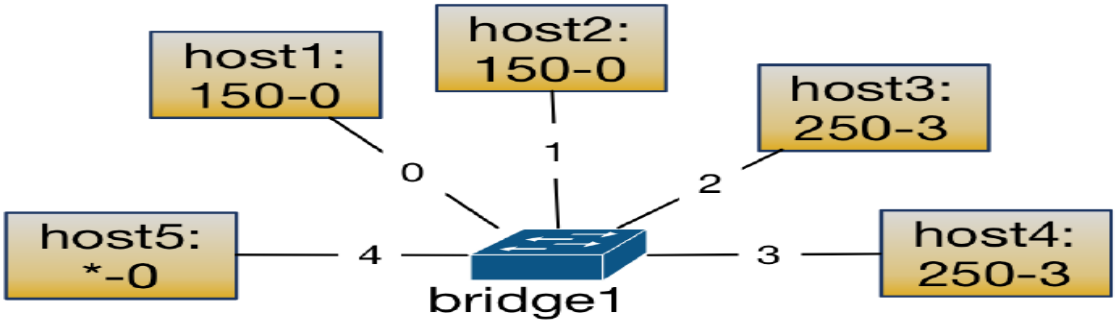
- It is a pure electronic device, not associated with any s/w.
- It is broadcasting and a passive device.
- Physical layer device.
- It puts an extra traffic load on the network. ☹

Switch:

- Data link layer and physical layer device.
- It is a pure electronic device but associated with s/w.
- Performance is good and network traffic is low.
- It maintains a lookup table to keep track on system.
- Cost is high.

Bridge:

- It is used to connect multiple subnets or LANs in a network.
- Filtering and forwarding is the design criteria for a bridge.
- Bridge also uses lookup table to keep track of LANs.
- Its principle is based on physical addressing system.
- Work on physical and Data link layer.
- Not used beyond the network.



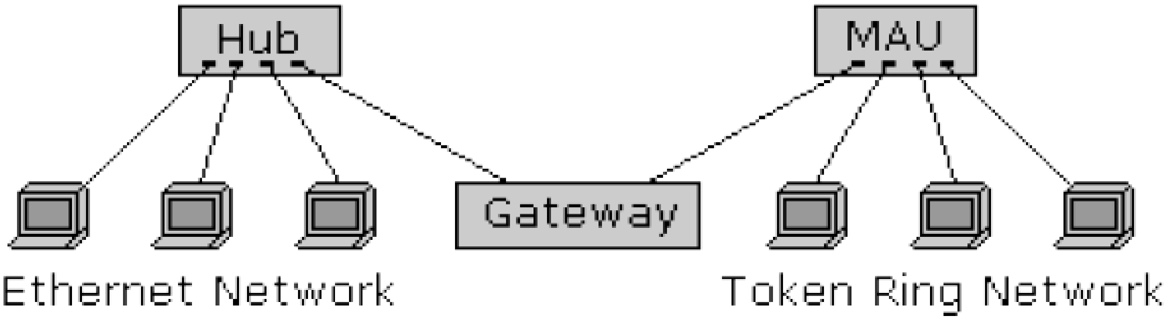
Router:

- It is a WAN device.
- Principle is based on logical addressing system.
- Used to connect 2 or more different similar networks (TCP/IP).
- It is a sophisticated device which runs routing algorithms.

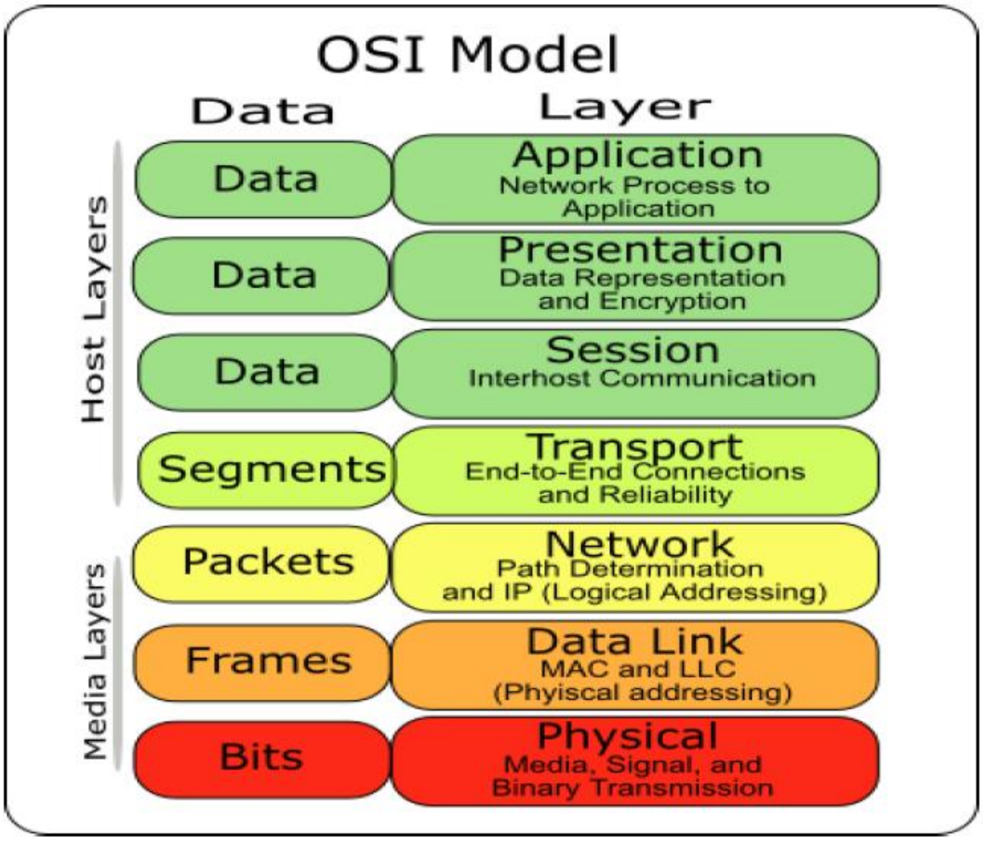
Brouter: Bridge + Router **Gateway:**

- It is used to connect two different dissimilar networks.
- It is also a WAN device and operating principle is based on logical addressing system.

- Also known as protocol converter.



OSI LAYER :


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

Introduction: - Types of Computer Networks, Network Software - Protocol Hierarchies, Connection oriented and Connection less hierarchies, Reference Models - ISO-OSI Reference Model, TCP/IP Reference Model – Comparison of OSI and TCP/IP reference models.

Physical Layer: - Guided Transmission Media– Twisted Pair, Coaxial and Fiber Optics, WirelessTransmission- Radio and Microwave transmission, Communication Satellites – GEO,MEO, LEO.Comparison of Network hardware - Repeaters, Routers, Bridges, Gateways, Hub and Cable Modem

Advantages of Network

- **Speed**. Sharing and transferring files within Networks are very rapid. Thus saving time, while maintaining the integrity of the file.
- **Cost**. Individually licensed copies of many popular software programs can be costly. Networkable versions are available at considerable savings. Shared programs, on a network allows for easier upgrading of the program on one single file server, instead of upgrading individual workstations.

- **Security**. Sensitive files and programs on a network are passwords protected or designated as "copy inhibit," so that you do not have to worry about illegal copying of programs.
- **Centralized Software Management**. Software can be loaded on one computer (the file server) eliminating that need to spend time and energy installing updates and tracking files on independent computers throughout the building.
- **Resource Sharing.** Resources such as, printers, fax machines and modems can be shared.
- **Electronic Mail.** E-mail aids in personal and professional communication.
- **Flexible Access.** Access their files from computers throughout the firm.
- **Workgroup Computing**. Workgroup software (such as Microsoft BackOffice) allows many users to work on a document or project concurrently.

Categories (Types) and connections of networks

Types of Network

Networks are discussed in terms of their
(i) transmission technology/type of connection and their
(ii) scale / size

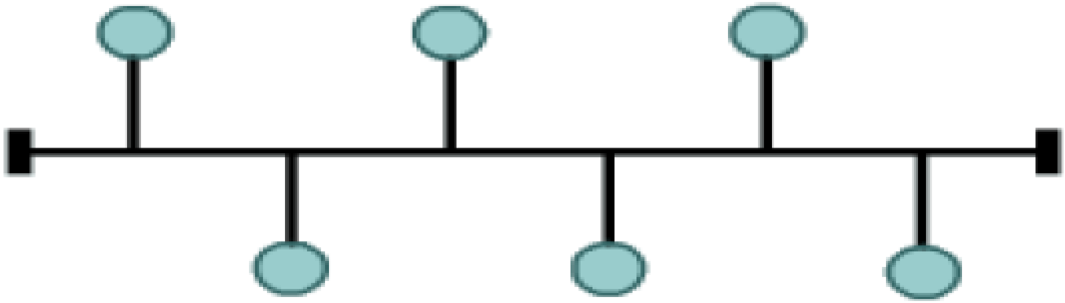
Types of Network Based on the *Transmission Technology/*Type of Connection
1 .Broadcast Networks (Multi point)
2 .Point-to-point Networks

1 .Broadcast Networks (Multi point)

Single communication channel shared by all computers

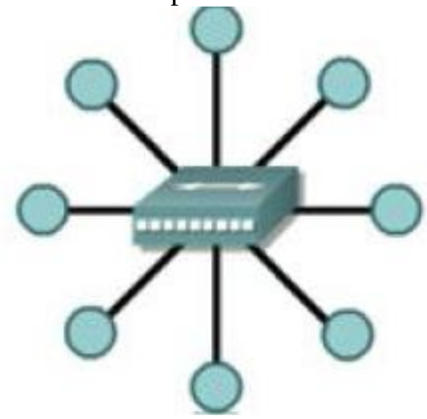
- Packets (short messages) sent by a computer contains an address specifying the destination computer

- All computers connected to the network receive the packet. The destination computer processes the packet while all other computers discard the packet
- In multipoint (multi drop) connection the capacity of channel is shared either temporary or spatially



2. Point-to-Point Networks

- Consists of many connections between individual pairs of computers
- Sending a packet between two computer may involve the packet being forwarded by intermediate computers
- Multiple routes of different lengths possible, therefore routing algorithms are important



Types of Network Based on the SCALE / SIZE

- **PAN (Personal Area Network)**
- **LAN (Local Area Network),**
- **MAN (Metropolitan area network)**
- **WAN (Wide Area Network)**
- **PAN (Personal Area Network)**

A personal area network (PAN) is a computer network used for data transmission amongst devices such as computers, telephones, tablets and personal digital assistants.

A wireless personal area network (WPAN) is a low-powered PAN carried over a short distance wireless network technology such as Bluetooth, ZigBee etc.

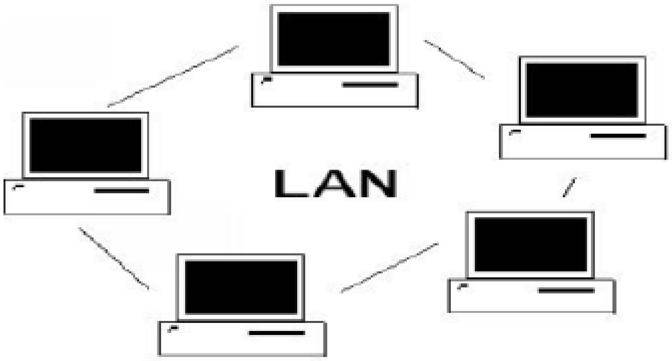
LAN (Local area networks)

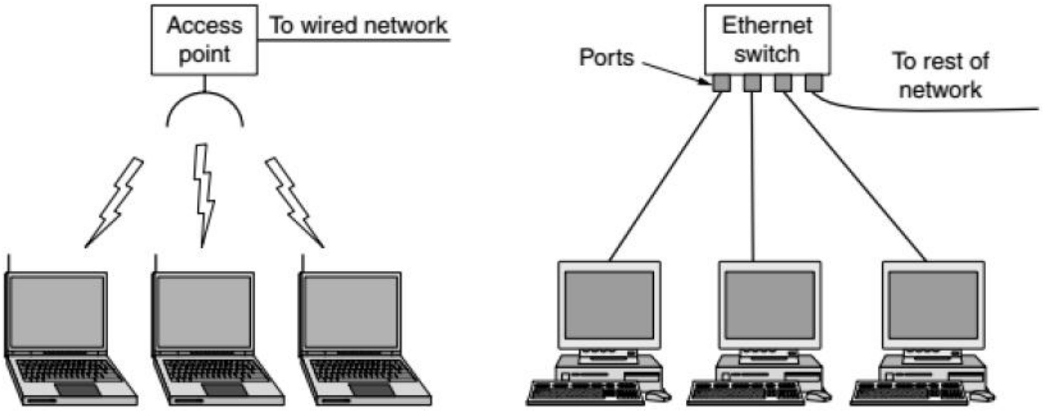
Generally called **LANs**, are privately-owned **networks within a single building or campus of**

up to a few kilometers in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information.
LAN configuration consists of:

- A file server
- A workstation
- Cables

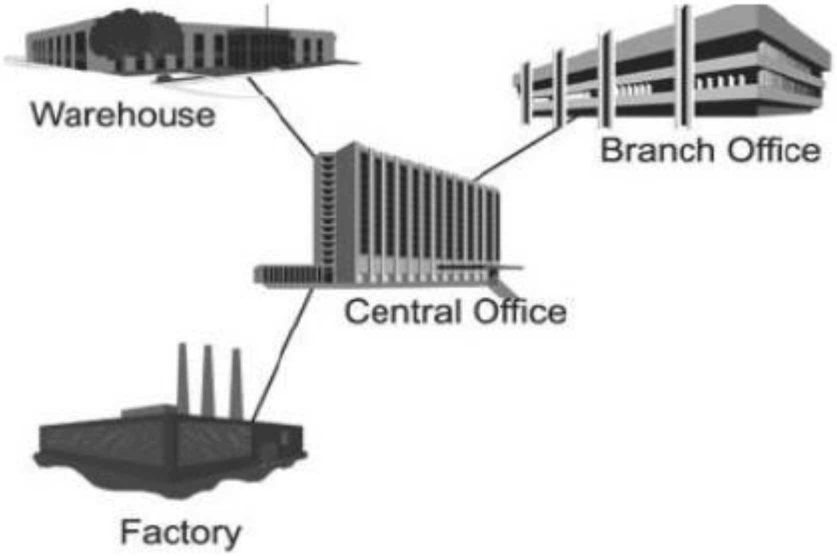
The topology of many wired LANs is built from point-to-point links. IEEE802.3, popularly called **Ethernet**



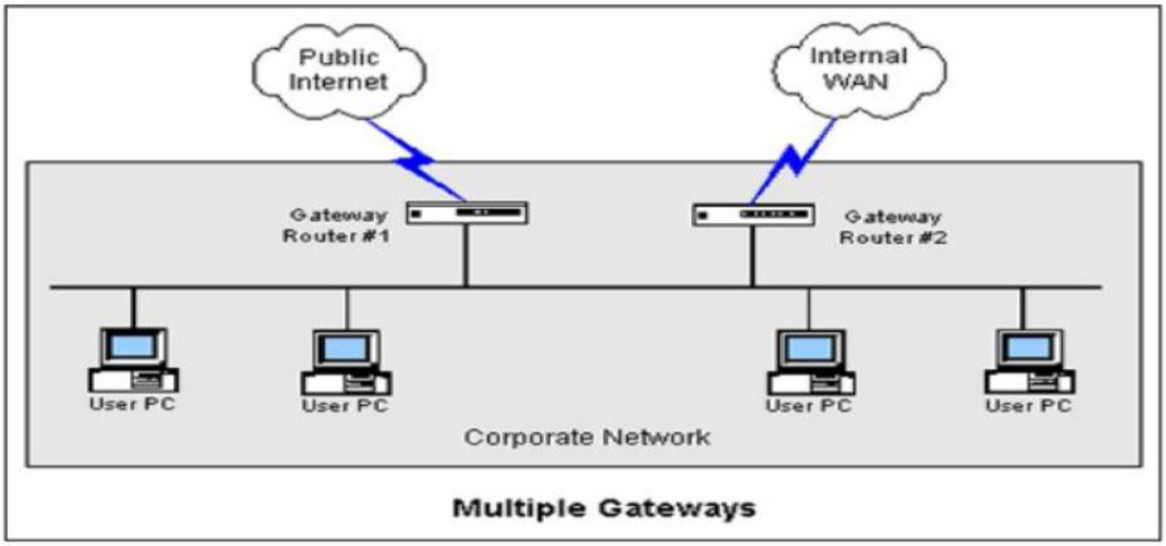


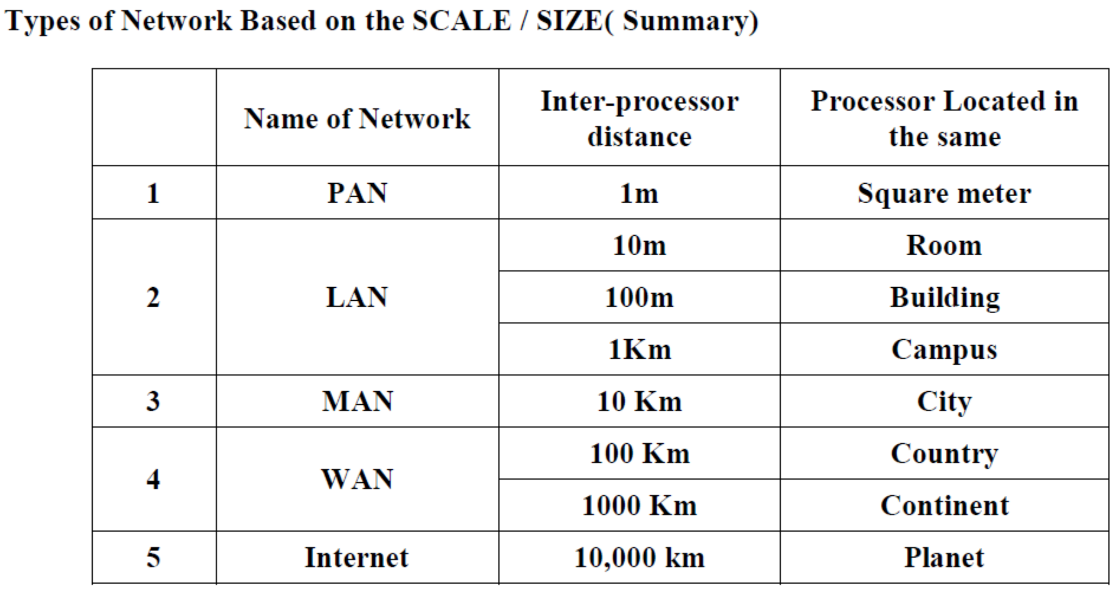
Wireless and wired LANs. (a) 802.11. (b) Switched Ethernet.

MAN (Metropolitan area network):
A metropolitan area network (MAN) is a large
computer network that usually spans a city or a large campus. A MANusually interconnects a number of local area networks (LANs) using a high-capacity backbone technology, such as fiber-optical links, and provides uplink services to wide area networks and the Internet Eg. A metropolitan area network based on cable TV.



WAN (Wide Area Network)
A **WAN** spans a **large geographic area, such as a state, provinceor country.** WANs often connect multiple smaller networks, such as local area networks (LANs)or metro area networks (MANs). Eg.ISP network.





Types of Network Based on the SCALE / SIZE(Summary)

	Name of Network	Inter-processor distance	Processor Located in the same
1	PAN	1m	Square meter
2	LAN	10m	Room
		100m	Building
		1Km	Campus
3	MAN	10 Km	City
4	WAN	100 Km	Country
		1000 Km	Continent
5	Internet	10,000 km	Planet

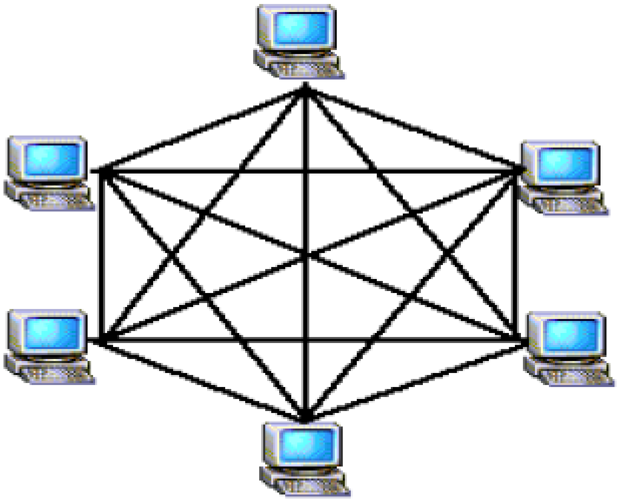
Network Topologies

Topology refers to the way a network is laid out either physically or logically. Two or more devices connect to a link; two or more links form a topology. It is the geographical representation of the relationship of all the links and linking devices to each other.

1. MeshTopology
2. StarTopology
3. Tree(Extended Star) Topology
4. BusTopology
5. Ring Topology
6. HybridTopology

1. Mesh Topology:

Here every device has a dedicated point to point link to every other device. A fully connected mesh can have n(n-1)/2 physical channels to link n devices. It must have n-1 IO ports.



Advantages:

1. They use dedicated links so each link can only carry its own data load. So traffic problem can be avoided.
2. It is robust. If any one link get damaged it cannot affect others
3. It gives privacy and security
4. Fault identification and fault isolation are easy.

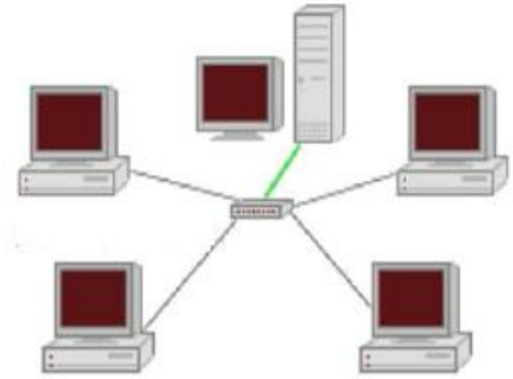
Disadvantages:

1. The amount of cabling and the number IO ports required are very large. Since every device is connected to each other devices through dedicated links.
2. The sheer bulk of wiring is larger then the available space
3. Hardware required to connect each device is highly expensive.

2. STAR TOPOLOGY:

Here each device has a dedicated link to the central 'hub'. There is no direct traffic between devices. The transmission are occurred only through the central controller namely hub.

Advantages:



1. Less expensive then mesh since each device is connected only to the hub.
2. Installation and configuration are easy.
3. Less cabling is need then mesh.
4. Robustness.
5. Easy to fault identification & isolation.

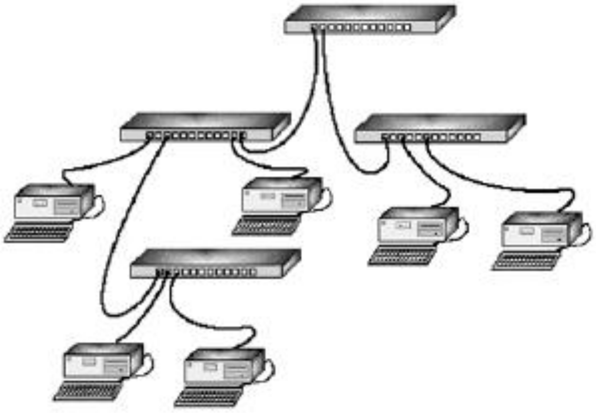
Disadvantages:

1. Even it requires less cabling then mesh when compared with other topologies it still large.

3. TREE TOPOLOGY (Extended Star Topology)

It is a variation of star. Instead of all devices connected to a central hub here most of the devices are connected to a secondary hub that in turn connected with central hub. The central hub is an active hub. An active hub contains a repeater, which regenerate the received bit pattern before sending.

The secondary hub may be active or passive. A passive hub means it just precedes a physical



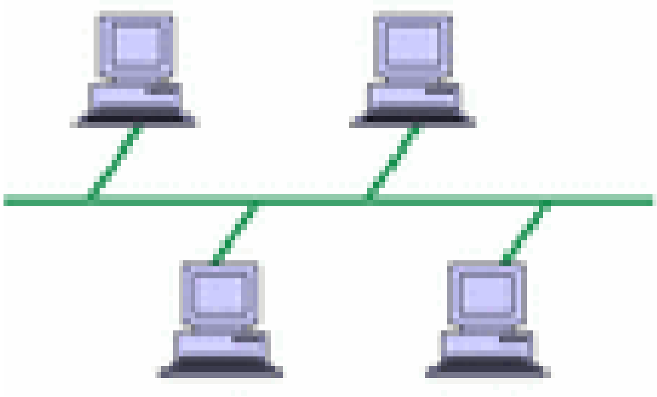
connection only

Advantages:

1. Can connect more than star.
2. The distance can be increased.
3. Can isolate and prioritize communication between different computers.

4. BUS TOPOLOGY:

A bus topology is multipoint. Here one long cable is act as a backbone to link all the devices are connected to the backbone by drop lines and taps. A drop line is the connection between the devices and the cable. A tap is the splice into the main cable or puncture the



cover.

Advantages:

1. Ease of installation.
2. Less cabling

Disadvantages:

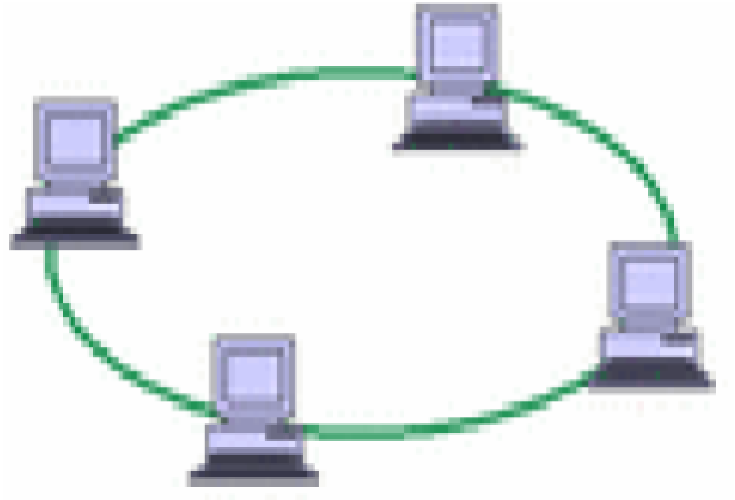
1. Difficult reconfiguration and fault isolation.
2. Difficult to add new devices.
3. Signal reflection at top can degradation in quality
4. If any fault in backbone can stops all transmission

5. RING TOPOLOGY

Each node is connected to exactly two other nodes, forming a ring. Can be visualized as a circular configuration. Requires at least three nodes

Advantages:

1. Easy to install.
2. Easy to reconfigure.



3. Fault identification is easy.

Disadvantages:

1. Unidirectional traffic.
2. Break in a single ring can break entire network

6. HYBRID TOPOLOGY

A combination of any two or more network topologies.

NETWORK SOFTWARE

Protocol Hierarchies

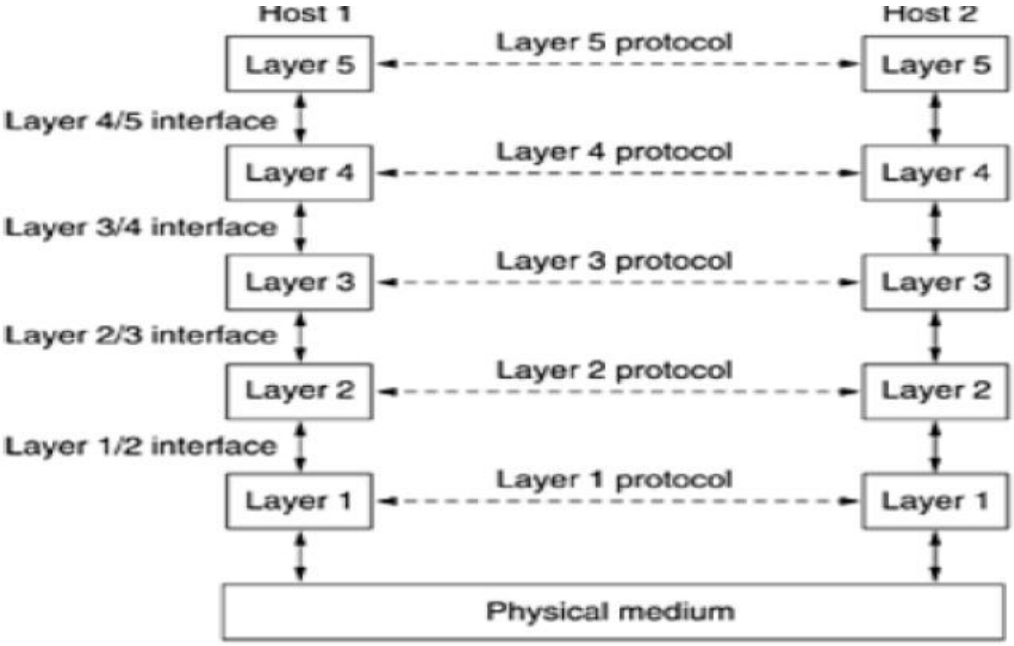
Layering

- To reduce their design complexity, most networks are organized as a stack of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network.
- The purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented.
- Layer n on one machine carries on a conversation with layer n on another machine. The rules and conventions used in this conversation are collectively known as the layer n protocol.

Layering provides two nice features.

- It decomposes the problem of building a network into more manageable components. Rather than implementing a monolithic piece of software that does everything implement several layers, each of which solves one part of the problem
- It provides more modular design. To add some new service, it is enough to modify the functionality at one layer, reusing the functions provided at all the other layers.

A five-layer network is illustrated in figure below. The entities comprising the corresponding layers on different machines are called peers. The peers may be processes, hardware devices, or even human beings. In other words, it is the peers that communicate by using the protocol.



Between each pair of adjacent layers is an **interface**. The interface defines which primitive operations and services the lower layer makes available to the upper one.No data are directly transferred from layer n on one machine to layer n on another machine. Instead, each layer passes data and control information to the layer immediately below it, until the lowest layer is reached. Below layer 1 is the physical medium through which actual communication occurs. A set of layers and protocols is called **network** architecture. A list of protocols used by a certain system, one protocol per layer, is called a **protocol stack**.

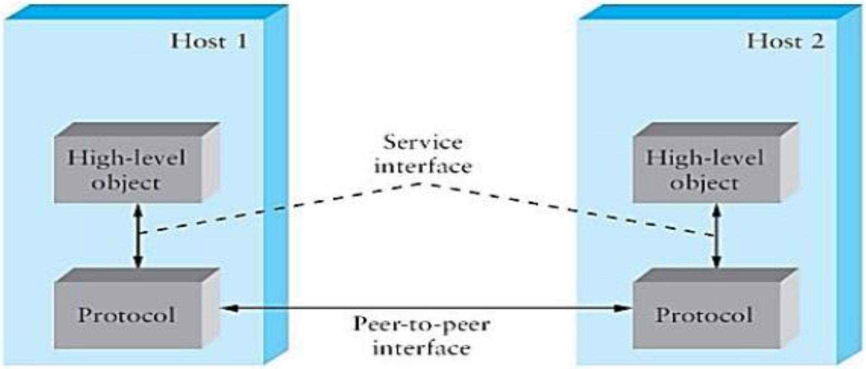
Protocols

A protocol is an agreement between the communicating parties on how communication is to Proceed. A protocol is a set of rules that governs data communication. It defines what is Communicated, how it is communicated, and when it is communicated. The key elements of a Protocol is syntax, semantics and timing.

Each protocol defines two different interfaces.

- Service interface - to the other objects on the same computer that want to use its Communication services. This service interface defines the operations that local objects can perform on the protocol.
- Peer interface - to its counterpart (peer) on another machine. It also defines the form and meaning of messages exchanged between protocol peers to implement the

communication service.



Encapsulation

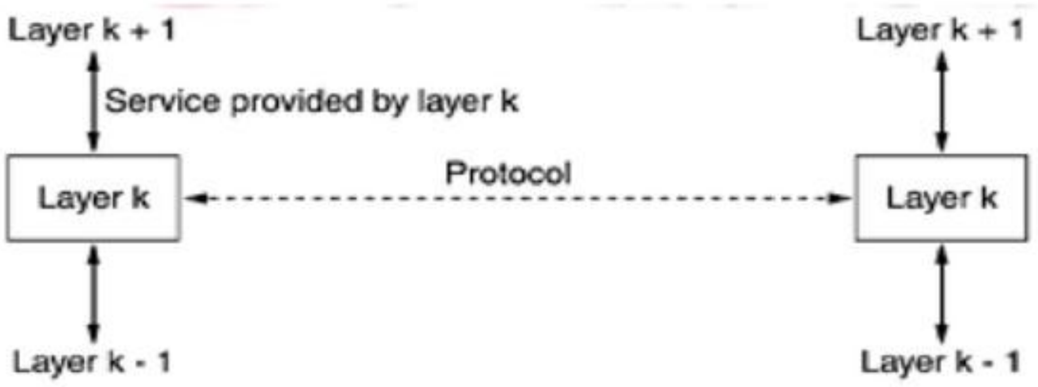
Control information must be added with the data to instruct the peer how to handle with the received message. It will be added into the header or trailer.

- Header - Small data structure from few bytes to few kilobytes attached to the front of message.
- Trailer – Information will be added at the end of the message
- Payload or message body – Data send by the program In this case data is encapsulated with new message created by protocol at each level.

The Relationship of Services to Protocols

A service is a set of primitives (operations) that a layer provides to the layer above it. The service defines what operations the layer is prepared to perform on behalf of its users, but it says nothing at all about how these operations are implemented. A service relates to an interface between two layers, with the lower layer being the service provider and the upper layer being the service user.

A protocol, is a set of rules governing the format and meaning of the packets, or messages that are exchanged by the peer entities within a layer. Entities use protocols to implement their service definitions. They are free to change their protocols at will, provided they do not change the service visible to their users. In this way, the service and the protocol are completely decoupled



.

The OSI Model

An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture.

The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.

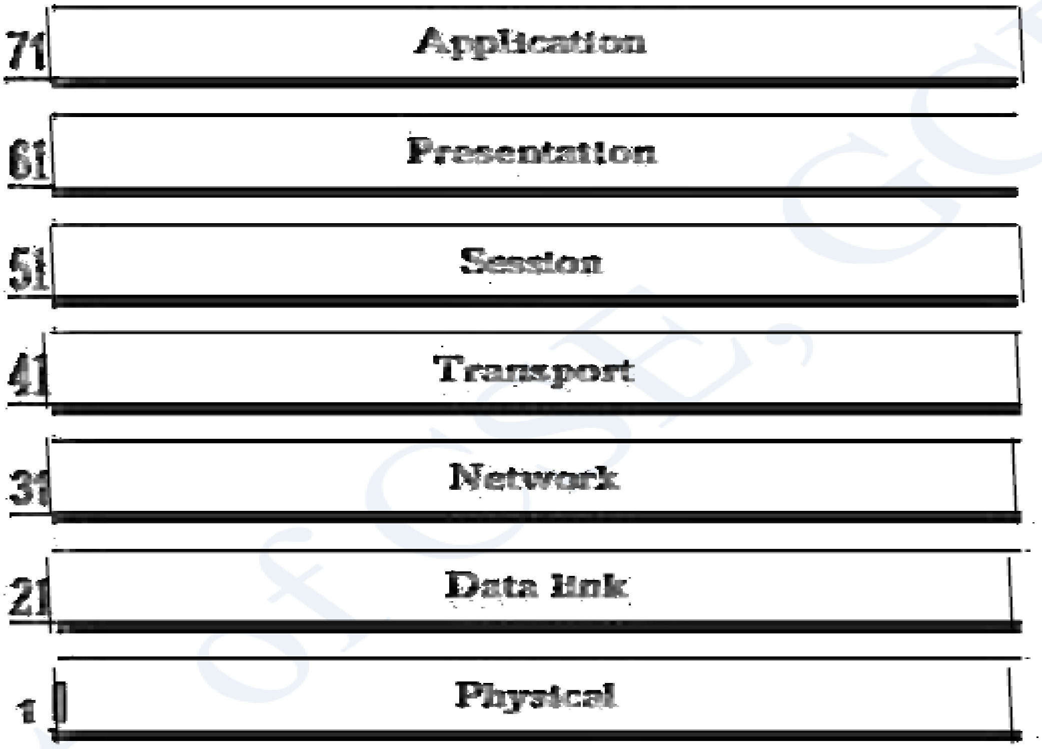


Fig. 2.1: OSI layers

Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium.

Responsible for movements of individual bits from one hop (node) to the next.

The physical layer is also concerned with the following:

- **Physical characteristics of interfaces and medium:** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- **Representation of bits:** The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how Os and I s are changed to signals).

- **Data rate:** The transmission rate-the number of bits sent each second-is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
- **Synchronization of bits:** The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.
- **Line configuration:** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
- **Physical topology:** The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).
- **Transmission mode:** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer).

Responsible for moving frames from one hop (node) to the next.

Other responsibilities of the data link layer include the following:

- **Framing:** The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- **Physical addressing:** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.
- **Flow control:** If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- **Error control:** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.
- **Access control:** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Network Layer

The network layer is **responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).** Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

Other responsibilities of the network layer include the following:

- **Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes th logical addresses of the sender and receiver. We discuss logical addresses later in this chapter.
- **Routing.** When independent networks or links are connected to create internetworks (network of networks) or a large network, the connecting devices (called routers or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

Transport Layer

The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets.

Other responsibilities of the transport layer include the following:

- **Service-point addressing:** Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.
- **Segmentation and reassembly:** A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.
- **Connection control:** The transport layer can be either connectionless or connection oriented.A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- **Flow control:** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- **Error control:** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

Session Layer

The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction among communicating systems. The session layer is responsible for dialog control and synchronization.

Also used to establish, manage and terminate sessions.

Specific responsibilities of the session layer include the following:

- **Dialog control:** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half duplex (one way at a time) or full-duplex (two ways at a time) mode.
- **Synchronization:** The session layer allows a process to add checkpoints, or synchronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent.

Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.

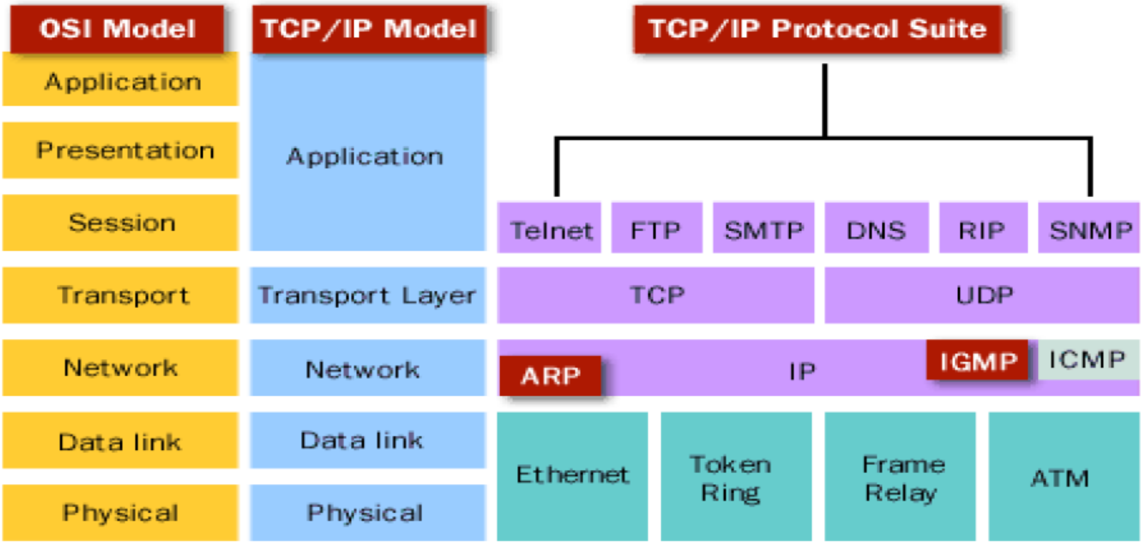
Specific responsibilities of the presentation layer include the following:

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

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services, ex. FTAM used for file transfer and management, X.400 for message handling and X.500 for directory services.

TCP/IP Protocol Suites:



ARP: Address resolution protocol
IP: Internet protocol
IGMP: Internet group message protocol
ICMP: Internet control message protocol
TCP: Transmission control protocol
UDP: User datagram protocol
Telnet: Terminal network
FTP: File transfer protocol
SMTP: Simple mail transfer protocol
DNS: Domain name system
RIP: Routing information protocol
SNMP: Simple network management protocol

1. The Host to Network Layer: Below the internet layer is great void. The TCP/IP reference model does not really say such about what happen here, except to point out that the host has connect to the network using some protocol so it can transmit IP packets over it. This protocol is not specified and varies from host to host and network to network.

2. Internet layer:
It is a connectionless internetwork layer forming a base for a packet-switching network. Its job is to allow hosts to insert packets into any network and have them to deliver independently to the Destination. They may appear in a different order than they were sent in each case it is job of higher layers to rearrange them in order to deliver them to proper destination.TCP/IP internet layer is very similar in functionality to the OSI network layer. Packet routing is very essential task in order to avoid congestion. For these reason it is say that TCP/IP internet layer perform same function as that of OSI network layer.
The internet layer defines an official packet format and protocol called IP (Internet Protocol) and its provides

- Best-effort delivery
- No error checking
- No tracking
- IP is a host-to-host protocol.

3. Transport layer:
In the TCP/IP model, the layer above the internet layer is known as transport layer. It is developed to permit entities on the source and destination hosts to carry on a conversation. It specifies 2 end-to-end protocols
i) TCP (Transmission Control Protocol)
ii) UDP (User Datagram Protocol)

TCP
It is a reliable connection-oriented protocol that permits a byte stream originating on one machine to be transported without error on any machine in the internet. It divides the incoming byte stream into discrete message and passes each one onto the internet layer. At the destination, the receiving TCP process collects the received message into the output stream. TCP deals with flow control to make sure a fast sender cannot swamp a slow receiver with more message than it can handle.

UDP
It is an unreliable, connectionless protocol for applications that do not want TCP's sequencing on flow control and wish to offer their own. It is also used for client-server type request-reply queries and applications in which prompt delivery is more important than accurate delivery such as transmitting speech or video.

4. Application Layer:
In TCP/IP model, session or presentation layer are not present. Application layer is present on the top of the Transport layer. It includes all the higher-level protocols which are virtual terminal (TELNET), file transfer (FTP) and electronic mail (SMTP). The virtual terminal protocol permits a user on one machine to log into a distant machine and work there. The file transfer protocol offers a way to move data efficiently from one machine to another. Electronic mail was used for file transfer purpose but later a specialized protocol was developed for it.

The Application Layer defines following protocols
i) File Transfer Protocol (FTP)
It was designed to permit reliable transfer of files over different platforms. At the transport layer to ensure reliability, FTP uses TCP. FTP offers simple commands and makes the differences in storage methods across networks transparent to the user. The FTP client is able to interact with any FTP server; therefore the FTP server must also be able to interact with any FTP client. FTP does not offer a user interface, but it does offer an application program interface for file transfer. The client part of the protocol is called as FTP and the server part of the protocol is known as FTPd. The suffix "d" means Daemon this is a legacy from Unix computing where a daemon is a piece of software running on a server that offers a service.

ii) Hyper Text Transfer Protocol

HTTP permits applications such as browsers to upload and download web pages. It makes use of TCP at the transport layer again to check reliability. HTTP is a connectionless protocol that sends a request, receives a response and then disconnects the connection. HTTP delivers HTML documents plus all of the other components supported within HTML such as JavaScript, Visual script and applets.

iii) Simple Mail Transfer Protocol

By using TCP, SMTP sends email to other computers that support the TCP/IP protocol suite. SMTP provides extension to the local mail services that existed in the early years of LANs. It supervises the email sending from the local mail host to a remote mail host. It is not reliable for accepting mail from local users or distributing received mail to recipients this is the responsibility of the local mail system. SMTP makes use of TCP to establish a connection to the remote mail host, the mail is sent, any waiting mail is requested and then the connection is disconnected. It can also return a forwarding address if the intended recipient no longer receives email at that destination. To enable mail to be delivered across differing systems, a mail gateway is used.

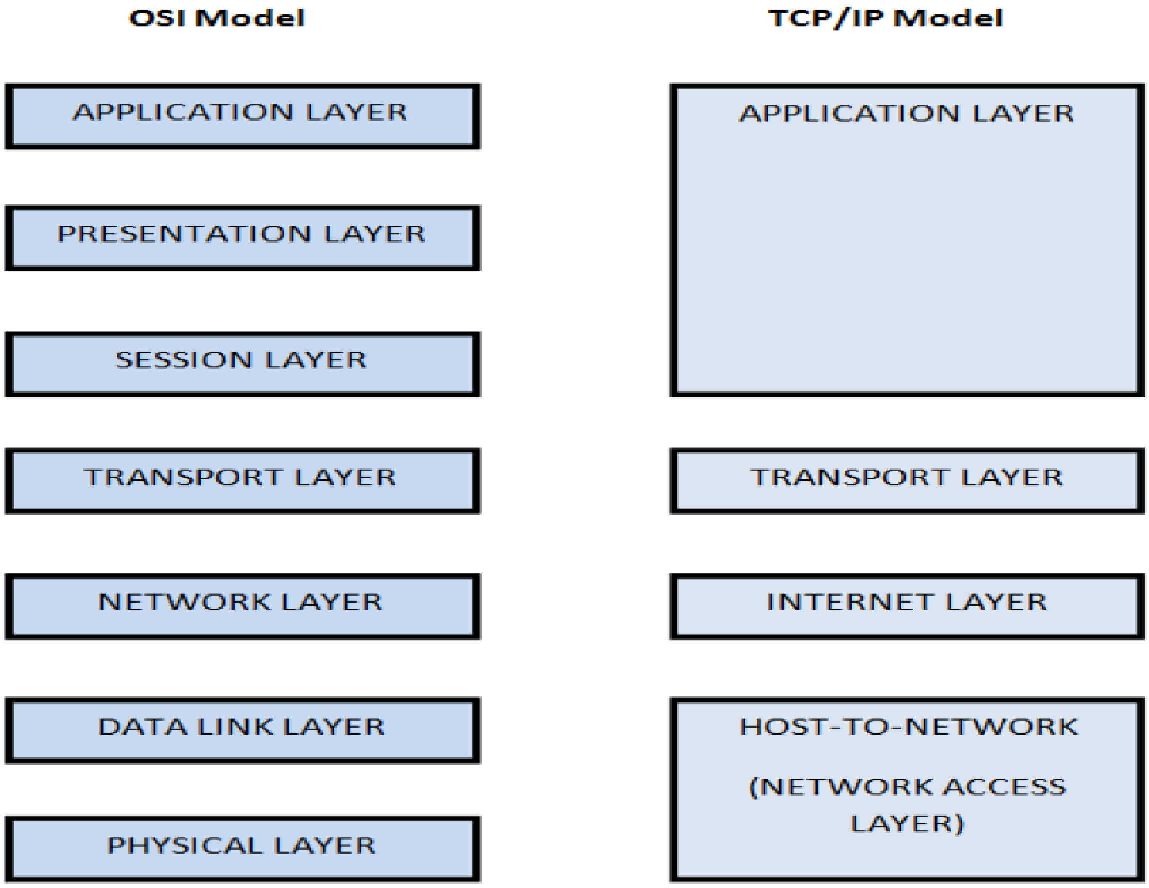
iv) Simple Network Management Protocol

For the transport of network management information, SNMP is used as standardized protocol. Managed network devices can be cross examined by a computer running to return details about their status and level of activity. Observing software can also trigger alarms if certain performance criteria drop below acceptable restrictions. At the transport layer SNMP protocol uses UDP. The use of UDP results in decreasing network traffic overheads.

Comparison of OSI Reference Model and TCP/IP Reference Model

Following are some major differences between OSI Reference Model and TCP/IP Reference Model, with diagrammatic comparison below. OSI(Open System Interconnection)	TCP/IP(Transmission Control Protocol / Internet Protocol)
1. OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user.	1. TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network.
2. In OSI model the transport layer guarantees the delivery of packets.	2. In TCP/IP model the transport layer does not guarantees delivery of packets. Still the TCP/IP model is more reliable.
3. Follows vertical approach.	3. Follows horizontal approach.

4. OSI model has a separate Presentation layer and Session layer.	4. TCP/IP does not have a separate Presentation layer or Session layer.
5. OSI is a reference model around which the networks are built. Generally it is used as a guidance tool.	5. TCP/IP model is, in a way implementation of the OSI model.
6. Network layer of OSI model provides both connection oriented and connectionless service.	6. The Network layer in TCP/IP model provides connectionless service.
7. OSI model has a problem of fitting the protocols into the model.	7. TCP/IP model does not fit any protocol
8. Protocols are hidden in OSI model and are easily replaced as the technology changes.	8. In TCP/IP replacing protocol is not easy.
9. OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them. It is protocol independent.	9. In TCP/IP, services, interfaces and protocols are not clearly separated. It is also protocol dependent.
10. It has 7 layers	10. It has 4 layers



ANALOG AND DIGITAL SIGNAL

One of the major functions of the physical layer is to move data in the form of electromagnetic signals across a transmission medium.

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

Analog and Digital Data

Data can be analog or digital. The term **analog data** refers to information that is continuous; **digital data** refers to information that has discrete states. For example, an analog clock that has hour, minute, and second hands gives information in a continuous form; the movements of the hands are continuous. On the other hand, a digital clock that reports the hours and the minutes will change suddenly from 8:05 to 8:06. Analog data, such as the sounds made by a human voice, take on continuous values. When someone speaks, an analog wave is created in the air. This can be captured by a microphone and converted to an analog signal or sampled and converted to a digital signal. Digital data take on discrete values. For example, data are stored in computer memory in the form of Os and 1s. They can be converted to a digital signal or modulated into an analog signal for transmission across a medium.

Analog and Digital Signals

Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time.

A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and 0.

Periodic and Non periodic Signals

Both analog and digital signals can take one of two forms: periodic or non periodic.A periodic signal completes a pattern within a measurable time frame, called a period, and repeats that pattern over subsequent identical periods. The completion of one full pattern is called a cycle.

A non periodic signal changes without exhibiting a pattern or cycle that repeats over time. Both analog and digital signals can be periodic or non periodic. In data communications, we commonly use periodic analog signals (because they need less bandwidth) and non periodic digital signals (because they can represent variation in data).

Periodic Analog Signals

Periodic analog signals can be classified as simple or composite.

- A simple periodic analog signal, a sine wave, cannot be decomposed into simpler signals.
- A composite periodic analog signal is composed of multiple sine waves.

Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. A sine wave can be represented by three parameters: the peak amplitude, the frequency, and the phase.
These three parameters fully describe a sine wave.

Peak Amplitude

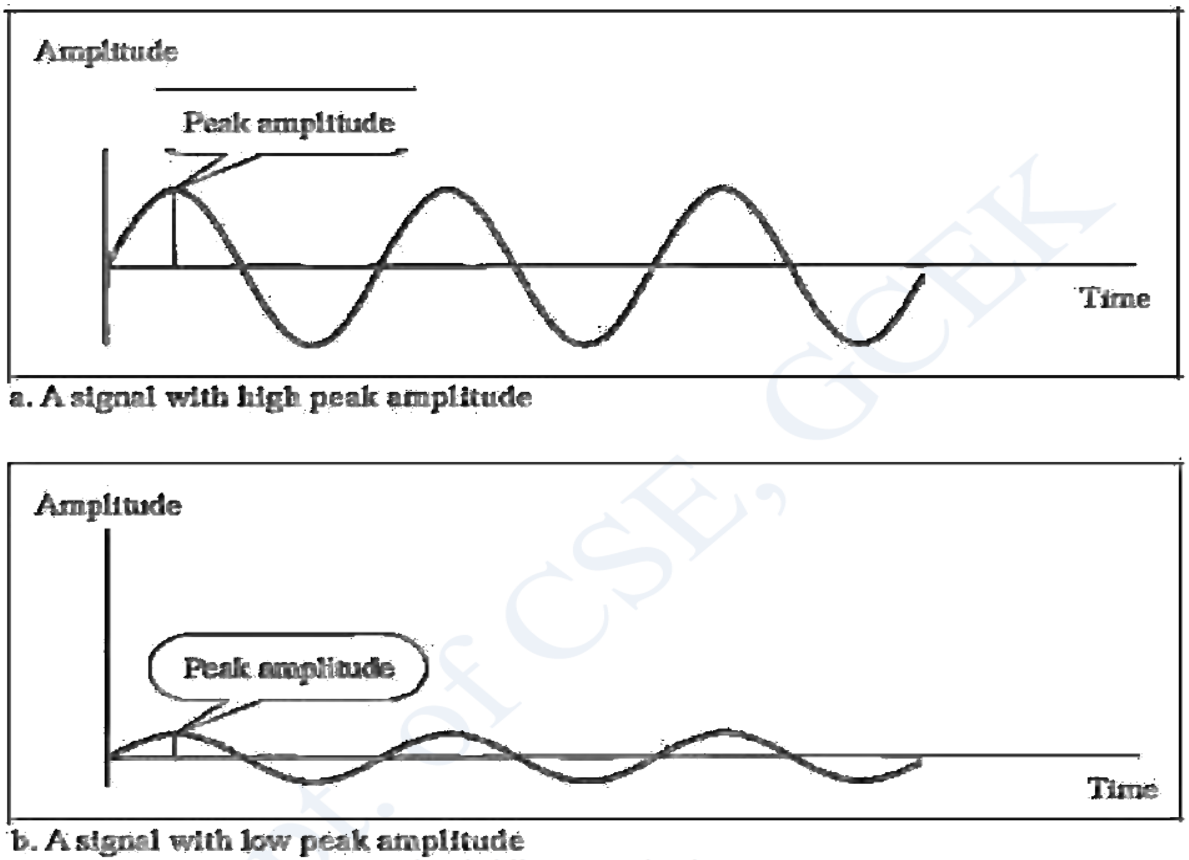
The peak amplitude of a signal is the absolute value of its highest intensity, proportional to the energy it carries. For electric signals, peak amplitude is normally measured in volts.

Period and Frequency

Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle. Frequency refers to the number of periods in 1s.

Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show. $f = 1/T$ and $T = 1/f$

Unit	Equivalent	Unit
Seconds (s)	1 s	Hertz (Hz)
Milliseconds (ms)	10^{-3} s	Kilohertz (kHz)
Microseconds (μs)	10^{-6} s	Megahertz (MHz
Nanoseconds (ns)	10^{-9} s	Gigahertz (GHz)
Picoseconds (ps)	10^{-12} s	Terahertz (THz)



a. A signal with high peak amplitude

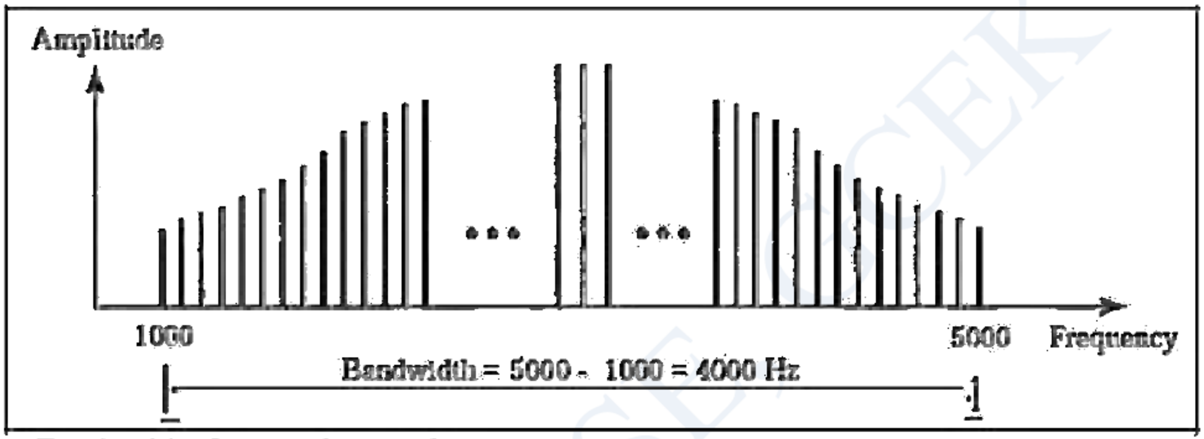
b. A signal with low peak amplitude

Fig. 2.3: signal with different amplitudes

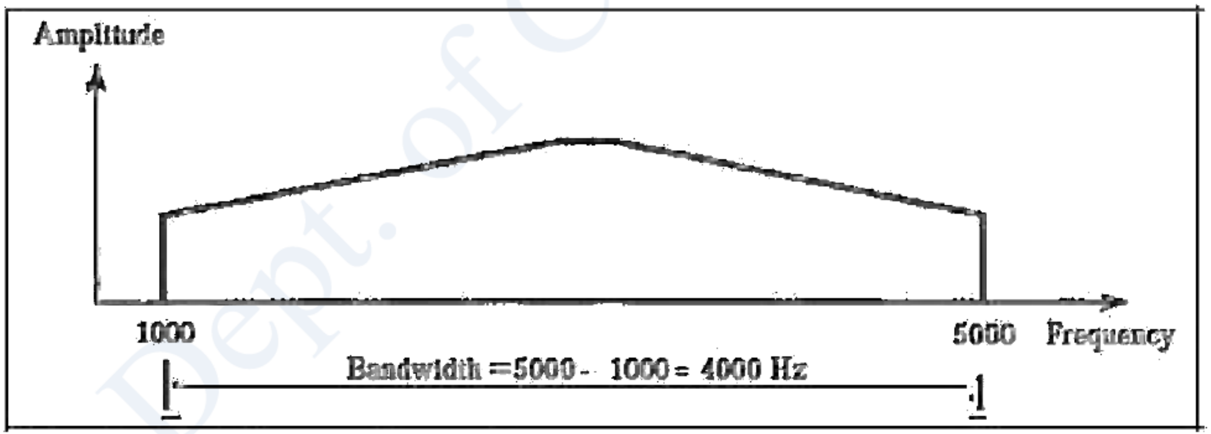
Composite Signals

Composite Signals

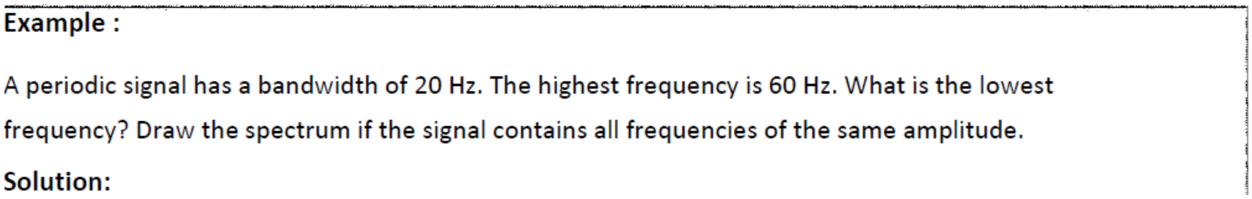
Simple sine waves have many applications in daily life. We can send a single sine wave to carry electric energy from one place to another. For example, the power company sends a single sine wave with a frequency of 60 Hz to distribute electric energy to houses and businesses. A single frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves. According to Fourier analysis, any composite signal is a combination of simple sine waves with different frequencies, amplitudes, and phases.



a. Bandwidth of a periodic signal



b. Bandwidth of a nonperiodic signal



Example :

A periodic signal has a bandwidth of 20 Hz. The highest frequency is 60 Hz. What is the lowest frequency? Draw the spectrum if the signal contains all frequencies of the same amplitude.

Solution:

Let f_h be the highest frequency, f_z the lowest frequency, and B the bandwidth.
Then B $=f_h - f_z$
20 $=60 - f_z$ or $f_z = 60 - 20 = 40$ Hz

Transmission Impairment

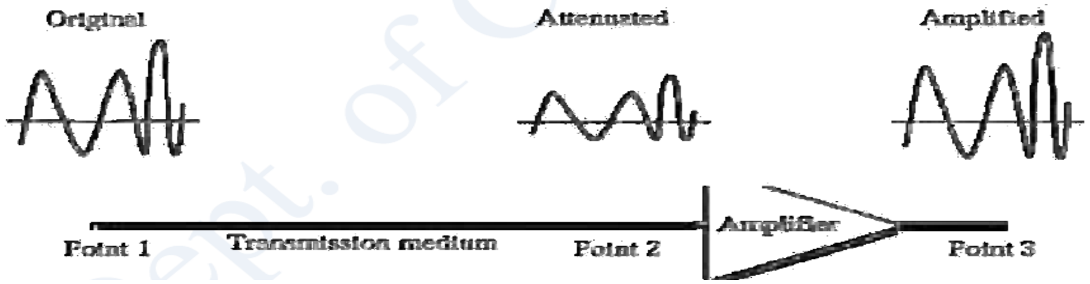
Signals travel through transmission media, which are not perfect. The imperfection causes signal impairment. This means that the signal at the beginning of the medium is not the same as the signal at the end of the medium. What is sent is not what is received.

Three causes of impairment are

attenuation, distortion, and **noise**.

- **Attenuation**

Attenuation means a loss of energy. When a signal, simple or composite, travels through a medium, it loses some of its energy in overcoming the resistance of the medium. That is why a wire carrying electric signals gets warm, if not hot, after a while. Some of the electrical energy in the signal is converted to heat. To compensate for this loss, amplifiers are used to amplify the signal.



Decibel

To show that a signal has lost or gained strength, engineers use the unit of the decibel.
The decibel (dB) measures the relative strengths of two signals or one signal at two different points.

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

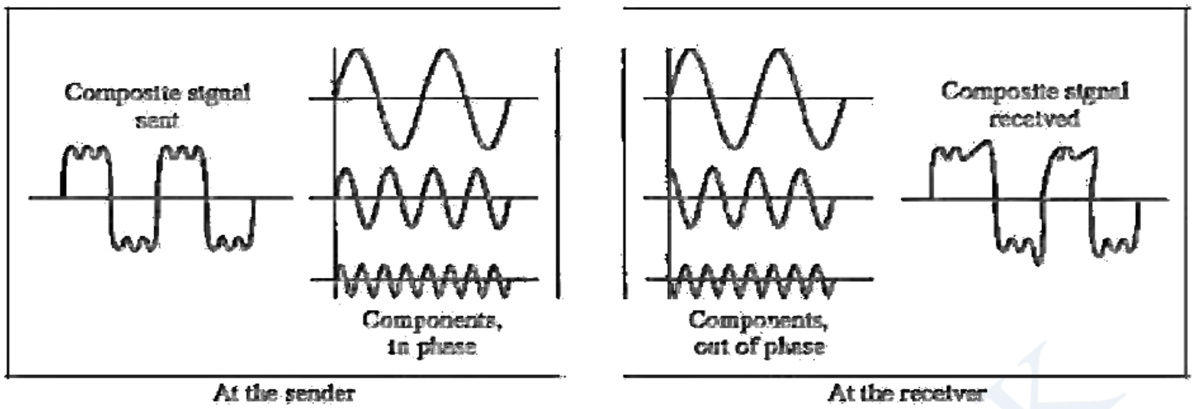
Note that the decibel is negative if a signal is attenuated and positive if a signal is amplified. Variables PI and P2 are the powers of a signal at points 1 and 2, respectively.

Example:
Suppose a signal travels through a transmission medium and its power is reduced to one-half. Find the attenuation (loss of power).
Solution:
dB=10 log (P/2P)= -3 dB

Example:
A signal travels through an amplifier, and its power is increased 10 times Find the amplification (gain of power).
Solution:
dB=10 log (10P/P)= 10 dB

- **Distortion**

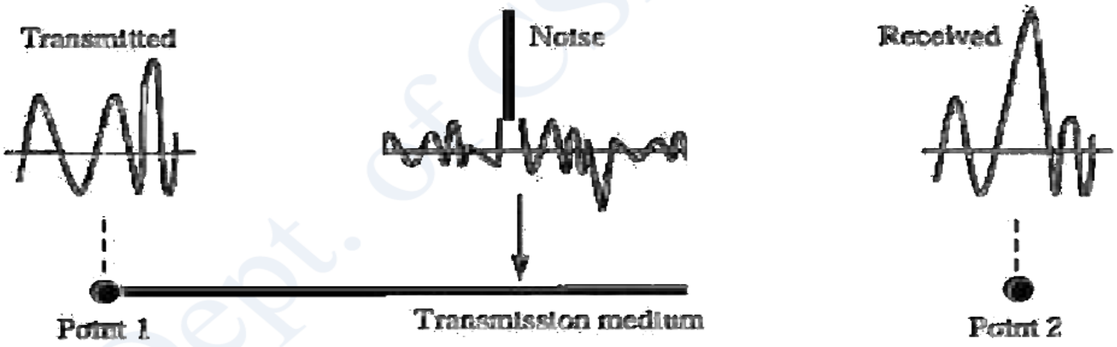
Distortion means that the signal changes its form or shape.Distortion can occur in a composite signal made of different frequencies. Each signal component has its own propagation speed (see the next section) through a medium and, therefore, its own delay in arriving at the final destination. Differences in delay may create a difference in phase if the delay is not exactly the same as the period duration.



In other words, signal components at the receiver have phases different from what they had at the sender. The shape of the composite signal is therefore not the same.

- **Noise**

Noise is another cause of impairment. Several types of noise, such as thermal noise, induced noise,crosstalk, and impulse noise, may corrupt the signal. Thermal noise is the random motion of electrons in a wire which creates an extra signal not originally sent by the transmitter. Induced noise comes from sources such as motors and appliances.



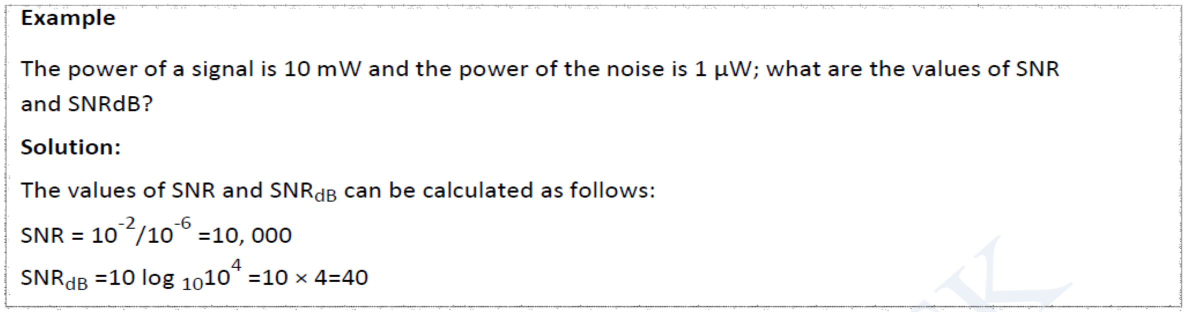
To find the theoretical bit rate limit, we need to know the ratio of the signal power to the noise power.

The signal-to-noise ratio is defined as:

SNR = average signal power/ average noise power

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

$$\mathrm{SNR}_{\mathrm{dB}} = 1\mathrm{Ologlo}\ \mathrm{SNR}$$



Example

The power of a signal is 10 mW and the power of the noise is 1 μW; what are the values of SNR and SNRdB?

Solution:

The values of SNR and $\mathrm{SNR}_{\mathrm{dB}}$ can be calculated as follows:

$\mathrm{SNR} = 10^{-2}/10^{-6} = 10,000$

$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} 10^{4} = 10 \times 4 = 40$

Performance measures:

The performance can be by checking the following parameters.

1) Bandwidth: can be represented in terms of hertz as well as bits per second. The bandwidth utilization should be high.

2) Throughput: is a measure of how fast we can actually send data through a network. So we may have a link with a bandwidth of 1 mbps, but we can only send let 300 kbps. So the throughput is 300kbps. The throughput should be high.

3) Latency: it defines how long it takes for an entire message to completely arrive at the destination from the time the 1st bit is sent out from the source.

4)Latency= propagation time + transmission time + queuing time + processing time
The latency should be low.

UNIT-1

UNIT-2

UNIT-3

1.1 Introduction to Data Comunication

When we communicate, we are sharing information. This sharing can be local or remote. Between individuals, local communication usually occurs face to face, while remote communication takes place over distance. The term telecommunication, which includes telephony, telegraphy, and television, means communication at a distance (tele is Greek for "far").
Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on 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.

II. Accuracy- The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable.

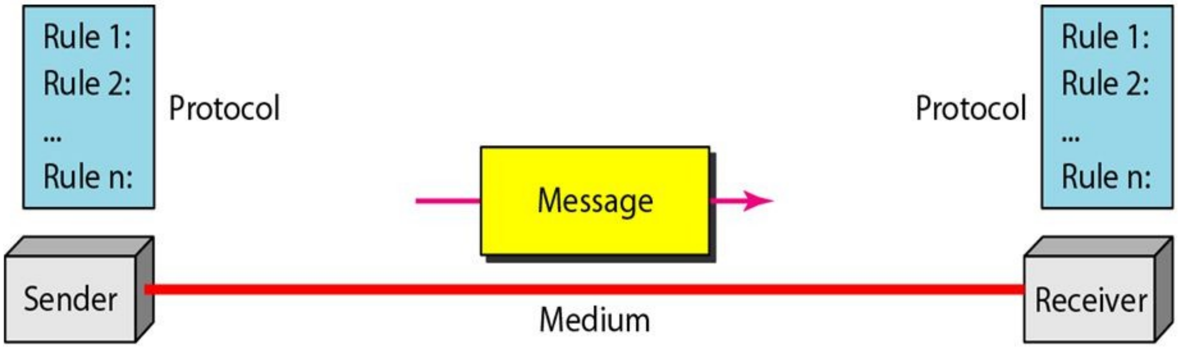
III. Timeliness- The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without significant delay. This kind of delivery is called real-time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

IV. 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 3D ms. If some of the packets arrive with 3D-ms delay and others with 4D-ms delay, an uneven quality in the video is the result.

1.1.1 Components of Data Communication

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.

II. Sender- The sender is the device that sends the data message. It can be a computer, workstation, telephone handset, video camera, and so on.



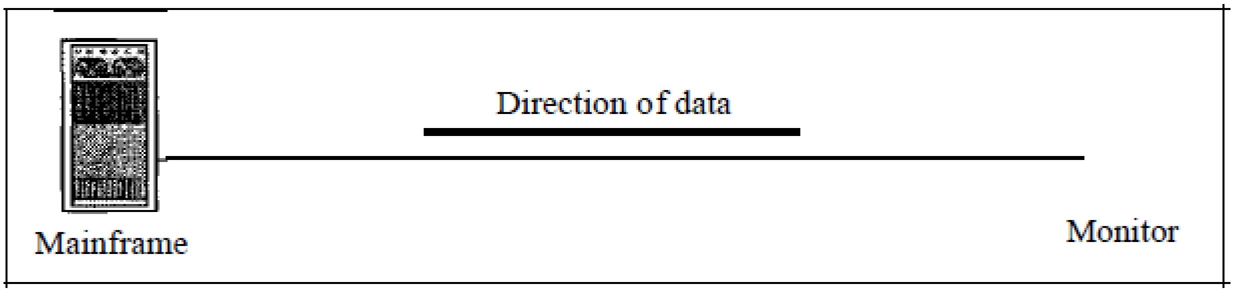
III. Receiver- The receiver is the device that receives the message. It can be a computer, workstation, telephone handset, television, and so on.

IV. 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.

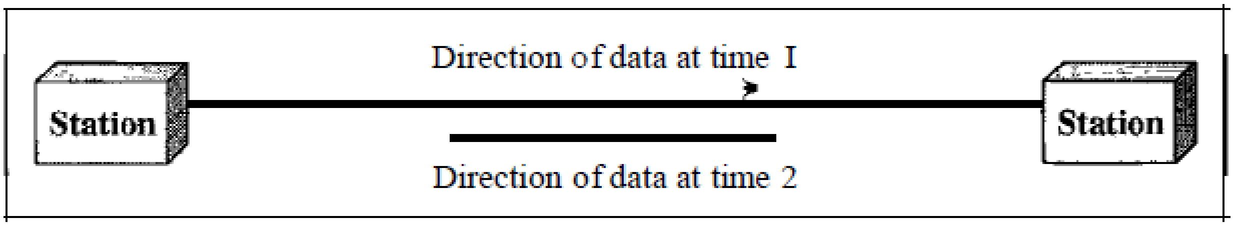
V. Protocol- A protocol is a set of rules that govern data communications. It represents an agreement between the communicating devices. Without a protocol, two devices may be connected but not communicating, just as a person speaking French cannot be understood by a person who speaks only Japanese.

1.1.2 Data Flow

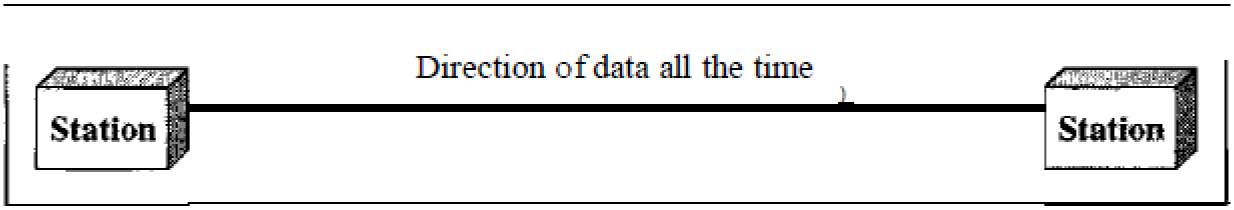
Communication between two devices can be simplex, half-duplex, or full-duplex



a. Simplex



b. Half-duplex



c. Full·duplex

I. 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. Keyboards and traditional monitors are examples of simplex devices. The keyboard can only introduce input; the monitor can only accept output. The simplex mode can use the entire capacity of the channel to send data in one direction.

II. 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. The half-duplex mode is like a one-lane road with traffic allowed in both directions. When cars are traveling in one direction, cars going the other way must wait. In a half-duplex transmission, the entire capacity of a channel is taken over by whichever of the two devices is transmitting at the time. Walkies-talkies and CB (citizens band) radios are both half-duplex systems. The half-duplex mode is used in cases where there is no need for communication in both directions at the same time; the entire capacity of the channel can be utilized for each direction.

III. Full-Duplex- In full-duplex mode (also called duplex), both stations can transmit and receive simultaneously. The full-duplex mode is like a tow-way street with traffic flowing in both directions at the same time. In full-duplex mode, signals going in one direction share the capacity of the link: with signals going in the other direction. This sharing can occur in two ways: Either the link must contain two physically separate transmission paths, one for sending and the other for receiving; or the capacity of the channel is divided between signals traveling in both directions. 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. The capacity of the channel, however, must be divided between the two directions.

1.2 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.

1.2.1 Network Criteria

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

I. 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.

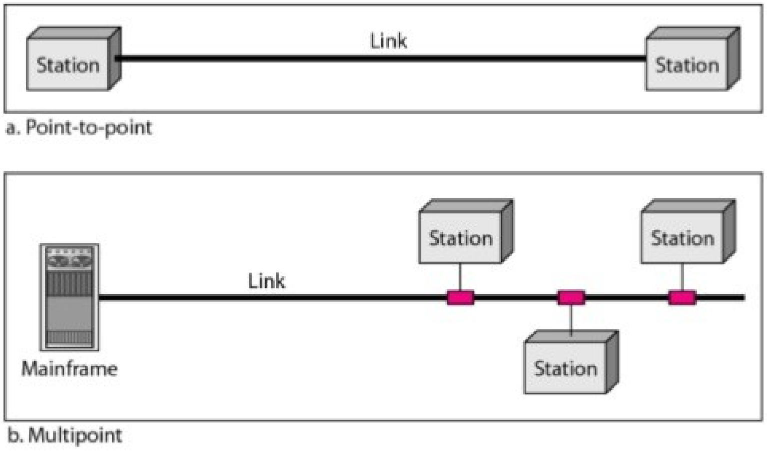
II. Reliability In addition to accuracy of delivery, network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in a catastrophe.

III. 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.

1.2.2 Types of Connections

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

I. 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.

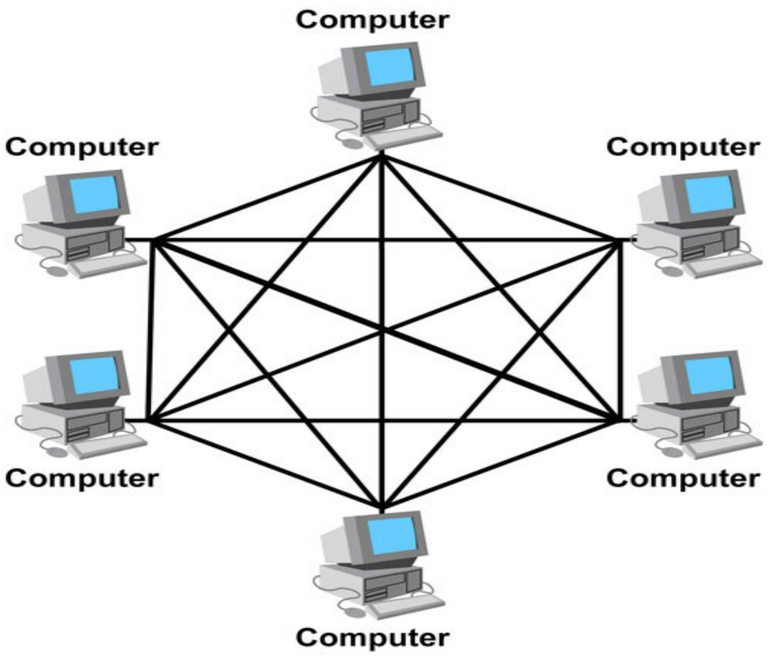


Multipoint A multipoint (also called multidrop) connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a spatially shared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

1.3 Topology

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.**

I. Mesh- In a mesh topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh network with n nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to n -1 nodes, node 2 must be connected to n - 1 nodes, and finally node n must be connected to n - 1 nodes. We need n(n - 1) physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need **n(n -1) /2** duplex-mode links. To accommodate that many links, every device on the network must have n - 1 input/output ports to be connected to the other n - 1 stations. **Example of a mesh** topology is the connection of telephone regional offices in which each regional office needs to be connected to every other regional office.



Advantage of Mesh Topology:-

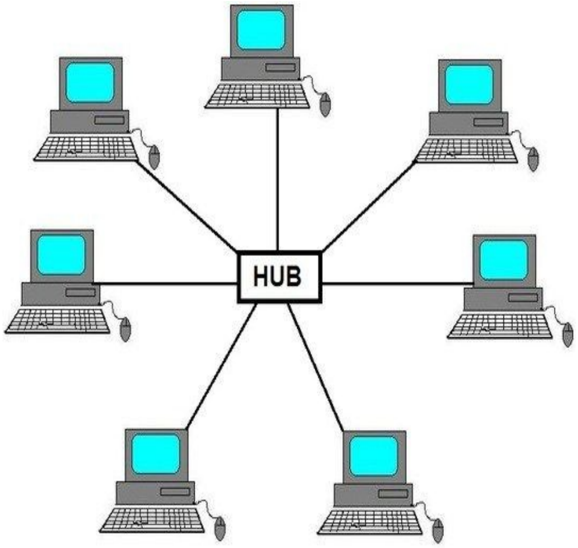
- The use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.
- A mesh topology is robust. If one link becomes unusable, it does not incapacitate the entire system
- There is the advantage of privacy or security. When every message travels along a dedicated line, only the intended recipient sees it. Physical boundaries prevent other users from gaining access to messages.
- point-to-point links make fault identification and fault isolation easy. Traffic can be routed to avoid links with suspected problems. This facility enables the network manager to discover the precise location of the fault and aids in finding its cause and solution.

Disadvantage of Mesh Topology:-

- Every device must be connected to every other device, installation and re-connection are difficult.
- the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.
- The hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.

II. Star Topology In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub. The devices are not directly linked to one another. Unlike a mesh topology, a star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device**.**

A star topology is less expensive than a mesh topology. In a star, each device needs only one link and one I/O port to connect it to any number of others. This factor also makes it easy to install and reconfigure. Far less cabling needs to be housed, and additions, moves, and deletions involve only one connection: between that device and the hub.



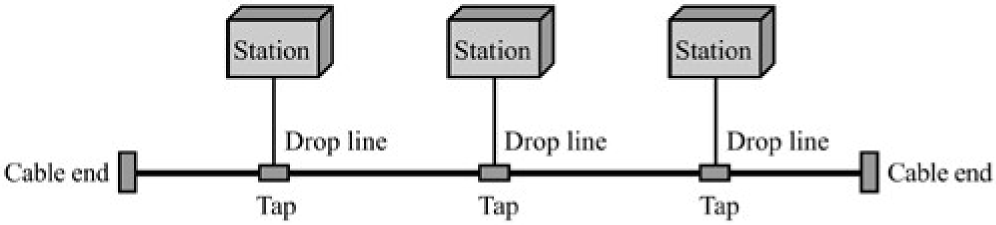
Advantage of Star Topology:-

- Easy to manage and maintain the network because each node require separate cable.
- Easy to locate problems because cable failure only affect a single user.
- Easy to extend the network without disturbing to the entire network.
- Due to Hub device network control and management is much easier.
- Fault identification and removing nodes in a network is easy.
- It provides very high speed of data transfer.

Disadvantage of Star Topology:-

- Entire performance of the network depends on the single device hub.
- If the hub device goes down, the entire network will be dead.
- Star topology requires more wires compared to the ring and bus topology.

III. Bus Topology - The preceding examples all describe point-to-point connections. A bus topology, on the other hand, is multipoint. One long cable acts as a backbone to link all the devices in a network. Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core. As a signal travels along the backbone, some ofits energy is transformed into heat. Therefore, it becomes weaker and weaker as it travels farther and farther. For this reason there is a limit on the number of taps a bus can support and on the distance between those taps



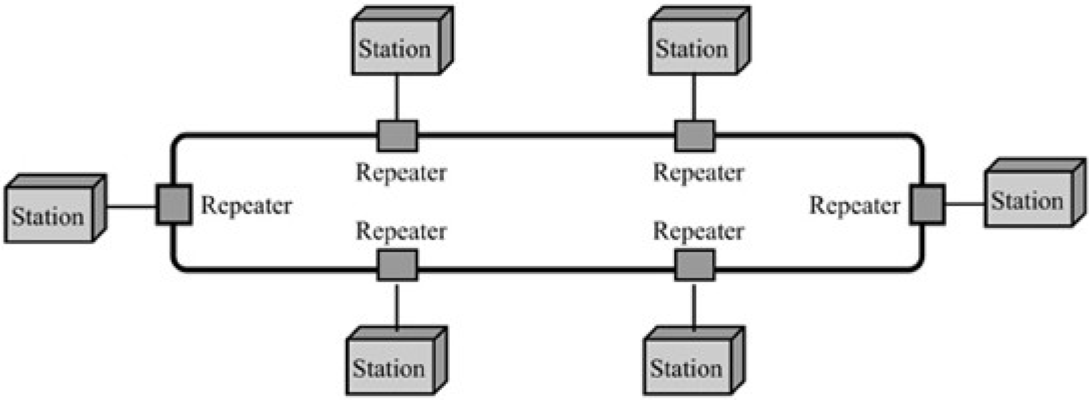
Advantage of Bus Topology:-

- **I**t 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.

Disadvantage 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

IV. Ring Topology- In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it. A signal is passed along the ring in one direction, from device to device, until it reaches its destination. Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.



A ring is relatively easy to install and reconfigure. Each device is linked to only its immediate neighbors (either physically or logically). To add or delete a device requires changing only two connections. The only constraints are media and traffic considerations (maximum ring length and number of devices). In addition, fault isolation is simplified. Generally in a ring, a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location. However, unidirectional traffic can be a disadvantage. In a simple ring, a break in the ring (such as a disabled station) can disable the entire network. This weakness can be solved by using a dual ring or a switch capable of closing off the break. Ring topology was prevalent when IBM introduced its local-area network Token Ring. Today, the need for higher-speed LANs has made this topology less popular.

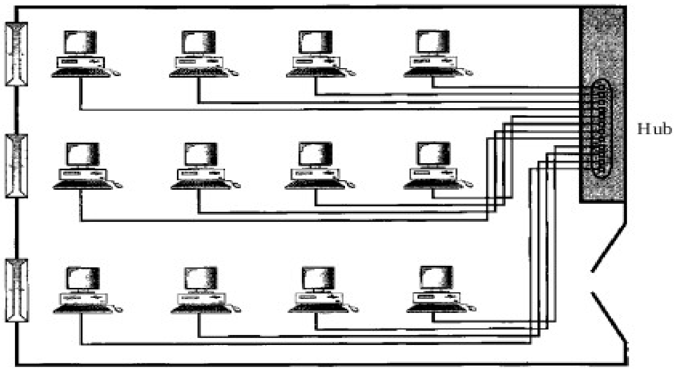
1.4 Network Category

we are generally referring to two primary categories: local-area networks and wide-area networks.

Local Area Network

A local area network (LAN) is usually privately owned and links the devices in a single office, building, or campus . Depending on the needs of an organization and the type of technology used, a LAN can be as simple as two PCs and a printer in someone's home office; or it can extend throughout a company and include audio and video peripherals. Currently, LAN size is limited to a few kilometers.

0 *An isolated IAN connecting 12 computers to a hub in a closet*

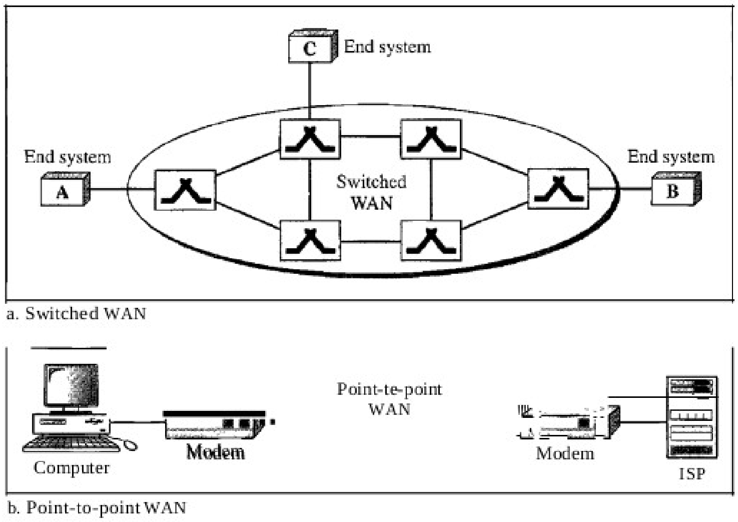


LANs are designed to allow resources to be shared between personal computers or workstations. The resources to be shared can include hardware (e.g, a printer), software (e.g., an application program), or data. A common example of a LAN, found in many business environments, links a work group of task-related computers, for example, engineering workstations or accounting PCs. One of the computers may be given a large capacity disk drive and may become a server to clients. Software can be stored on this central server and used as needed by the whole group. In this example, the size of the LAN may be determined by licensing restrictions on the number of users per copy of software, or by restrictions on the number of users licensed to access the operating system. In addition to size, LANs are distinguished from other types of networks by their transmission media and topology. In general, a given LAN will use only one type of transmission medium. The most common LAN topologies are bus, ring, and star. Early LANs had data rates in the 4 to 16 megabits per second (Mbps) range. Today, however, speeds are normally 100 or 1000 Mbps.

Wide Area Network

A wide area network (WAN) provides long-distance transmission of data, image, audio, and video information over large geographic areas that may comprise a country, a continent, or even the whole world. In Chapters 17 and 18 we discuss wide-area networks in greater detail. A WAN can be as complex as the backbones that connect the Internet or as simple as a dial-up line that connects a home computer to the Internet. We normally refer to the first as a switched WAN and to the second as a point-to-point WAN . The switched WAN connects the end systems, which usually comprise a router (internetworking connecting device) that connects to another LAN or WAN. The point-to-point WAN is normally a line leased from a telephone or cable TV provider that connects a home computer or a small LAN to an Internet service provider (lSP). This type of WAN is often used to provide Internet access.

1.11 *WANs: a switched WAN and a point-to-point WAN*



a. Switched WAN

b. Point-to-point WAN

An early example of a switched WAN is X.25, a network designed to provide connectivity between end users. As we will see in Chapter 18, X.25 is being gradually replaced by a high-speed, more efficient network called Frame Relay. A good example of a switched WAN is the asynchronous transfer mode (ATM) network, which is a network with fixed-size data unit packets called cells. We will discuss ATM in Chapter 18. Another example ofWANs is the wireless WAN that is becoming more and more popular.

Metropolitan Area Networks

A metropolitan area network (MAN) is a network with a size between a LAN and a WAN. It normally covers the area inside a town or a city. It is designed for customers who need a high-speed connectivity, normally to the Internet, and have endpoints spread over a city or part of city. A good example of a MAN is the part of the telephone company network that can provide a high-speed DSL line to the customer. Another example is the cable TV network that originally was designed for cable TV, but today can also be used for high-speed data connection to the Internet.

1.5 Protocol and Standards

In this section, we define two widely used terms: protocols and standards. First, we define protocol, which is synonymous with rule. Then we discuss standards, which are agreed-upon rules.

Protocols

In computer networks, communication occurs between entities in different systems. An entity is anything capable of sending or receiving information. However, two entities cannot simply send bit streams to each other and expect to be understood. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated, and when it is communicated. The key elements of a protocol are syntax, semantics, and timing.

- Syntax. The term syntax refers to the structure or format of the data, meaning the order in which they are presented. For example, a simple protocol might expect the first 8 bits of data to be the address of the sender, the second 8 bits to be the address of the receiver, and the rest of the stream to be the message itself.

- Semantics. The word semantics refers to the meaning of each section of bits. How is a particular pattern to be interpreted, and what action is to be taken based on that interpretation? For example, does an address identify the route to be taken or the final destination of the message?
- Timing. The term timing refers to two characteristics: when data should be sent and how fast they can be sent. For example, if a sender produces data at 100 Mbps but the receiver can process data at only 1 Mbps, the transmission will overload the receiver and some data will be lost.

Standards

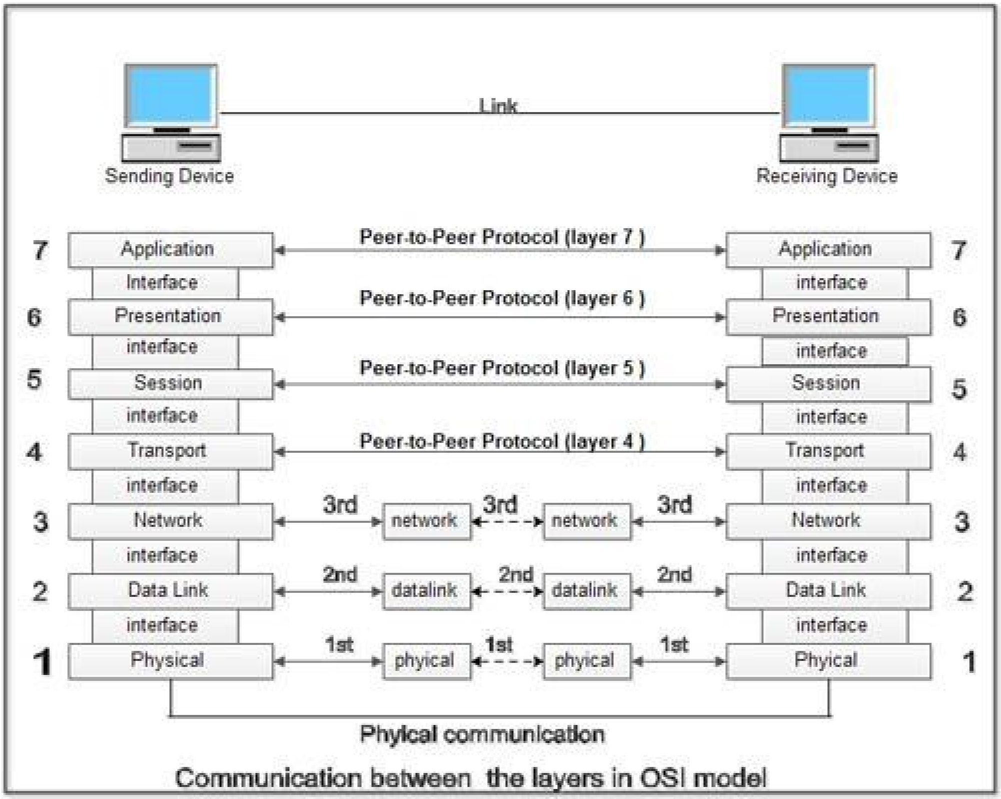
Standards are essential in creating and maintaining an open and competitive market for equipment manufacturers and in guaranteeing national and international interoperability of data and telecommunications technology and processes. Standards provide guidelines
to manufacturers, vendors, government agencies, and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communications. Data communication standards fall into two categories: de facto (meaning "by fact" or "by convention") and de jure (meaning "by law" or "by regulation").

- De facto. Standards that have not been approved by an organized body but have been adopted as standards through widespread use are de facto standards. De facto standards are often established originally by manufacturers who seek to define the functionality of a new product or technology
- De jure. Those standards that have been legislated by an officially recognized body are de jure standards.

1.6 The OSI Model

An ISO(International Standards Organization) introduce network communications model i.e Open System Interconnection(OSI) in 1970. An open system is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and inter operable. The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

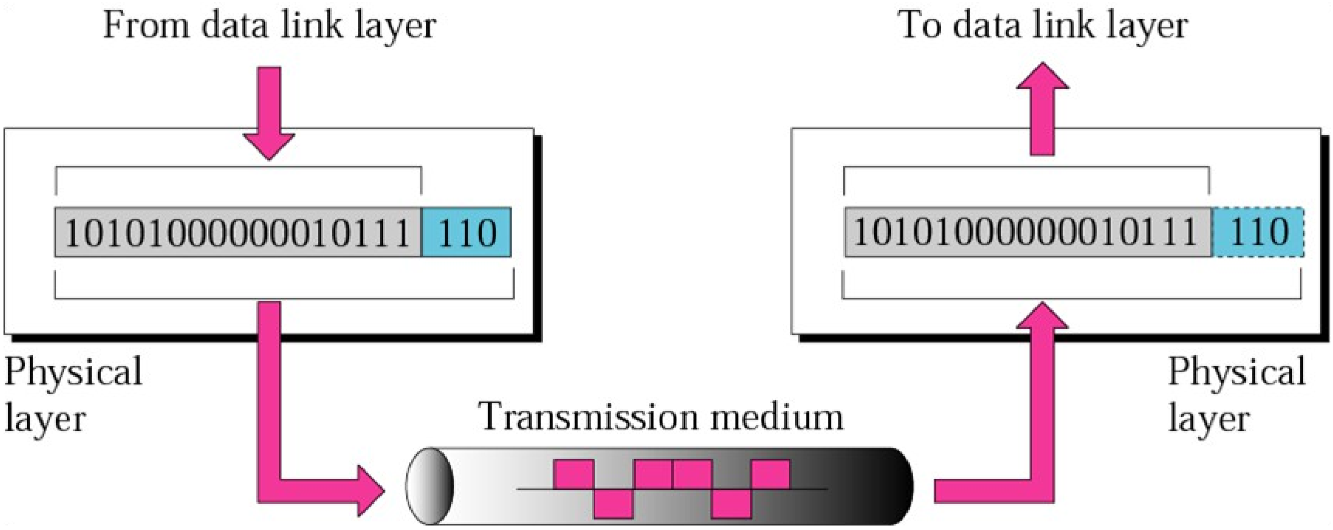
The OSI model is composed ofseven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7).



Communication between the layers in OSI model

Physical Layer

The physical layer coordinates the functions required to carry a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to Occur. Figure shows the position of the physical layer with respect to the transmission medium and the data link layer.



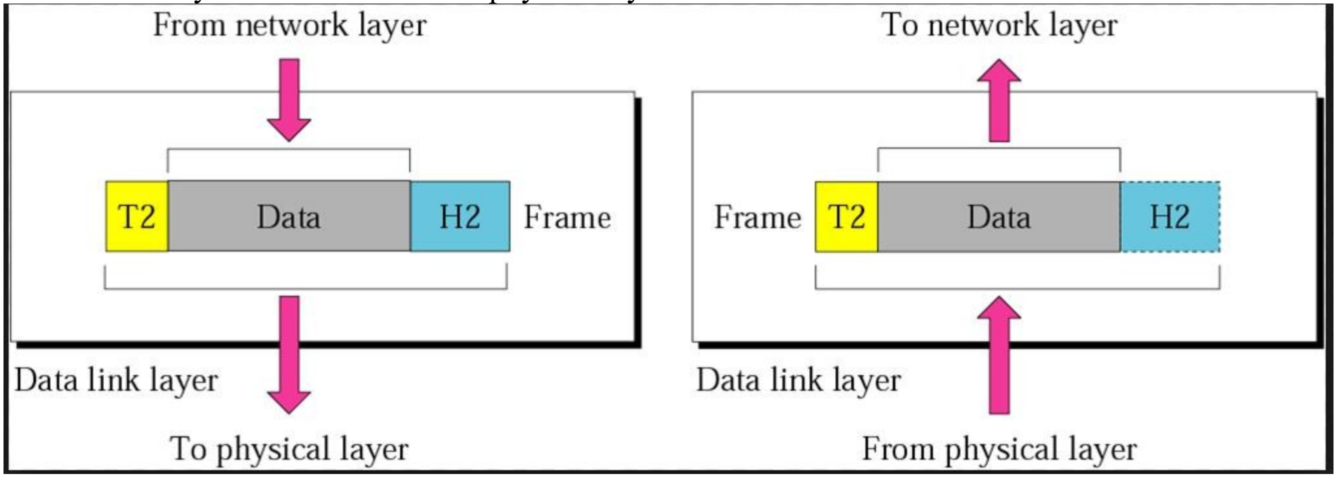
The physical layer is also concerned with the following:

- Physical characteristics of interfaces and medium. The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- Representation of bits. The physical layer data consists of a stream of bits (sequence of Os or 1s) with no interpretation. To be transmitted, bits must be encoded into signals--electrical or optical. The physical layer defines the type of encoding (how Os and Is are changed to signals).

- Data rate. The transmission rate-the number of bits sent each second-is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
- Synchronization of bits. The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.
- Line configuration. The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
- Physical topology. The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).
- Transmission mode. The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In simplex mode, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the half-duplex mode, two devices can send and receive, but not at the same time. In a full-duplex (or simply duplex) mode, two devices can send and receive at the same time.

Data Link Layer

The data link layer transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Figure shows the relationship of the data link layer to the network and physical layers.

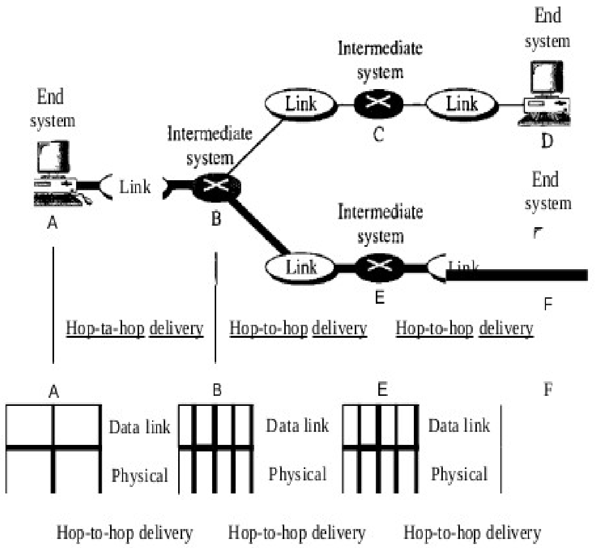


Other responsibilities of the data link layer include the following:

- Framing. The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
- Physical addressing. If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects the network to the next one.
- Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
- Error control. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.
- Access control. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Note:-The data link layer is responsible for moving frames from one hop (node) to the next.
As the figure shows, communication at the data link layer occurs between two adjacent nodes. To send data from A to F, three partial deliveries are made. First, the data link layer at A sends a frame to the data link layer at B (a router). Second, the data link layer at B sends a new frame to the data link layer at E. Finally, the data link layer at E sends a new frame to the data link layer at F.

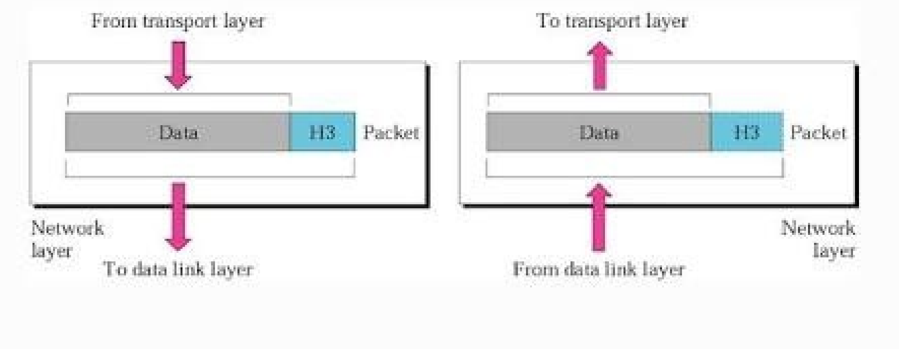
Figure 2.7 *Hop-to-hop delivery*



Note that the frames that are exchanged between the three nodes have different values in the headers. The frame from A to B has B as the destination address and A as the source address. The frame from B to E has E as the destination address and B as the source address. The frame from E to F has F as the destination address and E as the source address. The values of the trailers can also be different if error checking includes the header of the frame.

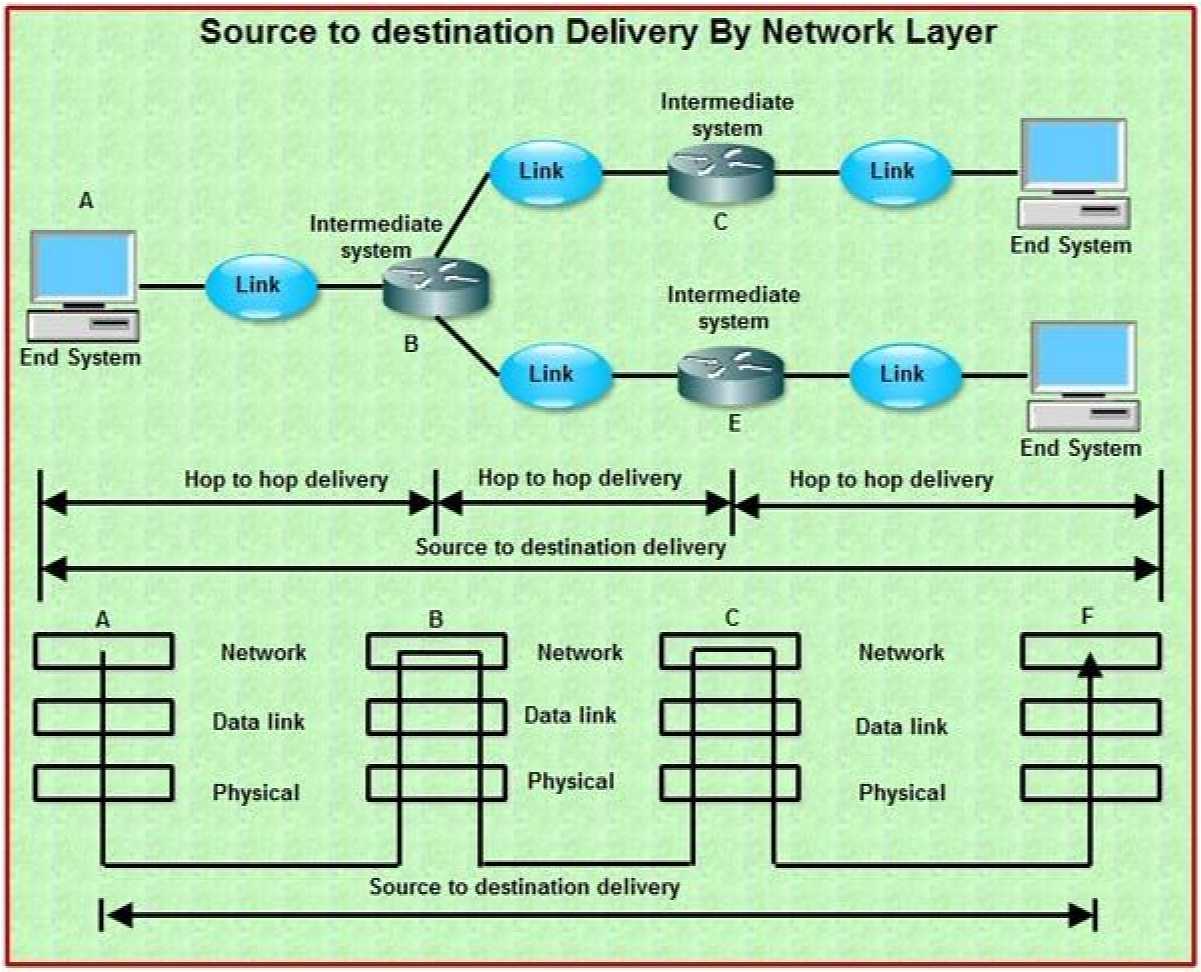
Network Layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination. If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. Figure shows the relationship of the network layer to the data link and transport layers.



Other responsibilities of the network layer include the following:

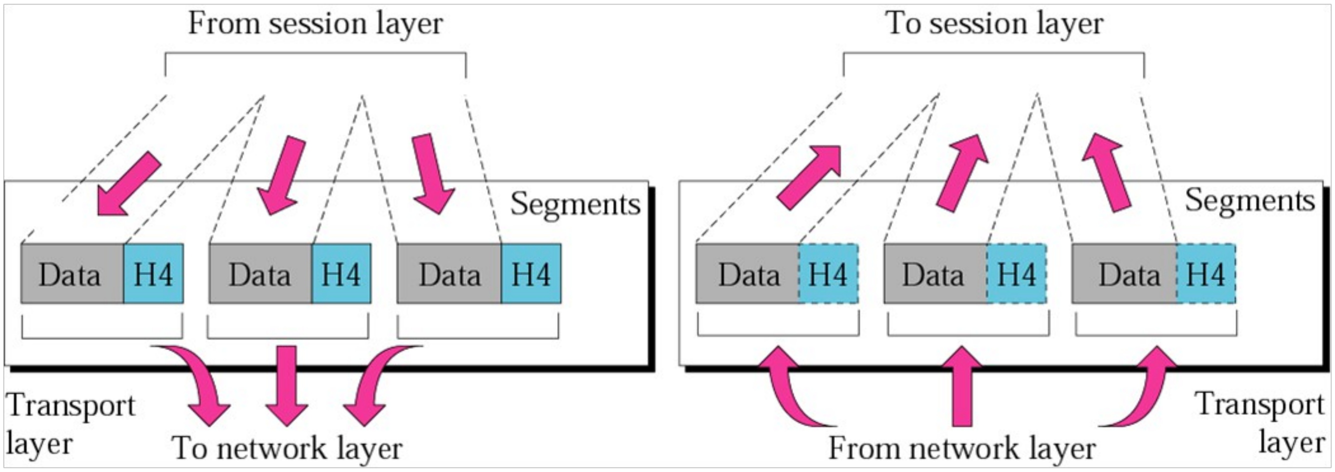
- Logical addressing. The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the logical addresses of the sender and receiver. We discuss logical addresses later in this chapter.
- Routing. When independent networks or links are connected to create intemetworks (network of networks) or a large network, the connecting devices (called routers or switches) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.



As the figure shows, now we need a source-to-destination delivery. The network layer at A sends the packet to the network layer at B. When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet. As we will see in later chapters, router B uses its routing table to find that the next hop is router E. The network layer at B, therefore, sends the packet to the network layer at E. The network
layer at E, in tum, sends the packet to the network layer at F .

Transport Layer

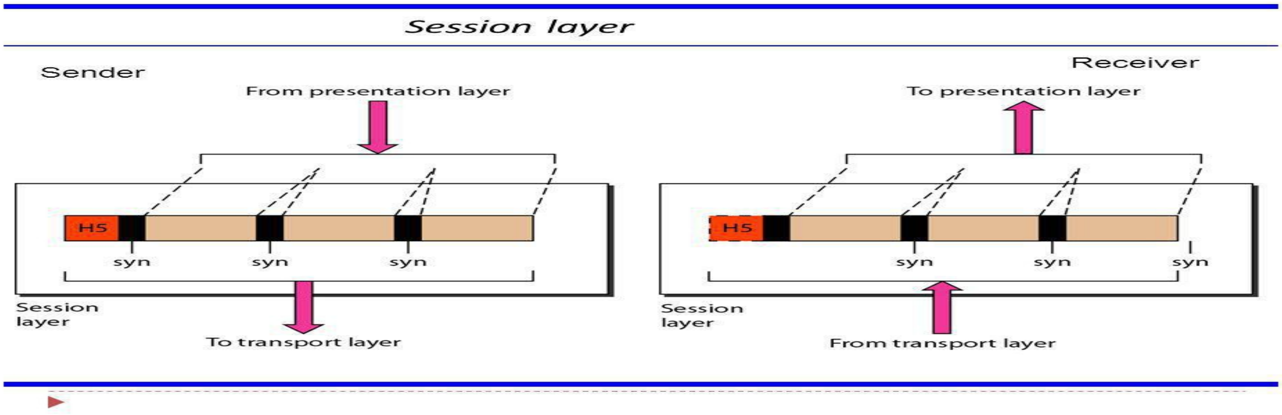
The transport layer is responsible for process-to-process delivery of the entire message. A process is an application program running on a host. Whereas the network layer oversees source-to-destination delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level. Figure shows the relationship of the transport layer to the network and session layers.



Other responsibilities of the transport layer include the following:

- Service-point addressing. Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header must therefore include a type of address called a service-point address (or port address). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.
- Segmentation and reassembly. A message is divided into transmittable segments, with each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in transmission.
- Connection control. The transport layer can be either connection less or connection-oriented. A connection less transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- Flow control. Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- Error control. Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to-process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

Session Layer The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains, and synchronizes the interaction among communicating systems.



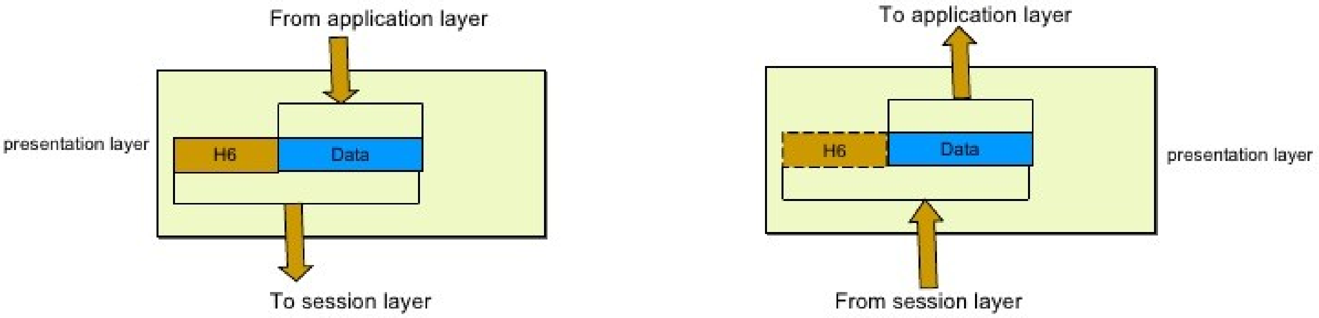
Specific responsibilities of the session layer include the following:

- Dialog control. The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half- duplex (one way at a time) or full-duplex (two ways at a time) mode.
- Synchronization. The session layer allows a process to add checkpoints, or syn Chronization points, to a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure illustrates the relationship of the session layer to the transport and presentation layers.

Presentation Layer

The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. Figure shows the relationship between the presentation layer and the application and session layers.

Specific responsibilities of the presentation layer include the following:



- Translation. The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The infonnation must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.
- Encryption. To carry sensitive information, a system must be able to ensure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- Compression. Data compression reduces the number of bits contained in the information. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

Figure 2.14 shows the relationship of the application layer to the user and the presentation layer. Of the many application services available, the figure shows only three:

XAOO (message-handling services), X.500 (directory services), and file transfer, access, and management (FTAM). The user in this example employs XAOO to send an e-mail message.

Specific services provided by the application layer include the following:

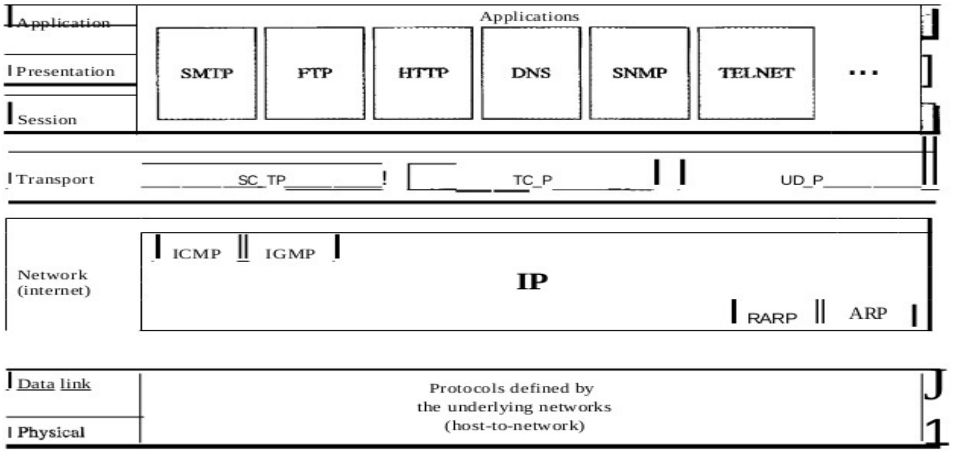
- Network virtual terminal. A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.

- File transfer, access, and management. This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- Mail services. This application provides the basis for e-mail forwarding and storage.
- Directory services. This application provides distributed database sources and access for global information about various objects and services.

1.7 TCP/IP Protocol Suite

The TCPIIP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. However, when TCP/IP is compared to OSI, we can say that the host-to-network layer is equivalent to the combination of the physical and data link layers. The internet layer is equivalent to the network layer, and the application layer is roughly doing the job of the session, presentation, and application layers with the transport layer in TCPIIP taking care of part of the duties of the session layer. So in this book, we assume that the TCPIIP protocol suite is made of five layers: physical, data link, network, transport, and application. The first four layers provide physical

Figure 2.16 *TCPIIP and OSI model*



standards, network interfaces, internetworking, and transport functions that correspond to the first four layers of the OSI model. The three topmost layers in the OSI model, however, are represented in TCPIIP by a single layer called the application layer.

TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality; however, the modules are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system. The term hierarchical means that each upper-level protocol is supported by one or more lower level protocols. At the transport layer, TCP/IP defines three protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP). At the network layer, the main protocol defined by TCP/IP is the Inter networking Protocol (IP); there are also some other protocols that support data movement in this layer.

A. Physical and Data Link Layers

At the physical and data link layers, TCPIIP does not define any specific protocol. It supports all the standard and proprietary protocols. A network in a TCPIIP inter network can be a local-area network or a wide-area network.

B. Network Layer

At the network layer (or, more accurately, the inter network layer), TCP/IP supports the Inter networking Protocol. IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP. Each of these protocols is described in greater detail in later chapters.

Inter networking Protocol (IP)

The Inter networking Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connection less protocol-a best-effort delivery service. The term best effort means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

IP transports data in packets called datagrams, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering data-grams once they arrive at their destination. The limited functionality of IP should not be considered a weakness, however. IP provides bare-bones transmission functions that free the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency.

Address Resolution Protocol

The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address, usually imprinted on the network interface card (NIC). ARP is used to find the physical address of the node when its Internet address is known.

Reverse Address Resolution Protocol

The Reverse Address Resolution Protocol (RARP) allows a host to discover its Internet address when it knows only its physical address. It is used when a computer is connected to a network for the first time or when a diskless computer is booted.

Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender. ICMP sends query and error reporting messages.

Internet Group Message Protocol

The Internet Group Message Protocol (IGMP) is used to facilitate the simultaneous transmission of a message to a group of recipients.

C. Transport Layer

Traditionally the transport layer was represented in TCP/IP by two protocols: TCP and UDP. IP is a host-to-host protocol, meaning that it can deliver a packet from one physical device to another. UDP and TCP are transport level protocols responsible or delivery of a message from a process (running program) to another process. A new transport layer protocol, SCTP, has been devised to meet the needs of some newer applications.

User Datagram Protocol

The User Datagram Protocol (UDP) is the simpler of the two standard TCPIIP transport protocols. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer.

Transmission Control Protocol

The Transmission Control Protocol (TCP) provides full transport-layer services to applications. TCP is a reliable stream transport protocol. The term stream, in this context, means connection-oriented: A connection must be established between both ends of a transmission before either can transmit data. At the sending end of each transmission, TCP divides a stream of data into smaller units called segments. Each segment includes a sequence number for reordering after receipt, together with an acknowledgment number for the segments received. Segments are carried across the internet inside of IP datagrams. At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers.

Stream Control Transmission Protocol

The Stream Control Transmission Protocol (SCTP) provides support for newer applications such as voice over the Internet. It is a transport layer protocol that combines the best features of UDP and TCP.

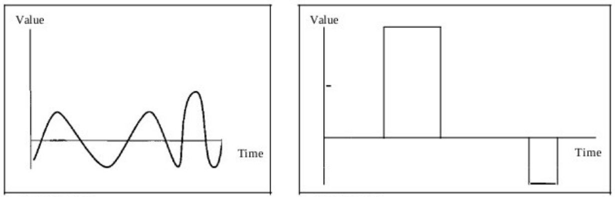
D. Application Layer

The application layer in TCPIIP is equivalent to the combined session, presentation, and application layers in the OSI modeL Many protocols are defined at this layer. We cover many of the standard protocols in later chapters.

1.8 Analog and Digital Signal

Like the data they represent, signals can be either analog or digital. An analog signal has infinitely many levels of intensity over a period of time. As the wave moves from value A to value B, it passes through and includes an infinite number of values along its path. A digital signal, on the other hand, can have only a limited number of defined values. Although each value can be any number, it is often as simple as 1 and O. The simplest way to show signals is by plotting them on a pair of perpendicular axes. The vertical axis represents the value or strength of a signal. The horizontal axis represents time.

Figure 3.1 *Comparison ofanalog and digital signals*



a. Analog signal

b. Digital signal

Figure illustrates an analog signal and a digital signal. The curve representing the analog signal passes through an infinite number of points. The vertical lines of the digital signal, however, demonstrate the sudden jump that the signal makes from value to value.

Periodic and Nonperiodic Signals

Both analog and digital signals can take one of two forms: periodic or nonperiodic (sometimes refer to as aperiodic, because the prefix a in Greek means "non"). A periodic signal completes a pattern within a measurable time frame, called a period, and repeats that pattern over subsequent identical periods. The completion of one full pattern is called a cycle. A nonperiodic signal changes without exhibiting a pattern or cycle that repeats over time. Both analog and digital signals can be periodic or nonperiodic. In data communications, we commonly use periodic analog signals (because they need less bandwidth) and nonperiodic digital signals (because they can represent variation in data).

1.9 Periodic Analog Signals

Periodic analog signals can be classified as simple or composite. A simple periodic analog signal, a sine wave, cannot be decomposed into simpler signals. A composite periodic analog signal is composed of multiple sine waves.

Sine Wave

The sine wave is the most fundamental form of a periodic analog signal. When we visualize it as a simple oscillating curve, its change over the course of a cycle is smooth and consistent, a continuous, rolling flow. Figure 3.2 shows a sine wave. Each cycle consists of a single arc above the time axis followed by a single arc below it. A sine wave can be represented by three parameters: the peak amplitude, the frequency, and the phase. These three parameters fully describe a sine wave.

Peak Amplitude

The peak amplitude of a signal is the absolute value of its highest intensity, proportional to the energy it carries. For electric signals, peak amplitude is normally measured in volts.

Period and Frequency

Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle. Frequency refers to the number of periods in I s. Note that period and frequency are just one characteristic defined in two ways. Period is the inverse of frequency, and frequency is the inverse of period, as the following formulas show.

Phase

The term phase describes the position of the waveform relative to time O. If we think of the wave as something that can be shifted backward or forward along the time axis, phase describes the amount of that shift. It indicates the status of the first cycle.

Wavelength

Wavelength is another characteristic of a signal traveling through a transmission medium. Wavelength binds the period or the frequency of a simple sine wave to the propagation speed of the medium.

Time and Frequency Domains

A sine wave is comprehensively defined by its amplitude, frequency, and phase. We have been showing a sine wave by using what is called a time-domain plot. The time domain plot shows changes in signal amplitude with respect to time (it is an amplitude versus-time plot). Phase is not explicitly shown on a time-domain plot. To show the relationship between amplitude and frequency, we can use what is called a frequency-domain plot. A frequency-domain plot is concerned with only the peak value and the frequency. Changes of amplitude during one period are not shown.

Bandwidth

The range of frequencies contained in a composite signal is its bandwidth. The bandwidth is normally a difference between two numbers. For example, if a composite signal contains frequencies between 1000 and 5000, its bandwidth is 5000 - 1000, or 4000.

1.10 Digital Signals

In addition to being represented by an analog signal, information can also be repre-sented by a digital signal. For example, a I can be encoded as a positive voltage and a 0 as zero voltage. A digital signal can have more than two levels. In this case, we can send more than 1 bit for each level. Figure 3.16 shows two signals, one with two levels and the other with four.

Bit Rate

Most digital signals are nonperiodic, and thus period and frequency are not appropriate characteristics. Another term-bit rate (instead ofjrequency)-is used to describe digital signals. The bit rate is the number of bits sent in Is, expressed in bits per second (bps). Figure 3.16 shows the bit rate for two signals.

Bit Length

We discussed the concept of the wavelength for an analog signal: the distance one cycle occupies on the transmission medium. We can define something similar for a digital signal: the bit length. The bit length is the distance one bit occupies on the transmission medium.

Bit length = propagation speed x bit duration

Digital Signal as a Composite Analog Signal

Based on Fourier analysis, a digital signal is a composite analog signal. The bandwidth is infinite, as you may have guessed. We can intuitively corne up with this concept when we consider a digital signal. A digital signal, in the time domain, comprises connected vertical and horizontal line segments. A vertical line in the time domain means a frequency of infinity (sudden change in time); a horizontal line in the time domain means a frequency of zero (no change in time). Going from a frequency of zero to a frequency of infinity (and vice versa) implies all frequencies in between are part of the domain.

Fourier analysis can be used to decompose a digital signal. If the digital signal is periodic, which is rare in data communications, the decomposed signal has a frequency- domain representation with an infinite bandwidth and discrete frequencies. If the digital signal is nonperiodic, the decomposed signal still has an infinite bandwidth, but the frequencies are continuous. Figure 3.17 shows a periodic and a nonperiodic digital signal and their bandwidths.

Transmission of Digital Signals

The previous discussion asserts that a digital signal, periodic or nonperiodic, is a composite analog signal with frequencies between zero and infinity. For the remainder of the discussion, let us consider the case of a nonperiodic digital signal, similar to the ones we encounter in data communications. The fundamental question is, How can we send a digital signal from point A to point B? We can transmit a digital signal by using one of two different approaches: baseband transmission or broadband transmission (using modulation).

Baseband Transmission

Baseband transmission means sending a digital signal over a channel without changing the digital signal to an analog signal. Figure 3.18 shows baseband transmission. Baseband transmission requires that we have a low-pass channel, a channel with a bandwidth that starts from zero. This is the case if we have a dedicated medium with a bandwidth constituting only one channel. For example, the entire bandwidth of a cable connecting two computers is one single channel. As another example, we may connect several computers to a bus, but not allow more than two stations to communicate at a time. Again we have a low-pass channel, and we can use it for baseband communication.

Figure 3.19 shows two low-pass channels: one with a narrow bandwidth and the other with a wide bandwidth. We need to remember that a low-pass channel with infinite bandwidth is ideal, but we cannot have such a channel in real life. However, we can get close. Let us study two cases of a baseband communication: a low-pass channel with a wide bandwidth and one with a limited bandwidth.

Case 1: Low-Pass Channel with Wide Bandwidth

If we want to preserve the exact form of a nonperiodic digital signal with vertical segments vertical and horizontal segments horizontal, we need to send the entire spectrum, the continuous range of frequencies between zero and infinity. This is possible if we have a dedicated medium with an infinite bandwidth between the sender and receiver that preserves the exact amplitude of each component of the composite signal. Although this may be possible inside a computer (e.g., between CPU and memory), it is not possible between two devices. Fortunately, the amplitudes of the frequencies at the border of the bandwidth are so small that they can be ignored. This means that if we have a medium, such as a coaxial cable or fiber optic, with a very wide bandwidth, two

stations can communicate by using digital signals with very good accuracy, as shown in

Figure 3.20. Note that!i is close to zero, andh is very high. Although the output signal is not an exact replica of the original signal, the data can still be deduced from the received signal. Note that although some of the frequencies are blocked by the medium, they are not critical.

Case 2: Low-Pass Channel with Limited Bandwidth

In a low-pass channel with limited bandwidth, we approximate the digital signal with an analog signal. The level of approximation depends on the bandwidth available. Rough Approximation Let us assume that we have a digital signal of bit rate N. If we want to send analog signals to roughly simulate this signal, we need to consider the worst case, a maximum number of changes in the digital signal. This happens when the signal carries the sequence 01010101 ... or the sequence 10101010· ... To simulate these two cases, we need an analog signal of frequency f = N12. Let 1 be the positive peak value and to be the negative peak value. We send 2 bits in each cycle; the frequency of the analog signal is one-half of the bit rate, or N12. However, just this one frequency cannot make all patterns; we need more components. The maximum frequency is NI2. As an example of this concept, let us see how a digital signal with a 3-bit pattern can be simulated by using analog signals. Figure 3.21 shows the idea. The two similar cases (000 and 111) are simulated with a signal with frequency f = 0 and a phase of 180° for 000 and a phase of 0° for 111. The two worst cases (010 and 101) are simulated with an analog signal with frequency f = NI2 and phases of 180° and 0°. The other four cases can only be simulated with an analog signal with f = NI4 and phases of 180°, 270°, 90°, and 0°. In other words, we need a channel that can handle frequencies 0, N14, and NI2. This rough approximation is referred to as using the first harmonic (NI2) frequency. The required bandwidth is Better Approximation To make the shape of the analog signal look more like that of a digital signal, we need to add more harmonics of the frequencies. We need to increase the bandwidth. We can increase the bandwidth to 3N12, 5N12, 7 NI2, and so on. Figure shows the effect of this increase for one of the worst cases, the pattern 010.

1.11 DATA RATE LIMITS

A very important consideration in data communications is how fast we can send data, in bits per second. over a channel. Data rate depends on three factors:

1. The bandwidth available
2. The level of the signals we use
3. The quality of the channel (the level of noise)

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

Noiseless Channel: Nyquist Bit Rate

For a noiseless channel, the Nyquist bit rate formula defines the theoretical maximum bit rate

BitRate = 2 x bandwidth x 10g2 L

In this formula, bandwidth is the bandwidth of the channel, L is the number of signal evels used to represent data, and BitRate is the bit rate in bits per second. According to the formula, we might think that, given a specific bandwidth, we can have any bit rate we want by increasing the number of signa11eve1s. Although the idea is theoretically correct, practically there is a limit. When we increase the number of signal 1eve1s, we impose a burden on the receiver. If the number of levels in a signal is just 2, the receiver can easily distinguish between a 0 and a 1. If the level of a signal is 64, the receiver must be very sophisticated to distinguish between 64 different levels. In other words, increasing the levels of a signal reduces the reliability of the system.

Noisy Channel: Shannon Capacity

In reality, we cannot have a noiseless channel; the channel is always noisy. In 1944, Claude Shannon introduced a formula, called the Shannon capacity, to determine the theoretical highest data rate for a noisy channel:

Capacity = bandwidth X $\log_2 (1 + \text{SNR})$

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

1.12 Performance

One important issue in networking is the performance of the network-how good is it? We discuss quality of service, an overall measurement of network performance. In this section, we introduce terms that we need for future chapters.

Bandwidth

One characteristic that measures network performance is bandwidth. However, the term can be used in two different contexts with two different measuring values: bandwidth in hertz and bandwidth in bits per second.

Bandwidth in Hertz

We have discussed this concept. Bandwidth in hertz is the range of frequencies contained in a composite signal or the range of frequencies a channel can pass. For example, we can say the bandwidth of a subscriber telephone line is 4 kHz.

Bandwidth in Bits per Seconds

The term bandwidth can also refer to the number of bits per second that a channel, a link, or even a network can transmit. For example, one can say the bandwidth of a Fast Ethernet network (or the links in this network) is a maximum of 100 Mbps. This means that this network can send 100 Mbps.

Relationship

There is an explicit relationship between the bandwidth in hertz and bandwidth in bits per seconds. Basically, an increase in bandwidth in hertz means an increase in bandwidth in bits per second. The relationship depends on whether we have baseband transmission or transmission with modulation.

In networking, we use the term bandwidth in two contexts.

- The first, bandwidth in hertz, refers to the range of frequencies in a composite signal or the range of frequencies that a channel can pass.
- The second, bandwidth in bits per second, refers to the speed of bit transmission in a channel or link.

Throughput

The throughput is a measure of how fast we can actually send data through a network. Although, at first glance, bandwidth in bits per second and throughput seem the same, they are different. A link may have a bandwidth of B bps, but we can only send T bps through this link with T always less than B. In other words, the bandwidth is a potential measurement of a link; the throughput is an actual measurement of how fast we can send data. For example, we may have a link with a bandwidth of 1 Mbps, but the devices connected to the end of the link may handle only 200 kbps. This means that we cannot send more than 200 kbps through this link. Imagine a highway designed to transmit 1000 cars per minute from one point to another. However, if there is congestion on the road, this figure may be reduced to 100 cars per minute. The bandwidth is 1000 cars per minute; the throughput is 100 cars per minute.

Latency (Delay)

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

Propagation Time

Propagation time measures the time required for a bit to travel from the source to the destination. The propagation time is calculated by dividing the distance by the propagation speed.
Propagation time = Distance / Propagation speed
The propagation speed of electromagnetic signals depends on the medium and on the frequency of the signaL For example, in a vacuum, light is propagated with a speed of 3 x 10 8 mfs. It is lower in air; it is much lower in cable.

Transmission Time

in data communications we don't send just 1 bit, we send a message. The first bit may take a time equal to the propagation time to reach its destination; the last bit also may take the same amount of time. However, there is a time between the first bit leaving the sender and the last bit arriving at the receiver. The first bit leaves earlier and arrives earlier; the last bit leaves later and arrives later. The time required for transmission of a message depends on the size of the message and the bandwidth of the channel.
Transmission time = Message size/Bandwidth

Queuing Time

The third component in latency is the queuing time, the time needed for each intermediate or end device to hold the message before it can be processed. The queuing time is not a fixed factor; it changes with the load imposed on the network. When there is heavy traffic on the network, the queuing time increases. An intermediate device, such as a router, queues the arrived messages and processes them one by one. If there are many messages, each message will have to wait.

Bandwidth-Delay Product

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

Jitter

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

UNIT-2

2.1 Types of Errors

Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference. This interference can change the shape of the signal. In a single-bit error, a 0 is changed to a 1 or a 1 to a O. In a burst error, multiple bits are changed. For example, a 11100 s burst of impulse noise on a transmission with a data rate of 1200 bps might change all or some of the 12 bits of information.

Single-Bit Error

The term single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.

To understand the impact of the change, imagine that each group of 8 bits is an ASCII character with a 0 bit added to the left. In Figure 10.1,00000010 (ASCII STX) was sent, meaning start of text, but 00001010 (ASCII LF) was received, meaning line feed. (For more information about ASCII code, see Appendix A.) Single-bit errors are the least likely type of error in serial data transmission. To understand why, imagine data sent at 1 Mbps. This means that each bit lasts only 1/1,000,000 s, or 1)ls. For a single-bit error to occur, the noise must have a duration of only 1)ls, which is very rare; noise normally lasts much longer than this.

Burst Error

The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1. In this case, 0100010001000011 was sent, but 0101110101100011 was received. Note that a burst error does not necessarily mean that the errors occur in consecutive bits. The length of the burst is measured from the first corrupted bit to the last corrupted bit. Some bits in between may not have been corrupted. A burst error is more likely to occur than a single-bit error. The duration of noise is normally longer than the duration of 1 bit, which means that when noise affects data, it affects a set of bits. The number of bits affected depends on the data rate and duration of noise. For example, if we are sending data at I kbps, a noise of 11100 s can affect 10 bits; if we are sending data at I Mbps, the same noise can affect 10,000 bits.

2.2 Detection Versus Correction

The correction of errors is more difficult than the detection. In error detection, we are looking only to see if any error has occurred. The answer is a simple yes or no. We are not even interested in the number of errors. A single-bit error is the same for us as a
burst error.

In error correction, we need to know the exact number of bits that are corrupted and more importantly, their location in the message. The number of the errors and the size of the message are important factors. If we need to correct one single error in an 8-bit data unit, we need to consider eight possible error locations; if we need to correct two errors in a data unit of the same size, we need to consider 28 possibilities. You can imagine the receiver's difficulty in finding 10 errors in a data unit of 1000 bits.

2.3 Block Coding

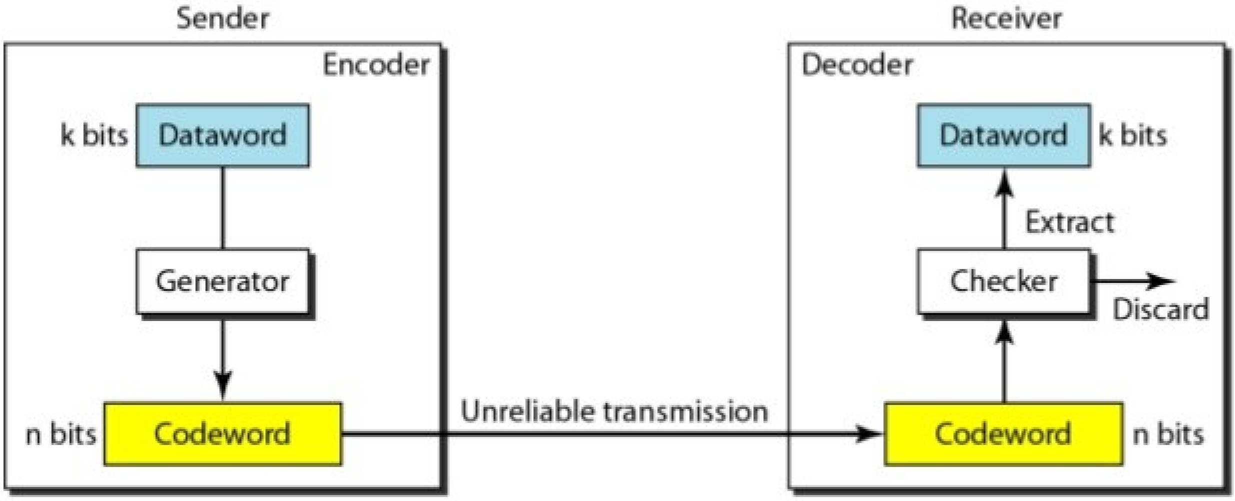
In block coding, we divide our message into blocks, each of k bits, called datawords. We add r redundant bits to each block to make the length n = k + r. The resulting n-bit blocks are called codewords. How the extra r bits is chosen or calculated is something we will discuss later. For the moment, it is important to know that we have a set of datawords, each of size k, and a set of codewords, each of size of n. With k bits, we can create a combination of 2^k datawords; with n bits, we can create a combination of 2^n codewords. Since n > k, the number of possible codewords is larger than the number of possible datawords. The block coding process is one-to-one; the same dataword is always encoded as the same codeword. This means that we have $2^n - 2^k$ codewords that are not used. We call these codewords invalid or illegal. Figure 10.5 shows the situation.

Error Detection

How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.
2. The original codeword has changed to an invalid one.

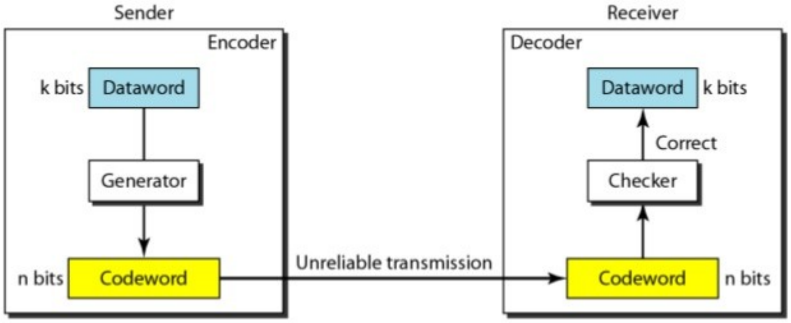
Figure shows the role of block coding in error detection.



The sender creates codewords out of datawords by using a generator that applies the rules and procedures of encoding (discussed later). Each codeword sent to the receiver may change during transmission. If the received codeword is the same as one of the valid codewords, the word is accepted; the corresponding dataword is extracted for use. If the received codeword is not valid, it is discarded. However, if the codeword is corrupted during transmission but the received word still matches a valid codeword, the error remains undetected. This type of coding can detect only single errors. Two or more errors may remain undetected.

Error Correction

As we said before, error correction is much more difficult than error detection. In error detection, the receiver needs to know only that the received codeword is invalid; in error correction the receiver needs to find (or guess) the original codeword sent. We can say that we need more redundant bits for error correction than for error detection. Figure 10.7 shows the role of block coding in error correction. We can see that the idea is the same as error detection but the checker functions are much more complex.



I. Comparing the received codeword with the first codeword in the table (01001 versus 00000), the receiver decides that the first codeword is not the one that was sent because there are two different bits.

2. By the same reasoning, the original codeword cannot be the third or fourth one in the table.

3. The original codeword must be the second one in the table because this is the only one that differs from the received codeword by 1 bit. The receiver replaces 01001 with 01011 and consults the table to find the dataword 01.

Hamming Distance

One of the central concepts in coding for error control is the idea of the Hamming distance. The Hamming distance between two words (of the same size) is the number of differences between the corresponding bits. We show the Hamming distance between two words x and y as d(x, y). The Hamming distance can easily be found if wc apply the XOR operation on the two words and count the number of Is in the result. Note that the Hamming distance is

a value greater than zero.

Minimum Hamming Distance

Although the concept of the Hamming distance is the central point in dealing with error detection and correction codes, the measurement that is used for designing a code is the minimum Hamming distance. In a set of words, the minimum Hamming distance is the smallest Hamming distance between all possible pairs. We use d min to define the minimum Hamming distance in a coding scheme. To find this value, we find the Hamming distances between all words and select the smallest one.

Three Parameters

Before we continue with our discussion, we need to mention that any coding scheme needs to have at least three parameters: the codeword size n, the dataword size k, and the minimum Hamming distance d min . A coding scheme C is written as C(n, k) with a separate expression for dmin- For example, we can call our first coding scheme C(3, 2) with d min = 2 and our second coding scheme C(5, 2) with d min ::= 3.

Hamming Distance and Error

Before we explore the criteria for error detection or correction, let us discuss the relationship

between the Hamming distance and errors occurring during transmission. When a codeword

is corrupted during transmission, the Hamming distance between the sent and received codewords is the number of bits affected by the error. In other words, the Hamming distance between the received codeword and the sent codeword is the number of bits that are corrupted

during transmission. For example, if the codeword 00000 is sent and 01101 is received, 3 bits

are in error and the Hamming distance between the two is d(OOOOO, 01101) = 3.

Minimum Distance for Error Detection

Now let us find the minimum Hamming distance in a code if we want to be able to detect

up to s errors. If s errors occur during transmission, the Hamming distance between the sent codeword and received codeword is s. If our code is to detect up to s errors, the minimum distance between the valid codes must be s + 1, so that the received codeword does not match a valid codeword. In other words, if the minimum distance between all valid codewords is s + 1, the received codeword cannot be erroneously mistaken for another codeword. The distances are not enough (s + 1) for the receiver to accept it as valid. The error will be detected. We need to clarify a point here: Although a code with d min = s + 1 may be able to detect more than s errors in some special cases, only s or fewer errors are guaranteed to be detected.

LINEAR BLOCK CODES

Almost all block codes used today belong to a subset called linear block codes. The use of nonlinear block codes for error detection and correction is not as widespread because their structure makes theoretical analysis and implementation difficult. We therefore concentrate on linear block codes.

The formal definition of linear block codes requires the knowledge of abstract algebra (particularly Galois fields), which is beyond the scope of this book. We therefore give an informal definition. For our purposes, a linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.

Example 10.10

Let us see if the two codes we defined in Table 10.1 and Table 10.2 belong to the class of linear block codes.

1. The scheme in Table 10.1 is a linear block code because the result of XORing any codeword with any other codeword is a valid codeword. For example, the XORing of the second and third codewords creates the fourth one.
2. The scheme in Table 10.2 is also a linear block code. We can create all four codewords by XORing two other codewords.

Minimum Distance for Linear Block Codes

It is simple to find the minimum Hamming distance for a linear block code. The minimum Hamming distance is the number of Is in the nonzero valid codeword with the smallest number of Is.

Example 10.11

In our first code (Table 10.1), the numbers of Is in the nonzero codewords are 2, 2, and 2. So the minimum Hamming distance is d min = 2. In our second code (Table 10.2), the numbers of Is in the nonzero codewords are 3, 3, and 4. So in this code we have d min = 3.

Some Linear Block Codes

Let us now show some linear block codes. These codes are trivial because we can easily find the encoding and decoding algorithms and check their performances.

Simple Parity-Check Code

Perhaps the most familiar error-detecting code is the simple parity-check code. In this code, a k-bit dataword is changed to an n-bit codeword where n = k + 1. The extra bit,

Our first code (Table 10.1) is a parity-check code with k -= 2 and n =3. The code in Table 10.3 is also a parity-check code with k = 4 and n = 5.

Figure 10.10 shows a possible structure of an encoder (at the sender) and a decoder (at the receiver).

The encoder uses a generator that takes a copy of a 4-bit dataword (ao, aI' a2' and a3) and generates a parity bit roo The dataword bits and the parity bit create the 5-bit codeword. The parity bit that is added makes the number of Is in the codeword even.

This is normally done by adding the 4 bits of the dataword (modulo-2); the result is the parity bit. In other words,

If the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1.

In both cases, the total number of 1s in the codeword is even.

The sender sends the codeword which may be corrupted during transmission. The receiver receives a 5-bit word. The checker at the receiver does the same thing as the generator in the sender with one exception: The addition is done over all 5 bits. The result, which is called the syndrome, is just 1 bit. The syndrome is 0 when the number of Is in the received codeword is even; otherwise, it is 1.

The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the dataword; if the syndrome is 1, the data portion of the received codeword is discarded. The dataword is not created.

Let us look at some transmission scenarios. Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver. We examine five cases:

1. No error occurs; the received codeword is 10111. The syndrome is O. The dataword 1011 is created.
2. One single-bit error changes aI' The received codeword is 10011. The syndrome is 1. No dataword is created.
3. One single-bit error changes roo The received codeword is 10110. The syndrome is 1. No dataword is created. Note that although none of the dataword bits are corrupted, no dataword is created because the code is not sophisticated enough to show the position of the corrupted bit.
4. An error changes ro and a second error changes a3' The received codeword is 00110. The syndrome is O. The dataword 0011 is created at the receiver. Note that here the dataword is

wrongly created due to the syndrome value. The simple parity-check decoder cannot detect an
even number of errors. The errors cancel each other out and give the syndrome a value of O.
5. Three bits-a3, az, and aI-are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.
A simple parity-check code can detect an odd number of errors.
A better approach is the two-dimensional parity check. In this method, the dataword is organized in a table (rows and columns). In Figure 10.11, the data to be sent, five
7-bit bytes, are put in separate rows. For each row and each column, 1 parity-check bit is calculated. The whole table is then sent to the receiver, which finds the syndrome for each
row and each column. As Figure 10.11 shows, the two-dimensional parity check can detect up to three errors that occur anywhere in the table (arrows point to the locations of the created nonzero syndromes). However, errors affecting 4 bits may not be detected.

Hamming Codes
Now let us discuss a category of error-correcting codes called Hamming codes. These codes were originally designed with d min = 3, which means that they can detect up to two
errors or correct one single error. Although there are some Hamming codes that can correct more than one error, our discussion focuses on the single-bit error-correcting code.
First let us find the relationship between n and k in a Hamming code. We need to
choose an integer m >= 3. The values of nand k are then calculated from mas n = 2 m - 1 and k ::: n - m. The number of check bits r = m.
All Hamming codes discussed in this book have d min = 3.
The relationship between m and n in these codes is n = 2 m - 1.
For example, if m = 3, then n ::: 7 and k::: 4. This is a Hamming code C(7, 4) with d min = 3.
Table 10.4 shows the datawords and codewords for this code.

CYCLIC CODES
Cyclic codes are special linear block codes with one extra property. In a cyclic code, if
a codeword is cyclically shifted (rotated), the result is another codeword. For example,
if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.
In this case, if we call the bits in the first word ao to a6' and the bits in the second word
b o to b 6, we can shift the bits by using the following:
In the rightmost equation, the last bit of the first word is wrapped around and
becomes the first bit of the second word.
Cyclic Redundancy Check
We can create cyclic codes to correct errors. However, the theoretical background
required is beyond the scope of this book. In this section, we simply discuss a category
of cyclic codes called the cyclic redundancy check (CRC) that is used in networks

such as LANs and WANs.

In the encoder, the dataword has k bits (4 here); the codeword has n bits (7 here). The size of the dataword is augmented by adding n - k (3 here) Os to the right-hand side of the word. The n-bit result is fed into the generator. The generator uses a divisor of size n - k + I (4 here), predefined and agreed upon. The generator divides the augmented dataword by the divisor (modulo-2 division). The quotient ofthe division is discarded; the remainder (r2rl ro) is appended to the dataword to create the codeword.

The decoder receives the possibly corrupted codeword. A copy of all n bits is fed to the checker which is a replica of the generator. The remainder produced by the checker is a syndrome of n - k (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function. If the syndrome bits are all as, the 4 leftmost bits of the codeword are accepted as the dataword (interpreted as no error); otherwise, the 4 bits are discarded (error).

Encoder

Let us take a closer look at the encoder. The encoder takes the dataword and augments it with n - k number of as. It then divides the augmented dataword by the divisor, as shown in Figure 10.15.

The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. However, as mentioned at the beginning of the chapter, in this case addition and subtraction are the same. We use the XOR operation to do both.

As in decimal division, the process is done step by step. In each step, a copy of the divisor is XORed with the 4 bits of the dividend. The result of the XOR operation (remainder) is 3 bits (in this case), which is used for the next step after 1 extra bit is pulled down to make it 4 bits long. There is one important point we need to remember in this type of division. If the leftmost bit of the dividend (or the part used in each step) is 0, the step cannot use the regular divisor; we need to use an all-Os divisor.

When there are no bits left to pull down, we have a result. The 3-bit remainder forms the check bits (r2' rl' and ro). They are appended to the dataword to create the codeword.

Decoder

The codeword can change during transmission. The decoder does the same division process as the encoder. The remainder of the division is the syndrome. If the syndrome is all Os, there is no error; the dataword is separated from the received codeword and accepted. Otherwise, everything is discarded. Figure 10.16 shows two cases: The left-hand figure shows the value of syndrome when no error has occurred; the syndrome is 000. The right-hand part of the figure shows the case in which there is one single error. The syndrome is not all Os (it is Oil).

Divisor

You may be wondering how the divisor] 011 is chosen. Later in the chapter we present some criteria, but in general it involves abstract algebra.

Hardware Implementation

One of the advantages of a cyclic code is that the encoder and decoder can easily and cheaply be implemented in hardware by using a handful of electronic devices. Also, a hardware implementation increases the rate of check bit and syndrome bit calculation. In this section, we try to show, step by step, the process. The section, however, is optional and does not affect the understanding of the rest of the chapter.

Divisor

Let us first consider the divisor. We need to note the following points:

1. The divisor is repeatedly XORed with part of the dividend.
2. The divisor has n - k + 1 bits which either are predefined or are all Os. In other words, the bits do not change from one dataword to another. In our previous example, the divisor bits were either 1011 or 0000. The choice was based on the leftmost bit of the part of the augmented data bits that are active in the XOR operation.
3. A close look shows that only n - k bits of the divisor is needed in the XOR operation. The leftmost bit is not needed because the result of the operation is always 0, no matter what the value of this bit. The reason is that the inputs to this XOR operation are either both Os or both 1s. In our previous example, only 3 bits, not 4, is actually used in the XOR operation.

Using these points, we can make a fixed (hardwired) divisor that can be used for a cyclic code if we know the divisor pattern. Figure 10.17 shows such a design for our previous example. We have also shown the XOR devices used for the operation.

At each clock tick, shown as different times, one of the bits from the augmented dataword is used in the XOR process. If we look carefully at the design, we have seven steps here, while in the paper-and-pencil method we had only four steps. The first three steps have been added here to make each step equal and to make the design for each step the same. Steps 1, 2, and 3 push the first 3 bits to the remainder registers; steps 4, 5, 6, and 7 match the paper-and-pencil design. Note that the values in the remainder register in steps 4 to 7 exactly match the values in the paper-and-pencil design. The final remainder is also the same.

The above design is for demonstration purposes only. It needs simplification to be practical. First, we do not need to keep the intermediate values of the remainder bits; we need only the final bits. We therefore need only 3 registers instead of 24. After the XOR operations, we do not need the bit values of the previous remainder. Also, we do

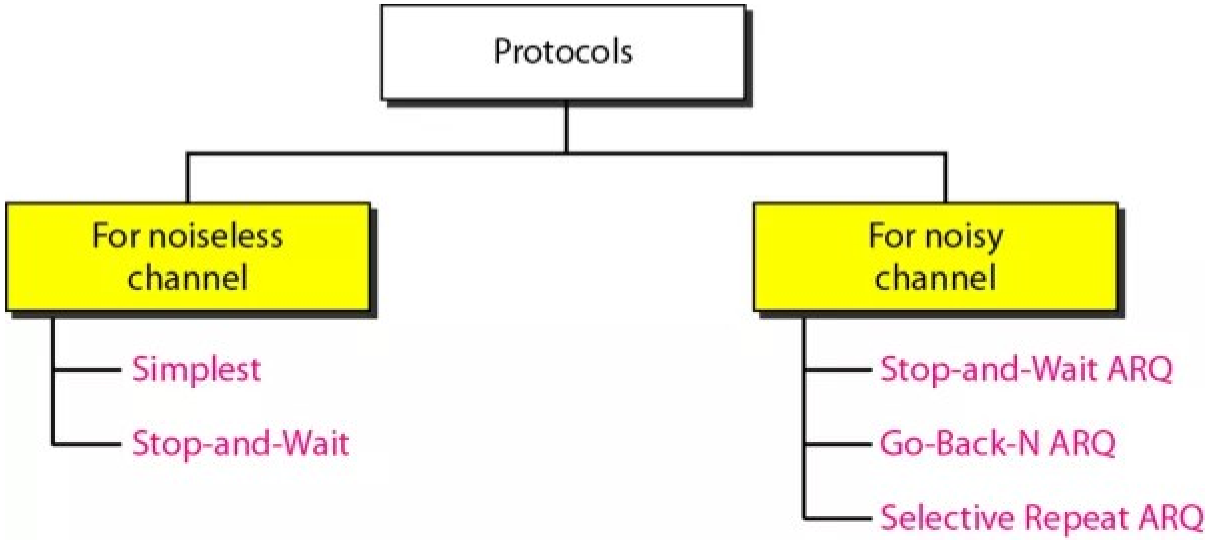
2.6 Checksum

The last error detection method we discuss here is called the checksum. The checksum is used in the Internet by several protocols although not at the data link layer. However, we briefly discuss it here to complete our discussion on error checking. Like linear and cyclic codes, the checksum is based on the concept of redundancy. Several protocols still use the checksum for error detection as we will see in future chapters, although the tendency is to replace it with a CRe. This means that the CRC is also used in layers other than the data link layer.

One's Complement

The previous example has one major drawback. All of our data can be written as a 4-bit word (they are less than 15) except for the checksum. One solution is to use one's complement arithmetic. In this arithmetic, we can represent unsigned numbers between 0 and 2 n - 1 using only n bits. t If the number has more than n bits, the extra leftmost bits need to be added to the n rightmost bits (wrapping). In one's complement arithmetic, a negative number can be represented by inverting all bits (changing a 0 to a 1 and a 1 to a 0). This is the same as subtracting the number from 2 n – 1.

2.9 Protocols

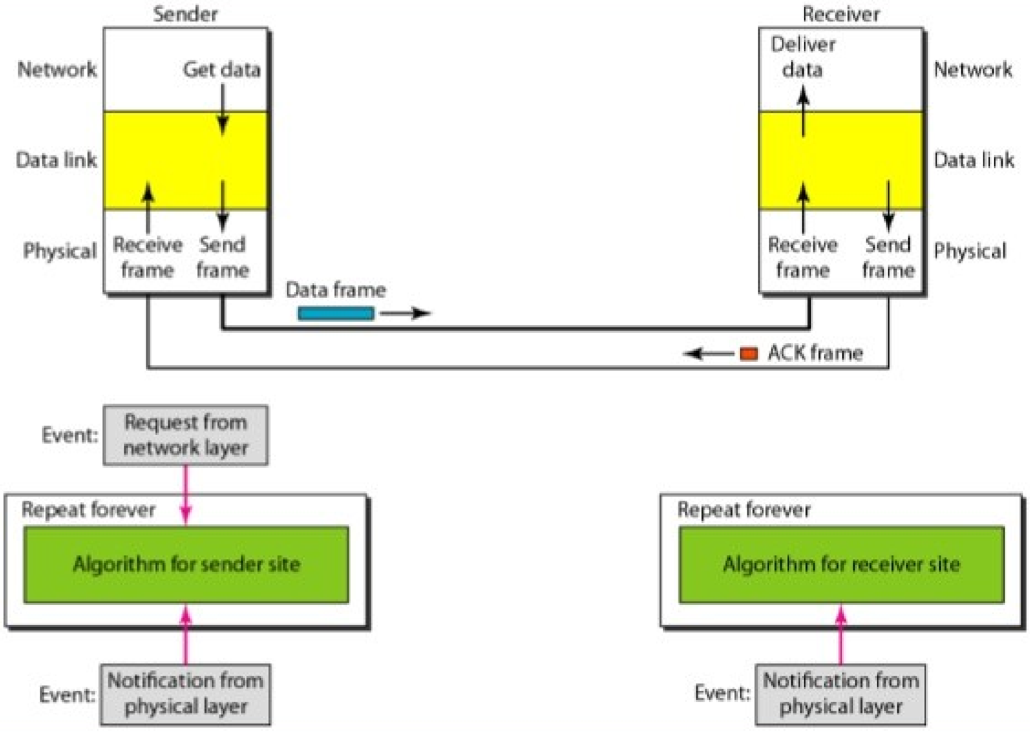


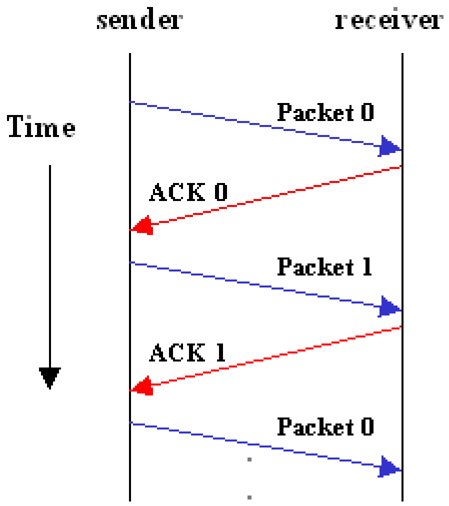
2.9.1 Stop and wait

If data frames arrive at the receiver site faster than they can be processed, the frames must be stored until their use. Normally, the receiver does not have enough storage space, especially if it is receiving data from many sources. This may result in either the discarding of frames or denial of service. To prevent the receiver from becoming overwhelmed with frames,we somehow need to tell the sender to slow down. There must be feedback from the receiver to the sender.

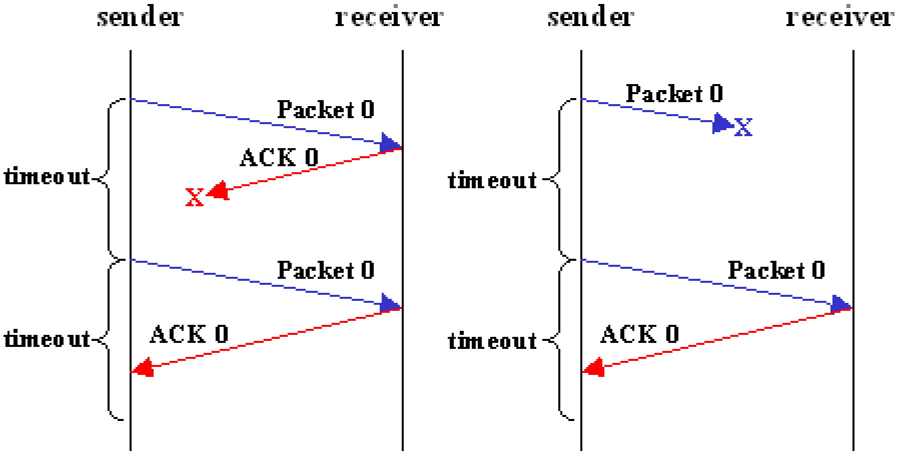
The protocol we discuss now is called the Stop-and-Wait Protocol because the sender sends one frame, stops until it receives confirmation from the receiver (okay to go ahead), and then sends the next frame. We still have unidirectional communication for data frames, but auxiliary ACK frames (simple tokens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

Figure 11.8 *Design of Stop-and-Wait Protocol*





After transmitting one packet, the sender waits for an ***acknowledgment (ACK)*** from the receiver before transmitting the next one. In this way, the sender can recognize that the previous packet is transmitted successfully and we could say "stop-n-wait" guarantees reliable transfer between nodes.
To support this feature, the sender keeps a record of each packet it sends.
Also, to avoid confusion caused by delayed or duplicated ACKs, "stop-n-wait" sends each packets with unique sequence numbers and receives that numbers in each ACKs.



If the sender doesn't receive ACK for previous sent packet after a certain period of time, the sender *times out* and *retransmits* that packet again. There are two cases when the sender doesn't receive ACK; One is when the ACK is lost and the other is when the frame itself is not transmitted.
To support this feature, the sender keeps timer per each packet.

2.9.2 Go-Back-N ARQ(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. In this section, we discuss one protocol that can achieve this goal; in the next section, we discuss a second. The first is called Go-Back-N Automatic Repeat Request (the rationale for the name will become clear later). In this protocol we can send several frames before receiving acknowledgments; we keep a copy of these frames until the acknowledg- ments arrive.

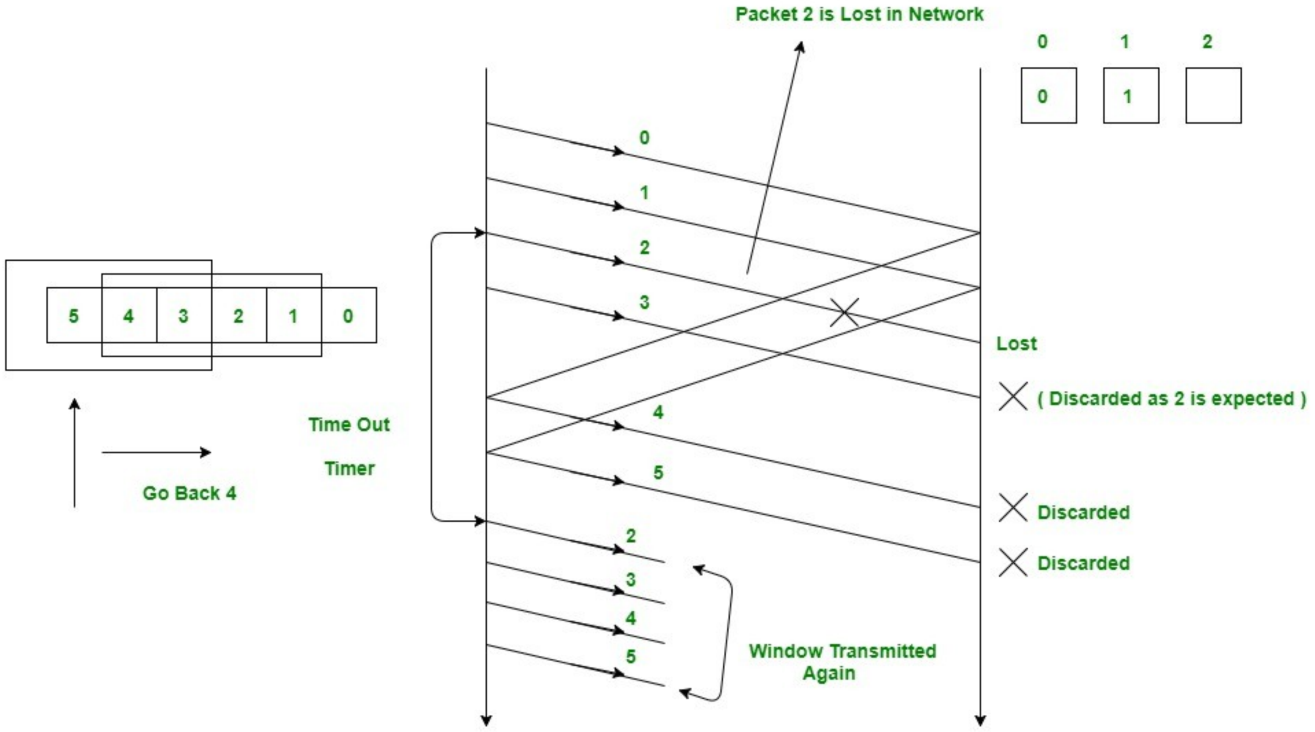
Sequence Numbers

Frames from a sending station are numbered sequentially. However, because we need to include the sequence number of each frame in the header, we need to set a limit. If the header of the frame allows m bits for the sequence number, the sequence numbers range from 0 to 2 m - 1. For example, if m is 4, the only sequence numbers are 0 through 15 inclusive. However, we can repeat the sequence. So the sequence numbers are

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

In other words, the sequence numbers are modulo-2 m .

Now what exactly happens in GBN, we will explain with a help of example. Consider the diagram given below. We have sender window size of 4. Assume that we have lots of sequence numbers just for the sake of explanation. Now the sender has sent the packets 0, 1, 2 and 3. After acknowledging the packets 0 and 1, receiver is now expecting packet 2 and sender window has also slided to further transmit the packets 4 and 5. Now suppose the packet 2 is lost in the network, Receiver will discard all the packets which sender has transmitted after packet 2 as it is expecting sequence number of 2. On the sender side for every packet send there is a time out timer which will expire for packet number 2. Now from the last transmitted packet 5 sender will go back to the packet number 2 in the current window and transmit all the packets till packet number 5. That’s why it is called Go Back N. Go back means sender has to go back N places from the last transmitted packet in the unacknowledged window and not from the point where the packet is lost.



2.9.3 Selective Repeat AQR

This protocol(SRP) is mostly identical to GBN protocol, except that buffers are used and the receiver, and the sender, each maintain a window of size. SRP works better when the link is very unreliable. Because in this case, retransmission tends to happen more frequently, selectively retransmitting frames is more efficient than retransmitting all of them. SRP also requires full duplex link. backward acknowledgements are also in progress.

•Sender’s Windows (Ws) = Receiver’s Windows (Wr).

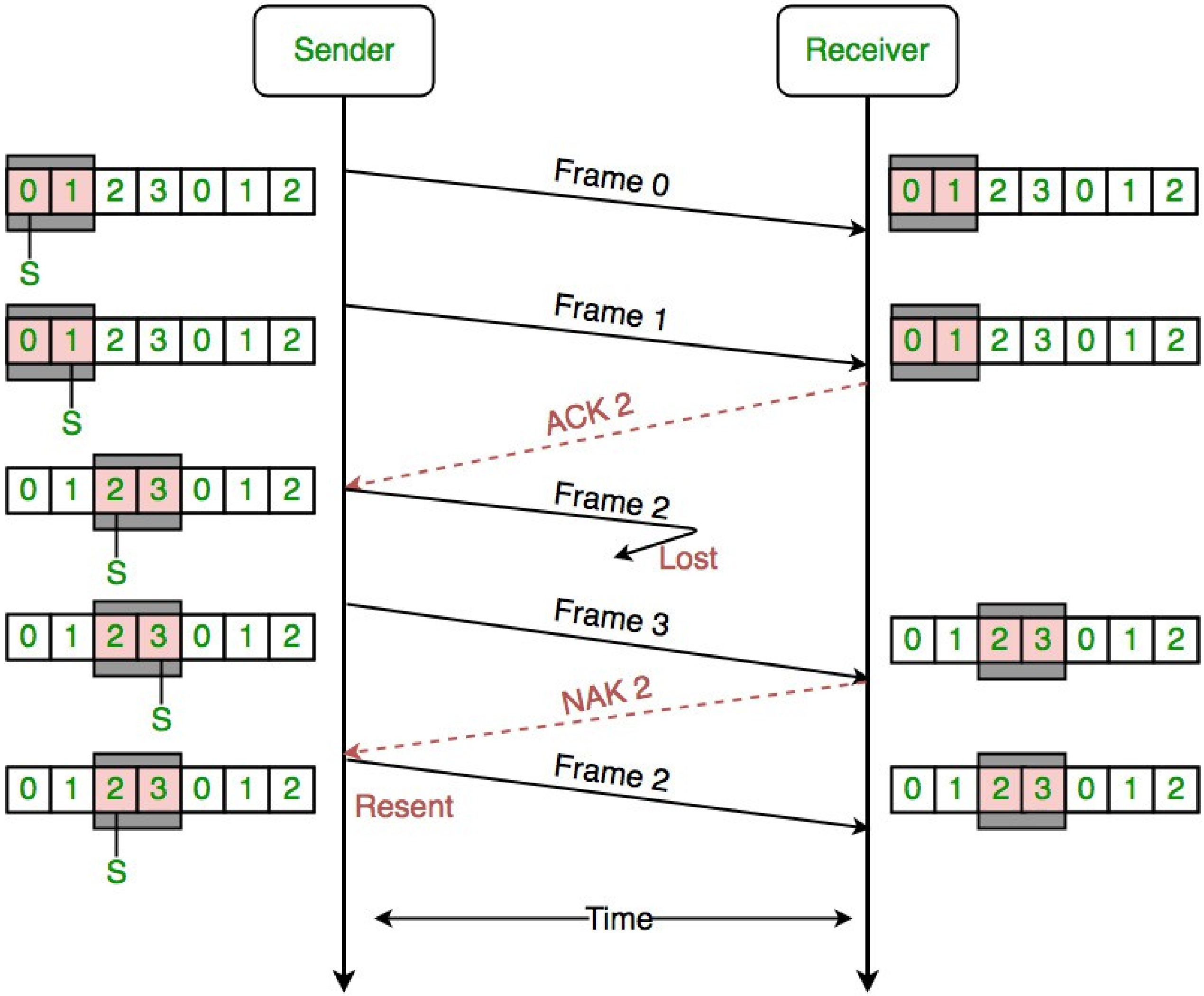
•Window size should be less than or equal to half the sequence number in SR protocol. This is to avoid packets being recognized incorrectly. If the windows size is greater than half the sequence number space, then if an ACK is lost, the sender may send new packets that the receiver believes are retransmissions.

•Sender can transmit new packets as long as their number is with W of all unACKed packets.

•Sender retransmit un-ACKed packets after a timeout – Or upon a NAK if NAK is employed.

•Receiver ACKs all correct packets.

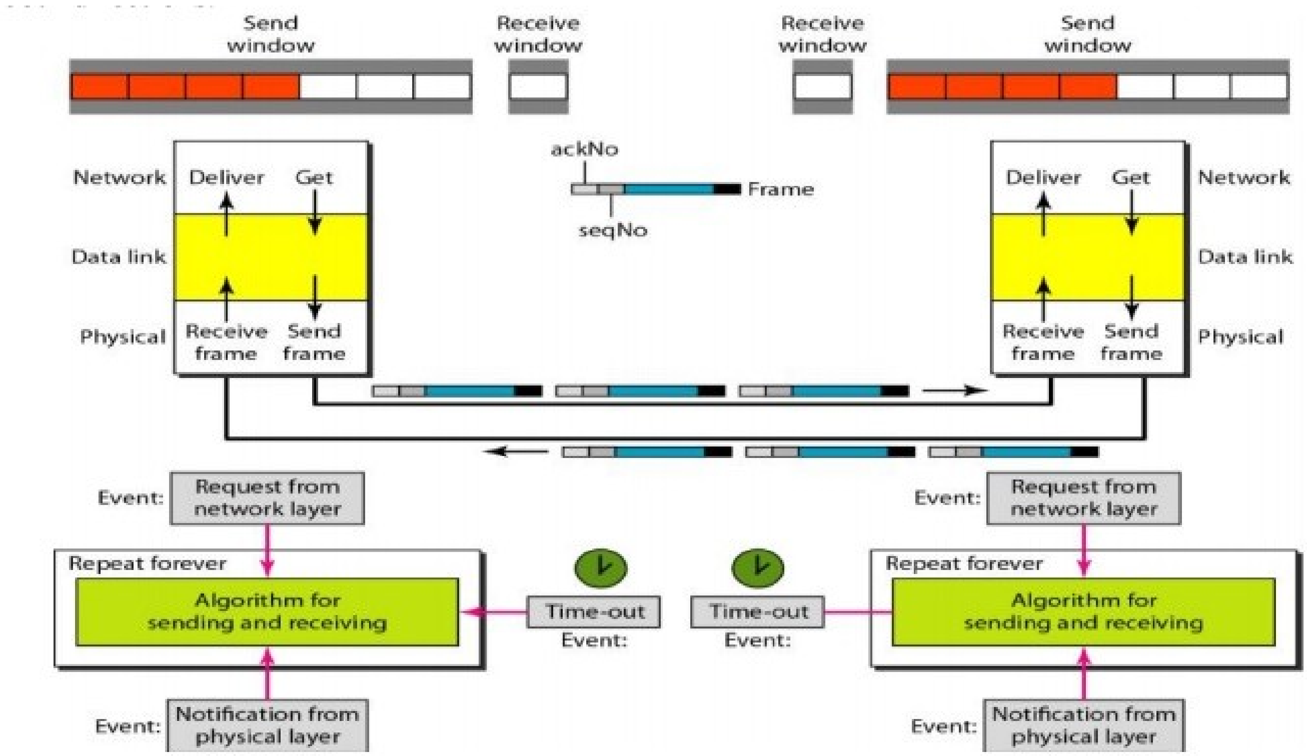
•Receiver stores correct packets until they can be delivered in order to the higher layer.



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

2.9.5 Piggy Backing

The three protocols we discussed in this section are all unidirectional: data frames flow in only one direction although control information such as ACK and NAK frames can travel in the other direction. In real life, data frames are normally flowing in both directions: from node A to node B and from node B to node A. This means that the control information also needs to flow in both directions. A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A. We show the design for a Go-Back-N ARQ using piggybacking in Figure 11.24. Note that each node now has two windows: one send window and one receive window. Both also need to use a timer. Both are involved in three types of events: request, arrival, and time-out. However, the arrival event here is complicated; when a frame arrives, the site needs to handle control information as well as the frame itself. Both of these concerns must be taken care of in one event, the arrival event. The request event uses only the send window at each site; the arrival event needs to use both windows. An important point about piggybacking is that both sites must use the same algo- rithm. This algorithm is complicated because it needs to combine two arrival events into one. We leave this task as an exercise.



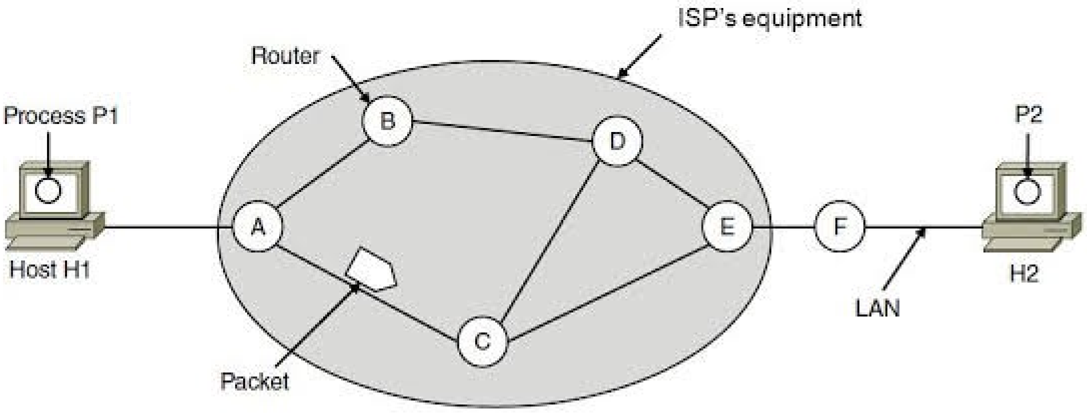
UNIT-3

3.1 Design Issue of Network Layer

3.1.1 Store-and-Forward Packet Switching

Before starting to explain the details of the network layer, it is worth restating the context in which the network layer protocols operate. This context can be seen in Fig. 5-1. The major components of the network are the ISP’s equipment (routers connected by transmission lines), shown inside the shaded oval, and the customers’ equipment, shown outside the oval. Host H1 is directly connected to one of the ISP’s routers, A, perhaps as a home computer that is plugged into a DSL modem. In contrast, H2 is on a LAN, which might be an office Ethernet, with a router, F, owned and operated by the customer. This router has a leased line to the ISP’s equipment. We have shown F as being outside the oval because it does not belong to the ISP. For the purposes of this chapter, however, routers on customer premises are considered part of the ISP network because they run the same algorithms as the ISP’s routers (and our main concern here is algorithms).

Store-and-Forward Packet Switching



The environment of the network layer protocols.

This equipment is used as follows. 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, as we have seen in previous chapters.

3.1.2 Services Provided to the Transport Layer

The network layer provides services to the transport layer at the network layer/transport layer interface. An important question is precisely what kind of services the network layer provides to the transport layer. The services need to be carefully designed with the following goals in mind:

1. The services should be independent of the router technology.
2. The transport layer should be shielded from the number, type, and topology of the routers present.
3. The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Given these goals, the designers of the network layer have a lot of freedom in writing detailed specifications of the services to be offered to the transport layer. This freedom often degenerates into a raging battle between two warring factions. The discussion centers on whether the network layer should provide connection- oriented service or connectionless service.

One camp (represented by the Internet community) argues that the routers' job is moving packets around and nothing else. In this view (based on 40 years of experience with a real computer network), the network is inherently unreliable, no matter how it is designed. Therefore, the hosts should accept this fact and do error control (i.e., error detection and correction) and flow control themselves.

This viewpoint leads to the conclusion that the network service should be con-nectionless, with primitives SEND PACKET and RECEIVE PACKET and little else. In particular, no packet ordering and flow control should be done, because the hosts are going to do that anyway and there is usually little to be gained by doing it twice. This reasoning is an example of the end-to-end argument, a design principle that has been very influential in shaping the Internet (Saltzer et al., 1984). Furthermore, each packet must carry the full destination address, because each packet sent is carried independently of its predecessors, if any. The other camp (represented by the telephone companies) argues that the net-work should provide a reliable, connection-oriented service. They claim that 100 years of successful experience with the worldwide telephone system is an excellent guide. In this view, quality of service is the dominant factor, and without connections in the network, quality of service is very difficult to achieve, especially for real-time traffic such as voice and video.

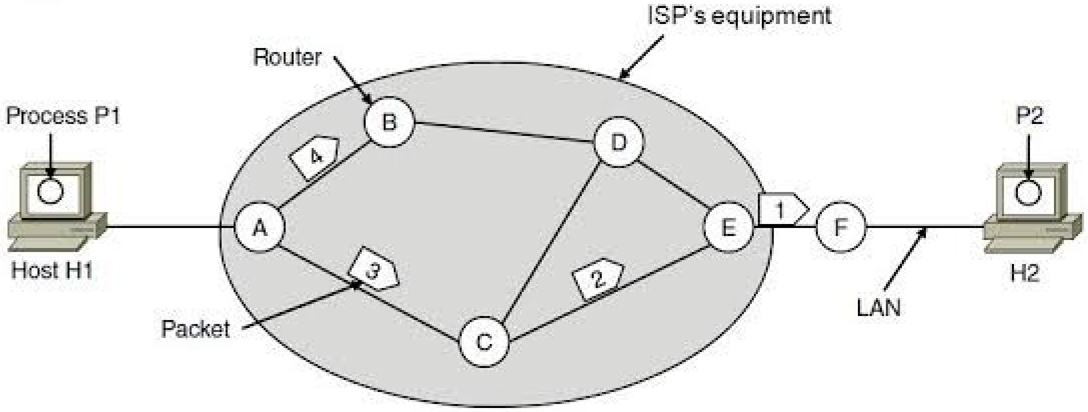
Even after several decades, this controversy is still very much alive. Early, widely used data networks, such as X.25 in the 1970s and its successor Frame Relay in the 1980s, were connection-oriented. However, since the days of the ARPANET and the early Internet, connectionless network layers have grown tremendously in popularity. The IP protocol is now an ever-present symbol of success. It was undeterred by a connection-oriented technology called ATM that was developed to overthrow it in the 1980s; instead, it is ATM that is now found in niche uses and IP that is taking over telephone networks. Under the covers, however, the Internet is evolving connection-oriented features as quality of service becomes more important. Two examples of connection-oriented technologies are MPLS (MultiProtocol Label Switching), which we will describe in this chapter, and VLANs, which we saw in Chap. 4. Both technologies are widely used.

3.1.3 Implementation of Connectionless Service

Having looked at the two classes of service the network layer can provide to its users, it is time to see how this layer works inside. Two different organizations are possible, depending on the type of service offered. 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. 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), in analogy with the physical circuits set up by the telephone system, and the network is called a virtual-circuit network. In this section, we will examine datagram networks; in the next one, we will examine virtual-circuit networks.

Let us now see how a datagram network works. Suppose that the process P1 in Fig. 5-2 has a long message for P2. It hands the message to the transport layer, with instructions to deliver it to process P2 on host H2. The transport layer code runs on H1, typically within the operating system. It prepends a transport header to the front of the message and hands the result to the network layer, probably just another procedure within the operating system.

Implementation of Connectionless Service



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

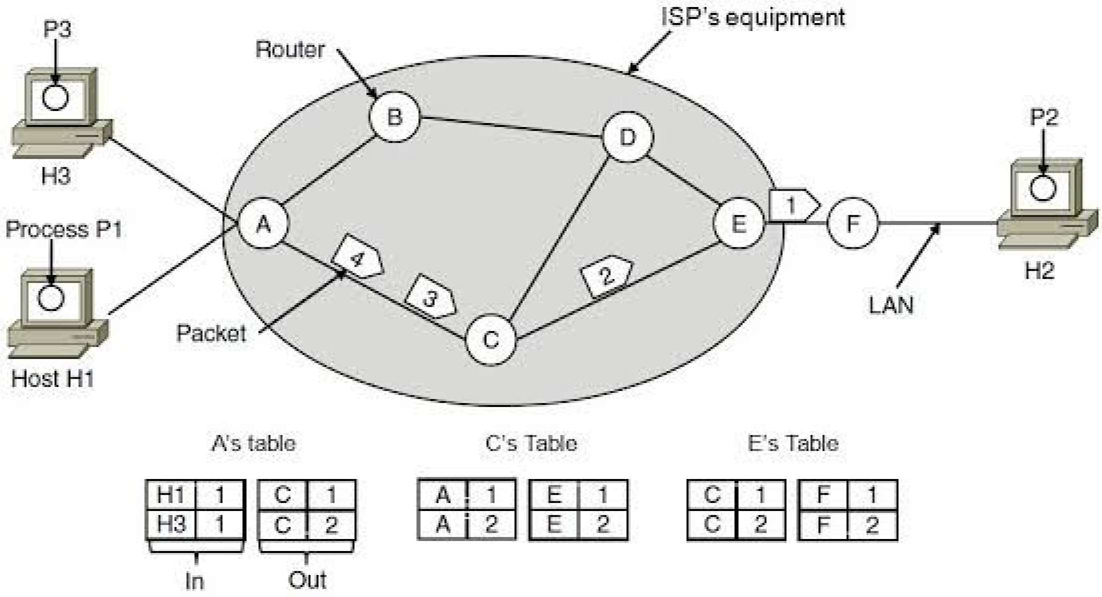
Routing within a datagram network

Let us assume for this example that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4, and send each of them in turn to router A using some point-to-point protocol, for example, PPP. At this point the ISP takes over. Every router has an internal table telling it where to send packets for each of the possible destinations. Each table entry is a pair consisting of a destination and the outgoing line to use for that destination. Only directly connected lines can be used. For example, in Fig. 5-2, A has only two outgoing lines—to B and to C—so every incoming packet must be sent to one of these routers, even if the ultimate destination is to some other router. A's initial routing table is shown in the figure under the label ''initially.'' At A, packets 1, 2, and 3 are stored briefly, having arrived on the incoming link and had their checksums verified. Then each packet is forwarded according to A's table, onto the outgoing link to C within a new frame. Packet 1 is then forwarded to E and then to F. When it gets to F, it is sent within a frame over the LAN to H2. Packets 2 and 3 follow the same route. However, something different happens to packet 4. When it gets to A it is sent to router B, even though it is also destined for F. For some reason, A decided to send packet 4 via a different route than that of the first three packets. Perhaps it has learned of a traffic jam somewhere along the ACE path and updated its routing table, as shown under the label “later.” The algorithm that manages the tables and makes the routing decisions is called the routing algorithm. Routing algorithms are one of the main topics we will study in this chapter. There are several different kinds of them, as we will see. IP (Internet Protocol), which is the basis for the entire Internet, is the dominant example of a connectionless network service. Each packet carries a destina- tion IP address that routers use to individually forward each packet. The addresses are 32 bits in IPv4 packets and 128 bits in IPv6 packets. We will describe IP in much detail later in this chapter.

3.1.4 Implementation of Connection-Oriented Service

For connection-oriented service, we need a virtual-circuit network. Let us see how that works. The idea behind virtual circuits is to avoid having to choose a new route for every packet sent, as in Fig. 5-2. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to. As an example, consider the situation shown in Fig. 5-3. Here, host H1 has established connection 1 with host H2. This connection is remembered as the first entry in each of the routing tables. The first line of A’s table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1.

Implementation of Connection-Oriented Service



A's table

In		Out	
H1	1	C	1
H3	1	C	2

C's Table

A	1	E	1
A	2	E	2

E's Table

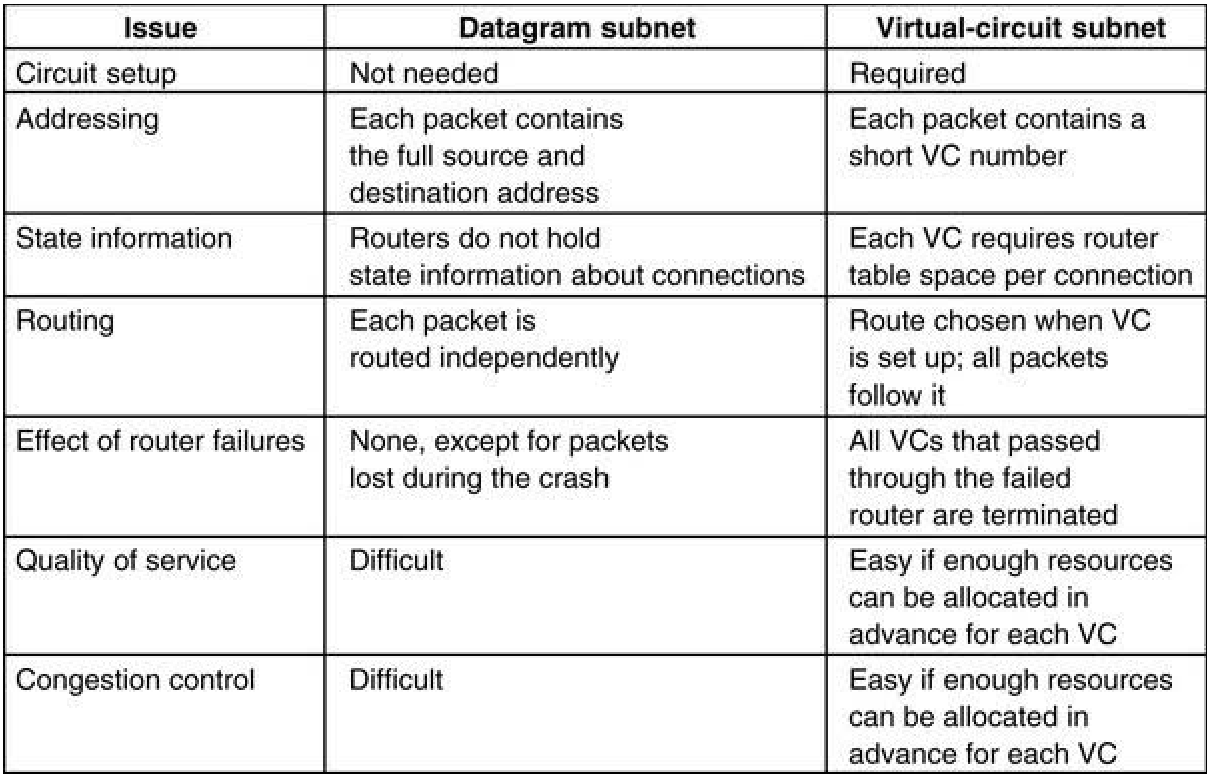
C	1	F	1
C	2	F	2

Routing within a virtual-circuit network

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 (MultiProtocol Label Switching). It is used within ISP networks in the Internet, with IP packets wrapped in an MPLS header having a 20-bit connection identifier or label. MPLS is often hidden from customers, with the ISP establishing long-term connections for large amounts of traffic, but it is increasingly being used to help when quality of service is important but also with other ISP traffic management tasks. We will have more to say about MPLS later in this chapter.

3.1.5 Comparison of Virtual-Circuit and Datagram Networks



Issue	Datagram subnet	Virtual-circuit subnet
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

3.2 ROUTING ALGORITHMS

The main function of the network layer is routing packets from the source machine to the destination machine. In most networks, packets will require multiple hops to make the journey. The only notable exception is for broadcast networks, but even here routing is an issue if the source and destination are not on the same network segment. The algorithms that choose the routes and the data structures that they use are a major area of network layer design. The routing algorithm is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. If the network uses datagrams internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the network uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the already established route. The latter case is sometimes called session routing because a route remains in force for an entire session (e.g., while logged in over a VPN).

It is sometimes useful to make a distinction between routing, which is making the decision which routes to use, and forwarding, which is what happens when a packet arrives. One can think of a router as having two processes inside it. One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables. This process is forwarding. The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play.

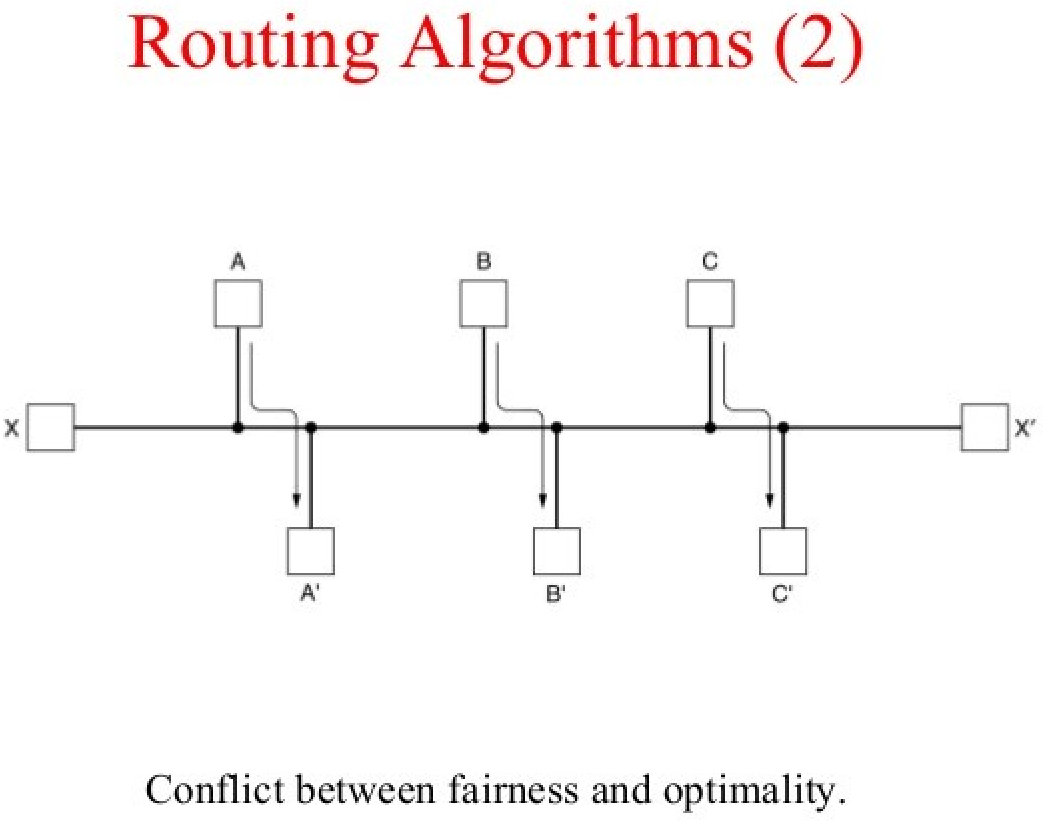
Regardless of whether routes are chosen independently for each packet sent or only when new connections are established, certain properties are desirable in a routing algorithm: correctness, simplicity, robustness, stability, fairness, and efficiency. Correctness and simplicity hardly require comment, but the need for robustness may be less obvious at first. Once a major network comes on the air, it may be expected to run continuously for years without system-wide failures. During that period there will be hardware and software failures of all kinds. Hosts, routers, and lines will fail repeatedly, and the topology will change many times. The routing algorithm should be able to cope with changes in the topology and traffic without requiring all jobs in all hosts to be aborted. Imagine the havoc if the network needed to be rebooted every time some router crashed!

Stability is also an important goal for the routing algorithm. There exist routing algorithms that never converge to a fixed set of paths, no matter how long they run. A stable algorithm reaches equilibrium and stays there. It should converge quickly too, since communication may be disrupted until the routing algorithm has reached equilibrium.

Fairness and efficiency may sound obvious—surely no reasonable person would oppose them—but as it turns out, they are often contradictory goals. As a simple example of this conflict, look at Fig. 5-5. Suppose that there is enough traffic between A and A′, between B and B′, and between C and C′ to saturate the horizontal links. To maximize the total flow, the X to X′ traffic should be shut off altogether. Unfortunately, X and X′ may not see it that way. Evidently, some compromise between global efficiency and fairness to individual connections is needed.

Before we can even attempt to find trade-offs between fairness and efficiency, we must decide what it is we seek to optimize. Minimizing the mean packet delay is an obvious candidate to send traffic through the network effectively, but so is maximizing total network throughput. Furthermore, these two goals are also in conflict, since operating any queueing system near capacity implies a long queueing delay. As a compromise, many networks attempt to minimize the distance a packet must travel, or simply reduce the number of hops a packet must make. Either choice tends to improve the delay and also reduce the amount of bandwidth consumed per packet, which tends to improve the overall network throughput as well.

Routing algorithms can be grouped into two major classes: nonadaptive and adaptive. Nonadaptive algorithms do not base their routing decisions on any



Routing Algorithms (2)

Conflict between fairness and optimality.

measurements or estimates of the current topology and traffic. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off- line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing. Because it does not respond to failures, static routing is mostly useful for situations in which the routing choice is clear. For example, router F in Fig. 5-3 should send packets headed into the network to router E regardless of the ultimate destination.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well. These dynamic routing algorithms differ in where they get their information (e.g., locally, from adjacent routers, or from all routers), when they change the routes (e.g., when the topology changes, or every ΔT seconds as the load changes), and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

In the following sections, we will discuss a variety of routing algorithms. The algorithms cover delivery models besides sending a packet from a source to a destination. Sometimes the goal is to send the packet to multiple, all, or one of a set of destinations. All of the routing algorithms we describe here make decisions based on the topology; we defer the possibility of decisions based on the traffic levels to Sec 5.3.

3.2.1 The Optimality Principle

Before we get into specific algorithms, it may be helpful to note that one can make a general statement about optimal routes without regard to network topology or traffic. This statement is known as the optimality principle (Bellman, 1957). 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 route. To see this, call the part of the route from I to J r 1 and the rest of the route r 2 . If a route better than r 2 existed from J to K, it could be concatenated with r 1 to improve the route from I to K, contradicting our statement that r 1 r 2 is optimal.

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 and is illustrated in Fig. 5-6(b), where the distance metric is the number of hops. The goal of all routing algorithms is to discover and use the sink trees for all routers.

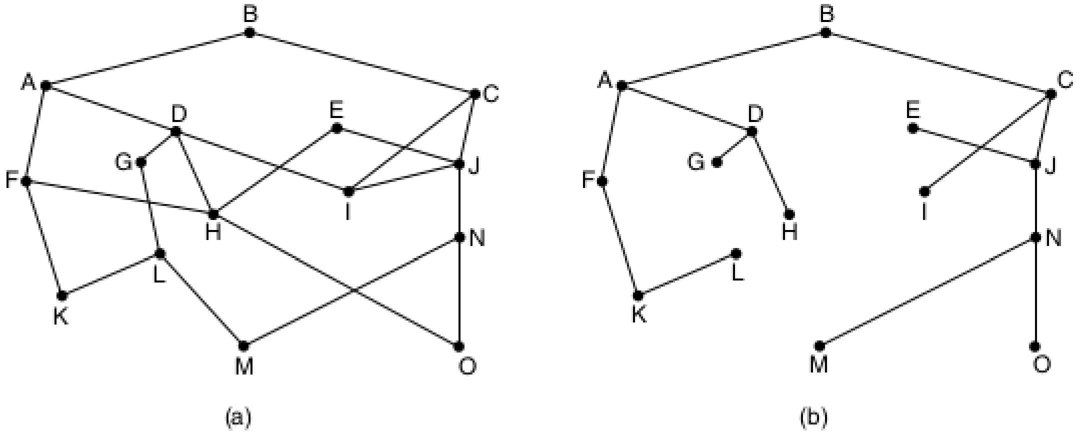


Figure 5-6. (a) A network. (b) A sink tree for router *B*.

Note that a sink tree is not necessarily unique; other trees with the same path lengths may exist. If we allow all of the possible paths to be chosen, the tree becomes a more general structure called a DAG (Directed Acyclic Graph). DAGs have no loops. We will use sink trees as a convenient shorthand for both cases. Both cases also depend on the technical assumption that the paths do not interfere with each other so, for example, a traffic jam on one path will not cause another path to divert.

Since a sink tree is indeed a tree, it does not contain any loops, so each packet will be delivered within a finite and bounded number of hops. In practice, life is not quite this easy. Links and routers can go down and come back up during operation, so different routers may have different ideas about the current topology. Also, we have quietly finessed the issue of whether each router has to individually acquire the information on which to base its sink tree computation or whether this information is collected by some other means. We will come back to these issues shortly. Nevertheless, the optimality principle and the sink tree provide a benchmark against which other routing algorithms can be measured.

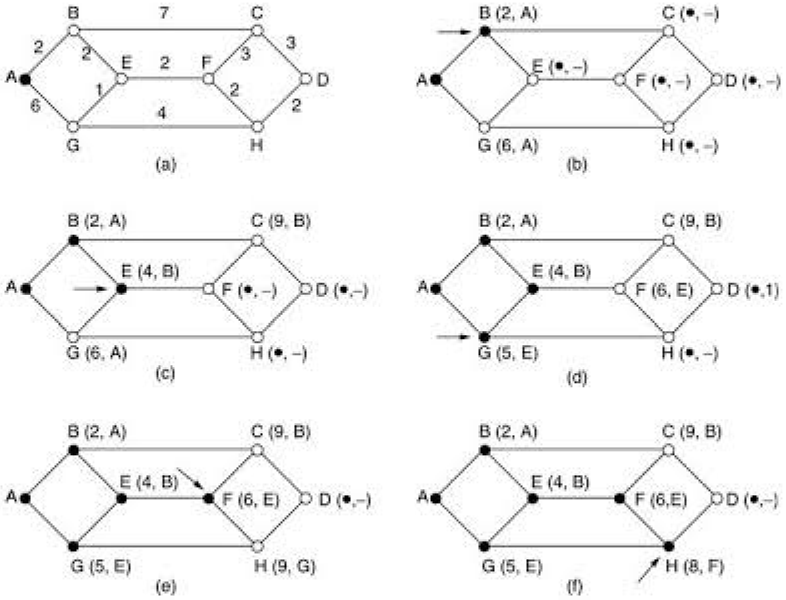
3.2.2 Shortest Path Algorithm

Let us begin our study of routing algorithms with a simple technique for com- puting optimal paths given a complete picture of the network. These paths are the ones that we want a distributed routing algorithm to find, even though not all routers may know all of the details of the network.

The idea is to build a graph of the network, with each node of the graph representing a router and each edge of the graph representing a communication line, or link. To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

The concept of a shortest path deserves some explanation. One way of measuring path length is the number of hops. Using this metric, the paths ABC and ABE in Fig. 5-7 are equally long. Another metric is the geographic distance in kilometers, in which case ABC is clearly much longer than ABE (assuming the figure is drawn to scale).

Shortest Path Routing



The first 5 steps used in computing the shortest path from A to D.
The arrows indicate the working node.

However, many other metrics besides hops and physical distance are also possible. For example, each edge could be labeled with the mean delay of a standard test packet, as measured by hourly runs. With this graph labeling, the shortest path is the fastest path rather than the path with the fewest edges or kilometers.

In the general case, the labels on the edges could be computed as a function of the distance, bandwidth, average traffic, communication cost, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.

Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to Dijkstra (1959) and finds the shortest paths between a source and all destinations in the network. Each node is labeled (in parentheses) with its distance from the source node along the best known path. The distances must be non-negative, as they will be if they are based on real quantities like bandwidth and delay. Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

To illustrate how the labeling algorithm works, look at the weighted, undirected graph of Fig. 5-7(a), where the weights represent, for example, distance. We want to find the shortest path from A to D. We start out by marking node A as permanent, indicated by a filled-in circle. Then we examine, in turn, each of the nodes adjacent to A (the working node), relabeling each one with the distance to A. Whenever a node is relabeled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. If the network had more than one shortest path from A to D and we wanted to find all of them, we would need to remember all of the probe nodes that could reach a node with the same distance.

Having examined each of the nodes adjacent to A, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. 5-7(b). This one becomes the new working node.

We now start at B and examine all nodes adjacent to it. If the sum of the label on B and the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled.

After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively labeled node with the smallest value. This node is made permanent and becomes the working node for the next round. Figure 5-7 shows the first six steps of the algorithm.

To see why the algorithm works, look at Fig. 5-7(c). At this point we have just made E permanent. Suppose that there were a shorter path than ABE, say