ALAGAPPA UNIVERSITY

[Accredited with A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]
KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION

Master of Computer Applications
31534

COMPUTER NETWORKS

III - Semester

ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003

Directorate of Distance Education

Master of Computer Applications

III - Semester

31534

COMPUTER NETWORKS

Author:
Dr. K. Thamodaran
Assistant Professor,
PG and Research Dept. of Computer Science
Marudupandiyar College,
Thanjavur - 613 403.

Reviewer:
Mr. S. Balasubramanian
Assistant Professor in Computer Science
Directorate of Distance Education
Alagappa University
Karaikudi – 03

SYLLABUS - BOOK MAPPING TABLE
COMPUTER NETWORKS

UNIT	SYLLABUS	MAPPING IN BOOK
	BLOCK 1 : INTRODUCTION & PHYSICALLAYER	
1	**Introduction:** Computer Networks - Applications - Line configuration - Topology - Transmission Modes	**Pages 1 - 17**
2	**Categories of Network:** LAN, MAN, WAN - OSI Layer.	**Pages 18 - 27**
3	**Physical Layer:**Analog and Digital Signals Performance - Transmission Media	**Pages 28 - 43**
	BLOCK 2 : DATA LINK LAYER	
4	**Data Link Layer:** Error Detection and Correction – Introduction – Block Coding – Cyclic Redundancy Check – Framing – Flow and Error Control.	**Pages 44 – 54**
5	**Data Link Layer Protocols:**Stop - Wait Protocol and Sliding Window Protocol - ARQ, Go-Back-N ARQ, Selective, and Repeat ARQ.	**Pages 55 – 68**
6	**Multiple Access Protocols:** ALOHA – CSMA – CSMA/CD – CSMA/CA.	**Pages 69 – 82**
	BLOCK 3 : NETWORK LAYER	
7	**Introduction:** Circuit Switching - Packet Switching - Message Switching - Virtual Circuit and Datagram Subnets	**Pages 83 – 96**
8	**Routing Algorithm:**Static Routing -Shortest Path Routing, Flooding, Flow Based Routing - Dynamic Routing - Distance Vector Routing, Link State Routing	**Pages 97 – 116**
9	**Other Routing Algorithms:** Hierarchical routing, Broad cast, Multicast Routing - Congestion Control Algorithms	**Pages 117-127**
	BLOCK 4 : TRANSPORT LAYER	
10	**Introduction:** Process-to-Process Delivery - UDP - TCP-Connection Oriented VsConnectionless Services.	**Pages 128 – 142**
11	**Applications and Services:** Domain Name System - Remote Logon – Mail Exchange - File Transfer	**Pages 143 – 159**

CONTENTS

BLOCK 2: DATA LINK LAYER

BLOCK3: NETWORK LAYER

BLOCK 4: TRANSPORT LAYER

BLOCK 5: NETWORK SECURITY

BLOCK -1: INTRODUCTION AND PHYSICAL LAYER

UNIT-1INTRODUCTION

STRUCTURE

1.0 INTRODUCTION

The Computers, communication technology and networks are playing vital role in our daily walk of life. This unit describes the information about significance of computer networks, architecture of computer networks, goals of computer networks, network topology, line configuration and different types of transmission modes, applications of computer networks.

1.1 OBJECTIVES

Subsequent to referring this unit, you will able to understand about:

- Concept of computer networks, communication devices.
- Computer network topologies include a bus, star, and ring.etc..
- Networks can be broadly classified as using either a peer-to-peer or client/server architecture.

- Computers on a network are sometimes called nodes. Computers and devices that allocate resources for a network are called servers.

1.2 COMPUTER NETWORKS

An interconnected collection of autonomous computers are called computer networks. Two computers are said to be interconnected if they are able to exchange information. The connections need not be via a copper wire; fiber optics, microwaves, and communication satellites can also be used. If one computer can forcibly start, stop, or control another one, the computers are not autonomous. A system with one control unit and many slaves is not a network; nor is a large computer with remote printers and terminals.

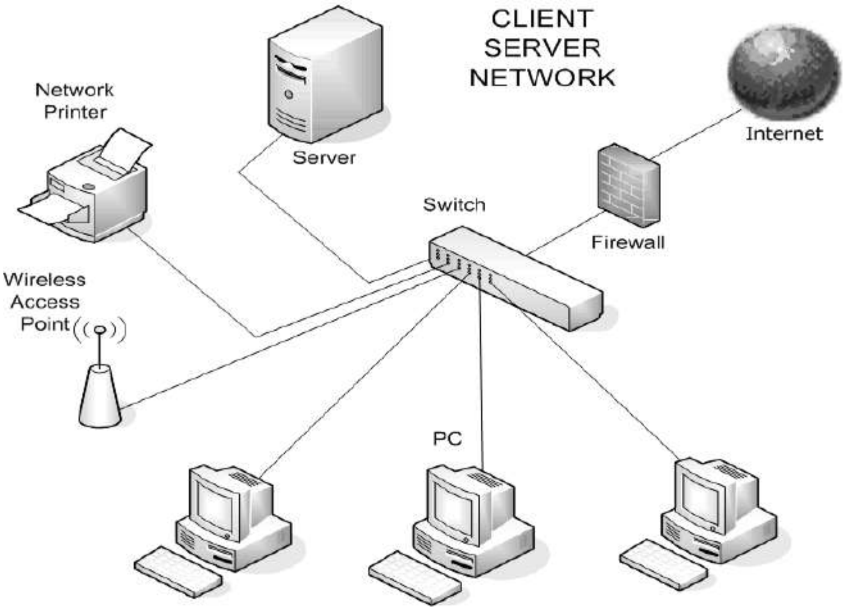


Figure 1.1: Architecture of Computer Networks

1.2.1 Goals of Computer Networks:

(i) Resource Sharing:

Put in slightly more general form, the issue here is resource sharing, and the goal is to make all programs, equipment, and especially data available to anyone on the network without regard to the physical location of the resource and the user. In other words, the mere fact that a user happens to be 1000 km away from his data should not prevent him from using the data as though they were local. This goal may be summarized by saying that it is an attempt to end the "tyranny of geography."

(ii) High Reliability:

A second goal is to provide high reliability by having alternative sources of supply. For example, all files could be replicated on two or three machines, so if one of them is unavailable (due to a hardware failure), the other copies could be used. In addition, the presence of multiple CPU s means that if one goes down, the others may be able to take over its work, although at reduced performance. For military, banking, air traffic control, nuclear reactor safety, and many other applications, the ability to continue operating in the face of hardware problems is of utmost importance.

(iii) Saving Money:

Another goal is saving money. Small computers have a much better price/performance ratio than large ones. Mainframes (room-size computers) are roughly a factor of ten faster than personal computers, but they cost a thousand times more. This imbalance has caused many systems designers to build systems consisting of personal computers, one per user, with data kept on one or more shared file server machines. In this model, the users are called clients, and the whole arrangement is called the client-server model which is presented in figure 1.1.

(iv) Communication Medium

A computer network can provide a powerful communication medium among widely separated employees. Using a network it is easy for two or more people who live far apart to write a report together. When one worker makes a change to an on-line document, the others can see the change immediately; instead of waiting for several days for a letter. Such a speedup makes cooperation among far-flung groups of people easy where it previously had been impossible. In the long run, the use of networks to enhance human-to-human communication will probably prove more important than technical goals such as improved reliability.

1.2.2 Components of Computer Networks

The computer networks are consists of several communication components. The Components of Computer Networks are exposed in figure 1.2.

NOTES

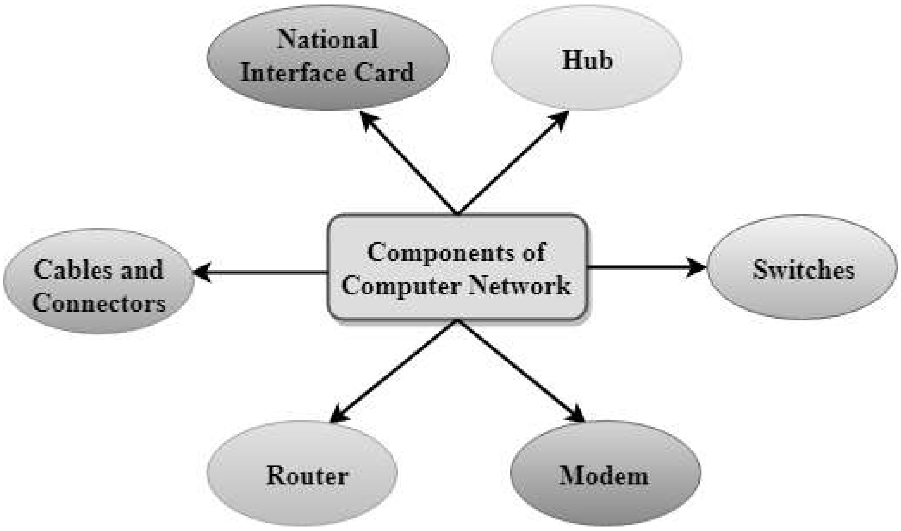


Figure 1.2: Components of Computer Networks

a). National Interface Card (NIC)

The National interface card (NIC) is a device that helps the computer to communicate with another device. The network interface card contains the hardware addresses, the data-link layer protocol use this address to identify the system on the network so that it transfers the data to the correct destination.

There are two types of National interface card (NIC) available.

i). **Wireless NIC:** All the modern laptops use the wireless NIC. In Wireless NIC, a connection is made using the antenna that employs the **radio wave technology**.

ii). **Wired NIC:** Cables use the **wired NIC** to transfer the data over the medium.

b). Hub

Hub is a central device that splits the network connection into multiple devices. When computer requests for information from a computer, it sends the request to the Hub. Hub distributes this request to all the interconnected computers.

c). Switches

Switch is a networking device that groups all the devices over the network to transfer the data to another device. A switch is better than Hub as it does not broadcast the message over the network, i.e., it sends the message to the device for which it belongs to. Therefore, we can say that switch sends the message directly from source to the destination.

d). Cables and Connectors

Cable is a transmission media that transmits the communication signals. **There are three types of cables:**

i). **Twisted pair cable:** It is a high-speed cable that transmits the data over **1Gbps** or more.

ii). **Coaxial cable:** Coaxial cable resembles like a TV installation cable. Coaxial cable is more expensive than twisted pair cable, but it provides the high data transmission speed.

iii). **Fibre optic cable:** Fibre optic cable is a high-speed cable that transmits the data using light beams. It provides high data transmission speed as compared to other cables. It is more expensive as compared to other cables, so it is installed at the government level.

e). Router

Router is a device that connects the LAN to the internet. The router is mainly used to connect the distinct networks or connect the internet to multiple computers.

f). Modem

The modem is a combination of Modulator and Demodulator for converting digital to analog signals and vice-versa. Modem connects the computer to the internet over the existing telephone line. A modem is not integrated with the computer motherboard. A modem is a separate part on the PC slot found on the motherboard.

1.3. APPLICATIONS OF COMPUTER NETWORKS

The Computer networks are occupied significant role in various fields such as:

a). **Cable TV**: This is widest usable thing today throughout the world.

b). **Cellular Telephone**: Wireless phone communication even while travelling through long distance.

c). **Directory Services**: It allows list of files to be stored in central location to speed up the world wide search operation. E.g. search engines like Google, Britannia, and Yahoo etc.

d). **Electronic Data Exchange (EDI)**: EDI allows business information (including documents such as purchase orders and services) to be transferred without using paper.

e). **Electronic Messaging**: E-mails transfer the messages between two and more users in a network. With this application user can transfer the information in the form of text, picture and voice.

f). **Financial services**: It include credit history searches, foreign exchange and investment services and electronic fund transfer (EFT), which allow a user to transfer money without going to bank.

g). **Information Services**: It includes Bulletin Boards and data bank. A '**www**' site offering the technical specification for a new product in an information services.

h). **Manufacturing**: computer networks are used today in many aspects of manufacturing, including the manufacturing processes itself. Two aspects that uses network to provide essential services are computer Assisted design (CAD) and computer Assisted manufacturing (CAM), both of which allow multiple user to work on a project simultaneously.

i). **Marketing and sales**: Marketing professional uses them to collect exchange and analyze data relating to customer needs and product development cycles. The sales application includes Teleshopping, which uses order entry computers or telephone connected to an order processing network, and online reservation services for railways, hotels, airlines, restaurants theatre etc.

j). **Teleconferencing**: It allows conference to occur without the participant being in the same location. It includes:

   i). **Text Conferencing**: Participant communicates through their keywords and computer monitors.

   ii). **Voice Conferencing**: Participant at a number of locations communicates simultaneously through phone (talk).

   iii). **Video Conferencing**: Participant can see as well as talk to another.

1.4 LINE CONFIGURATION

The sender and receiver are synchronized at bit level. Interface: The physical layer defines the transmission interface between devices and transmission medium. Line Configuration: A Network is nothing but a connection made through connection links between two or more devices such as computers, switches, routers, or servers printer or any other device that is capable to send and receive data. There are two ways to connect the devices:

a). Point-to-Point connection

b). Multipoint connection

a). Point-To-Point Connection

It is a protocol which is used as a communication link between two devices. It is simple to establish. The most common example for Point-to-Point connection (PPP) is a computer connected by telephone line. We can connect the two devices by means of a pair of wires or using a microwave or satellite link. For example, the Point-to-Point connection between remote control and Television for changing the channels.

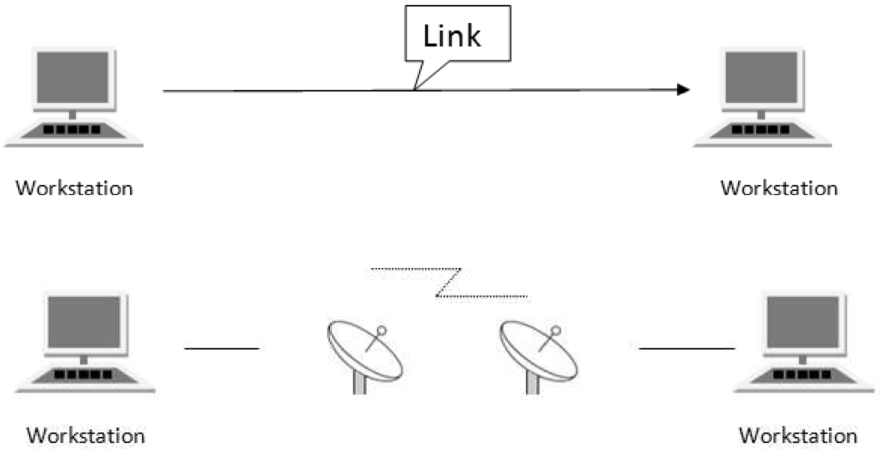


Figure 1.3.Point-to-Point connection

b). Multipoint Connection

Multipoint Connection is capable of connecting two or more devices and share a single link. There are two kinds of Multipoint Connections:

i). If the links are used simultaneously between many devices, then it is spatially shared line configuration.

ii). If user takes turns while using the link, then it is time shared (temporal) line configuration.

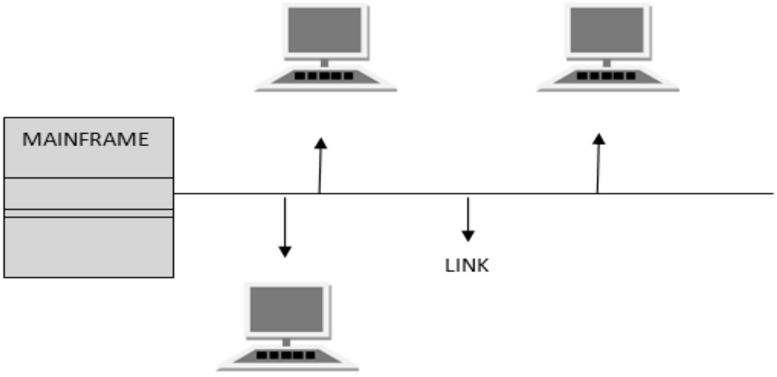


Figure 1.4.Multipoint Connection

1.5 TOPOLOGY

The physical layout of computer network is called network topology. The network topology is the pattern in which nodes (i.e., computers, printers, routers or other devices) are connected to a local area network (LAN) or other network via links (e.g., twisted pair copper wire cable or optical fiber cable). Topologies may define both physical and logical aspect of the network. Both logical and physical topologies could be same or different in a same network. A network topology is the arrangement of nodes -- usually switches, routers, or software switch or router features and connections in a network, often represented as a graph. The topology of the network and the relative locations of the source and destination of traffic flows on the network, determine the optimum path for each flow and the extent to which redundant options for routing exist in the event of a failure. There are two ways of defining network geometry: the physical topology and the logical (or signal) topology. A wide variety of physical topologies have been used in LANs, including ring, bus, mesh and star. Conversely, mapping the data flow between the components determines the logical topology of the network.

a). Bus Topology

In case of Bus topology, all devices share single communication line or cable. Bus topology may have problem while multiple hosts sending data at the same time. Therefore, Bus topology either uses CSMA/CD technology or recognizes one host as Bus Master to solve the issue. It is one of the simple forms of networking where a failure of a device does not affect the other devices. But failure of the shared communication line can make all other devices stop functioning. Both ends of the shared channel have line terminator. The data is sent in only one direction and as soon as it reaches the extreme end, the terminator removes the data from the line.

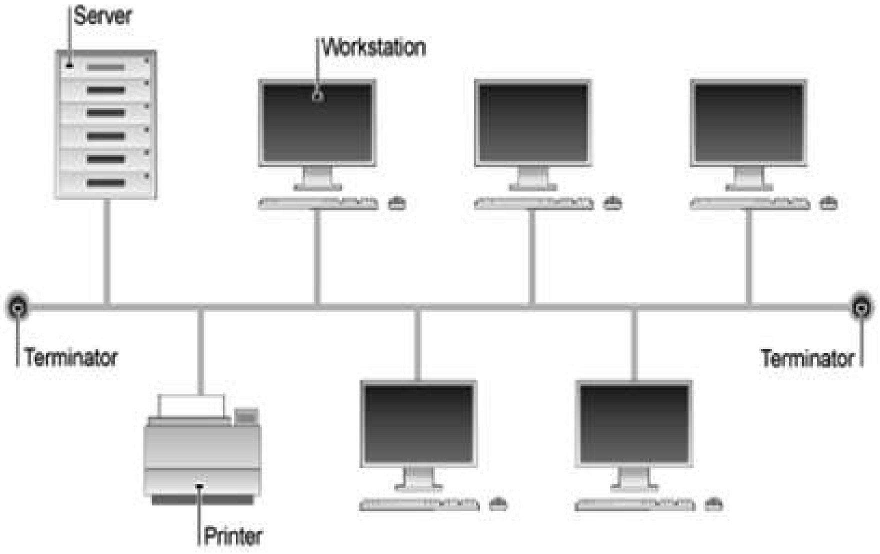


Figure 1.5. Bus Topology

b). Star Topology

All hosts in Star topology are connected to a central device, known as hub device, using a point-to-point connection. That is, there exists a point to point connection between hosts and hub. The hub device can be any of the following:

- Layer-1 device such as hub or repeater
- Layer-2 device such as switch or bridge
- Layer-3 device such as router or gateway

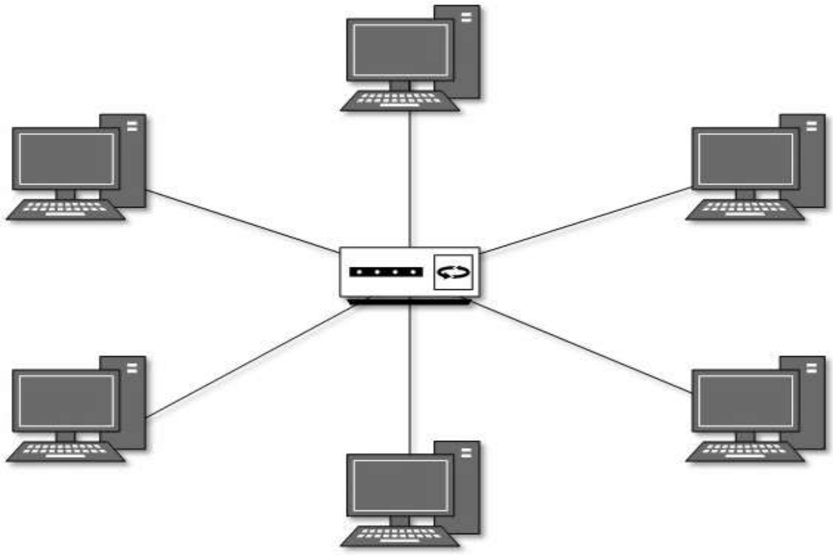


Figure 1.6. Star Topology

As in Bus topology, hub acts as single point of failure. If hub fails, connectivity of all hosts to all other hosts fails. Every communication between hosts takes place through only the hub. Star topology is not expensive as to connect one more host, only one cable is required and configuration is simple.

c). Ring Topology

In ring topology, each host machine connects to exactly two other machines, creating a circular network structure. When one host tries to communicate or send message to a host which is not adjacent to it, the data travels through all intermediate hosts. To connect one more host in the existing structure, the administrator may need only one more extra cable. Failure of any host results in failure of the whole ring. Thus, every connection in the ring is a point of failure. There are methods which employ one more backup ring.

NOTES

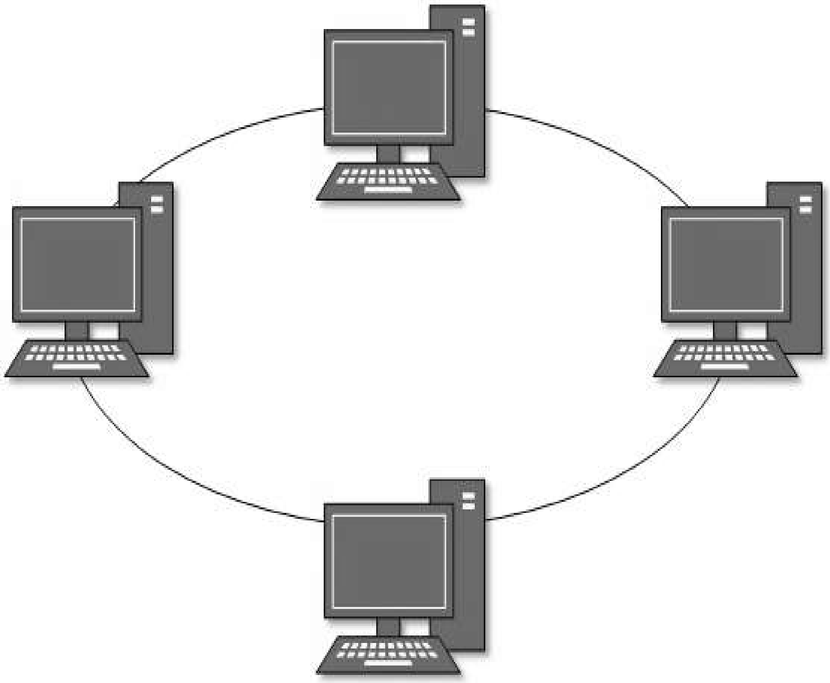


Figure 1.7. Ring Topology

d). Mesh Topology

In mesh topology, a host is connected to one or multiple hosts. This topology has hosts in point-to-point connection with every other host or may also have hosts which are in point-to-point connection with few hosts only. Hosts in Mesh topology also work as relay for other hosts which do not have direct point-to-point links. Mesh technology comes into two types:

- **Full Mesh:** All hosts have a point-to-point connection to every other host in the network. Thus for every new host n(n-1)/2 connections are required. It provides the most reliable network structure among all network topologies.

- **Partially Mesh:** Not all hosts have point-to-point connection to every other host. Hosts connect to each other in some arbitrarily fashion. This topology exists where we need to provide reliability to some hosts out of all.

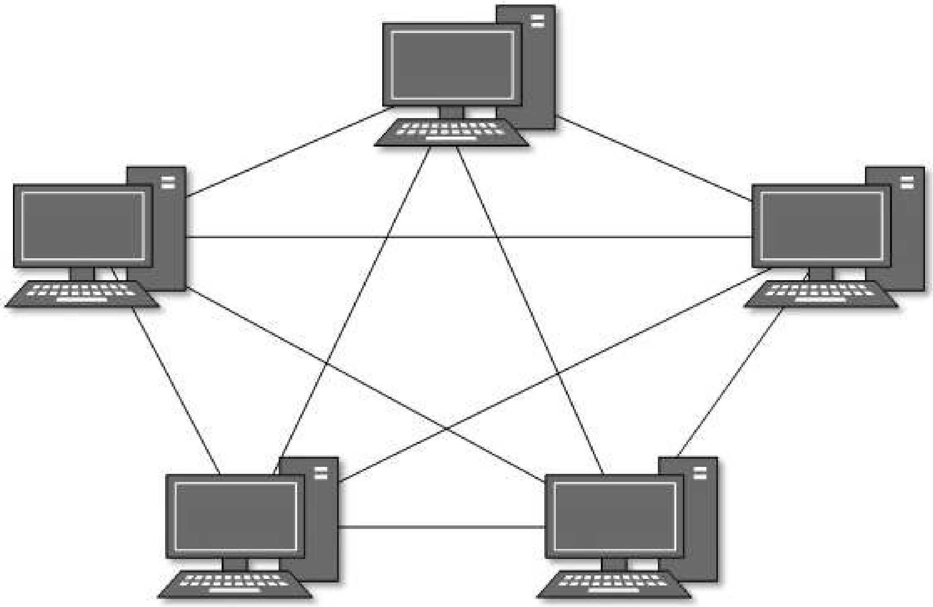


Figure 1.8. Mesh Topology

e). Tree Topology

The Tree Topology is also called as Hierarchical Topology and this is the most common form of network topology in use presently. This topology imitates as extended Star topology and inherits properties of Bus topology.

This topology divides the network into multiple levels or layers of network. Mainly in LANs, a network is bifurcated into three types of network devices. The lowermost is access-layer where computers are attached. The middle layer is known as distribution layer, which works as mediator between upper layer and lower layer. The highest layer is known as core layer, and is central point of the network, i.e. root of the tree from which all nodes fork.

All neighbouring hosts have point-to-point connection between them. Similar to the Bus topology, if the root goes down, then the entire network suffers even though it is not the single point of failure. Every connection serves as point of failure, failing of which divides the network into unreachable segment.

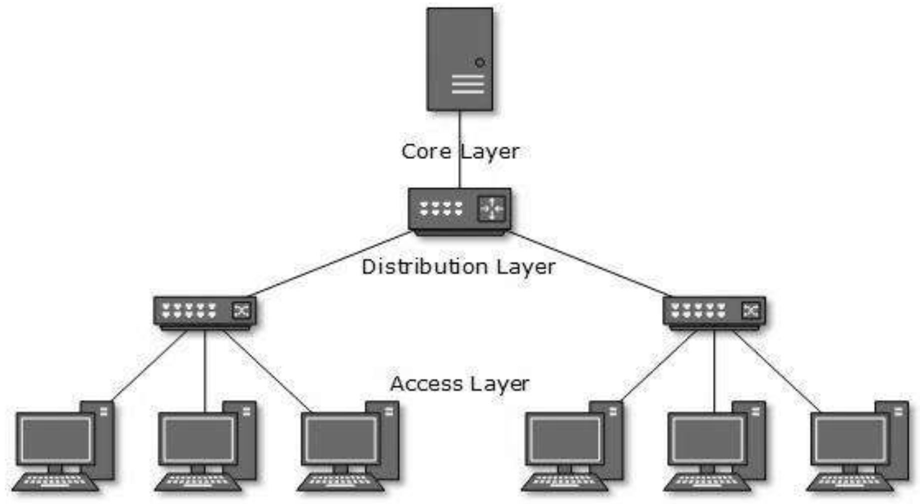


Figure 1.9. Tree Topology

f). Daisy Chain Topology

This topology connects all the hosts in a linear fashion. Similar to Ring topology, all hosts are connected to two hosts only, except the end hosts. Means, if the end hosts in daisy chain are connected then it represents Ring topology. Each link in daisy chain topology represents single point of failure. Every link failure splits the network into two segments. Every intermediate host works as relay for its immediate hosts.

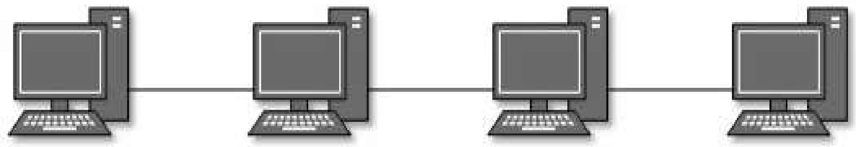


Figure 1.10. Daisy Chain Topology

g). Hybrid Topology

A network structure whose design contains more than one topology is said to be hybrid topology. Hybrid topology inherits merits and demerits of all the incorporating topologies. Hybrid topology is an interconnection of two or more basic network topologies, each of which contains its own nodes. The resulting interconnection allows the nodes in a given basic topology to communicate with other nodes in the same basic topology as well as those in other basic topologies within the hybrid topology.

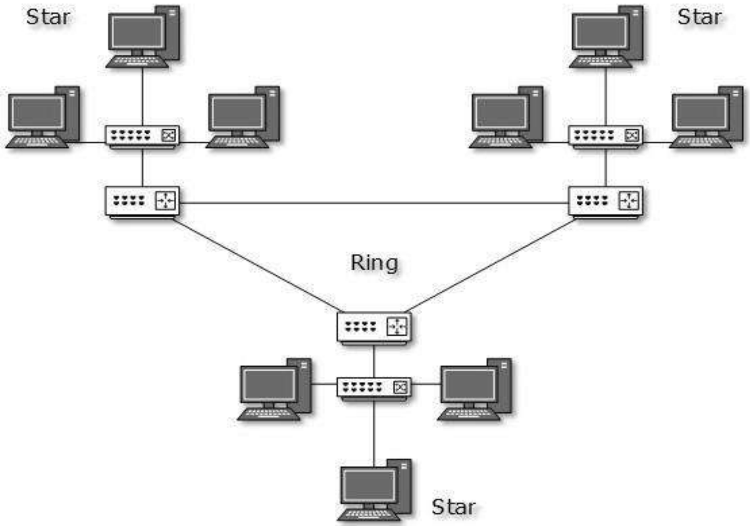


Figure 1.11. Hybrid Topology

The above picture represents an arbitrarily hybrid topology. The combining topologies may contain attributes of star, ring, bus, and daisy-chain topologies. Most WANs are connected by means of Dual-Ring topology and networks connected to them are mostly Star topology networks. Internet is the best example of largest hybrid topology.

1.6. TRANSMISSION MODES

The Transmission Mode means the direction of the flow of information between two communication devices i.e. it tells the direction of signal flow between the two connected devices. There are three ways or modes of data transmission:

i). Simplex,
ii). Half duplex (HDX),
iii). Full duplex (FDX).

a). Simplex

In a simplex transmission mode, the communication between sender and receiver occurs only in one direction. The sender can only send the data and the receiver can only receive the data. The receiver cannot reply to the sender. Simplex is like a one-way road in which the traffic travels only in one direction, no vehicle from the opposite direction is allowed to enter.

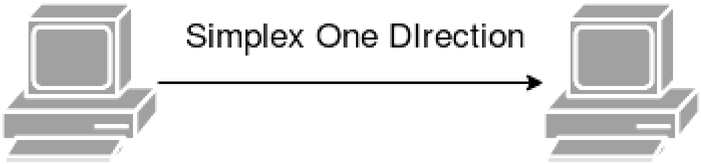


Figure 1.12. Simplex Mode of Data Transmission

NOTES

- **Examples:** Keyboards and monitors or displays, traditional FM radios etc. Traditional FM radios only output broadcasted or transmitted audio from stations, but does not communicate back.

- **Advantage**: The sender can use the full capacity of the medium to transmit data, so more data is transmitted at a time.

- **Disadvantage**: One way connection, so no inter-communication between devices.

b). Half Duplex

The communication between sender and receiver occurs in both the directions in a half duplex transmission but, one at a time. The sender and receiver both can send and receive the information but, only one is allowed to send at a time. Half duplex is still considered a one-way road, in which a vehicle traveling in the opposite direction of the traffic has to wait till the road is empty.

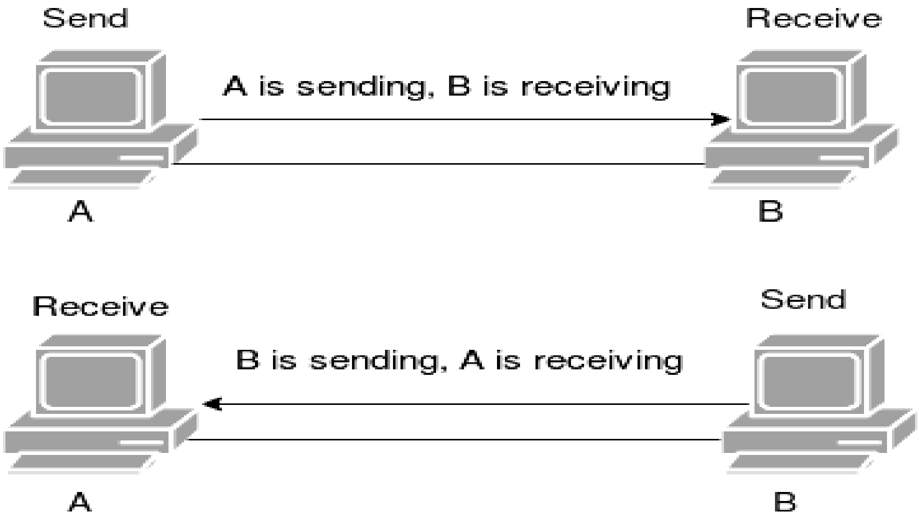


Figure 1.13. Half Duplex Mode of Data Transmission

- **Examples:** For example, in walkie-talkies, CB radios etc. the speaker at both ends can speak but they have to speak one by one. Both cannot speak simultaneously.

- **Advantage**: In half-duplex, both devices can send and receive data and also entire capacity of the transmission medium is used when transmitting data.

- **Disadvantage:** As when one device is sending data then another one must wait, this can cause delay in sending the data at the right time.

c). Full Duplex

In a full duplex transmission mode, the communication between sender and receiver can occur simultaneously. The sender and receiver can both transmit and receive at the same time. The full duplex transmission mode is like a two-way road in which traffic can flow in both directions at the same time. For example, in a telephone, two people communicate, and both are free to speak and listen at the same time.

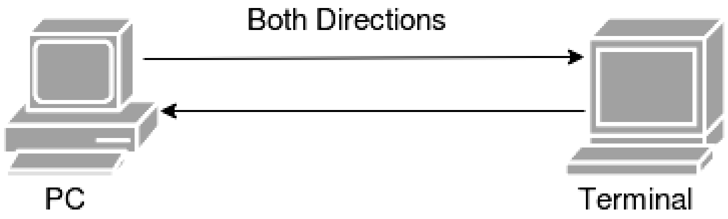


Figure: full-duplex mode

Figure 1.14. Full Duplex Mode of Data Transmission

- **Examples:** telephone, cell phone networks etc.
- **Advantages:** Both parties can talk and listen to each other at the same time.
- **Disadvantages:** If there is no dedicated path in the network then the capacity will be divided into two.

d). Comparison of Simplex, Half Duplex and Full Duplex Mode of Data Transmission

The Comparison between three modes of transmission is that in a simplex mode of transmission the communication is unidirectional, or one-way; whereas in the half duplex mode of transmission the communication is two-directional, but the channel is interchangeably used by both of the connected devices. On the other hand, in the full duplex mode of transmission, the communication is bi-directional or two-way, and the channel is used by both of the connected devices simultaneously. The full duplex transmission mode offers the best performance among the three as it maximizes the bandwidth.

The comparison between the simplex, half duplex and full duplex mode of data transmission is presented in table 1.1.

NOTES

Table 1.1.Comparison Chart of Simplex, Half Duplex and Full Duplex Mode of Data Transmission

Basis for Comparison	Simplex	Half Duplex	Full Duplex
Direction of Communication	Unidirectional	Two-directional, one at a time	Two-directional, simultaneously
Send / Receive	Sender can only send data.	Sender can send and receive data, but one a time.	Sender can send and receive data simultaneously.
Performance	Least performing mode of transmission.	Better than Simplex	Most performing mode of transmission.
Example	Keyboard and monitor	Walkie-talkie	Telephone

1.7 CHECK YOUR PROGRESS QUESTIONS

1. Define Point-To-Point Connection.
2. What is Tree Topology?

1.8 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. The protocol which is used as a communication link between two devices. It is simple to establish. The most common example for Point-to-Point connection (PPP) is a computer connected by telephone line.

2. Tree Topology is also known as Hierarchical Topology, this is the most common form of network topology in use presently. This

topology imitates as extended Star topology and inherits properties of Bus topology.

1.9 SUMMARY

A computer network is a digital telecommunications network which allows nodes to share resources. In computer networks, computing devices exchange data with each other using connections (data links) between nodes. These data links are established over cable media such as wires or optic cables, or wireless media such as Wi-Fi. The computer networks are interconnected to create internetworks. Network topology is a physical layout of computer networks. Three types of transmission modes are available to transmit data between source and destination computers in dissimilar manner.

1.10 KEY WORDS

- Network Topology: The physical layout of computer networks is known as Network Topology.
- High Reliability is defined as having alternative sources of supply.
- Communication Medium: A computer network can provide a powerful communication medium among widely separated people. Using a network it is easy for two or more people who live far apart to write a report together.
- The Transmission Mode means the direction of the flow of information between two communication devices i.e. it tells the direction of signal flow between the two connected devices.

1.11 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Define resource sharing.
2. What is meant by bus topology?.
3. Differentiate half duplex and full duplex mode of communication.

Long Answer Questions

1. Explain about applications of computer networks.
2. Discuss about components of computer networks.

1.12 FURTHER READINGS

1. Computer Networks, 3rd Edition, Andrew S Tanenbaum, Pearson Education, 2010.
2. Data and Computer Communications, William Stallings, 8th Edition, Prentice Hall, 2007.
3. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.

UNIT-2 CATEGORIES OF NETWORK

STRUCTURE

2.0 INTRODUCTION

In this unit, we can have the information about categories of computer networks such as PAN, LAN, MAN, WAN and Internetwork based on geographical area. In addition to that the architecture of ISO OSI network model and its layers are analyzed. A network can be as small as distance between your mobile phone and its Bluetooth headphone and as large as the internet itself, covering the whole geographical world.

2.1 OBJECTIVES

After going through this unit, we will get idea about various network architecture systems. Understand the concept of seven layers of ISO model. The public uses many small communication devices within a single room to millions of devices spread across the entire globe, and can be defined based on purpose and/or size through accessing the internet or printing a document to downloading an attachment from an email, networks are the backbone of every activity today.

2.2 PERSONAL AREA NETWORK (PAN)

The Personal Area Network (PAN) is smallest network which is very personal to a user. This may include Bluetooth enabled devices or infra-red enabled devices. PAN has connectivity range up to 10 meters. PAN may include wireless computer keyboard and mouse, Bluetooth enabled headphones, wireless printers, and TV remotes. For example, Piconet is

Bluetooth-enabled Personal Area Network which may contain up to 8 devices connected together in a master-slave fashion.

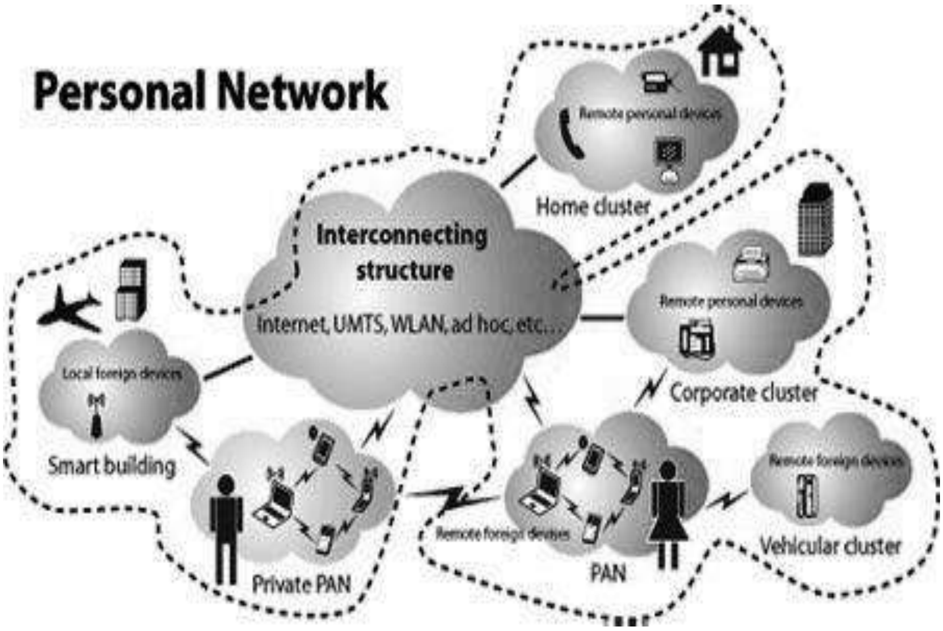


Figure 2.1.Personal Area Network

2.3 LOCAL AREA NETWORK (LAN)

A computer network spanned inside a building and operated under single administrative system is generally termed as Local Area Network (LAN). Usually, LAN covers an organization offices, schools, colleges or universities. Number of systems connected in LAN may vary from as least as two to as much as 16 million. LAN provides a useful way of sharing the resources between end users. The resources such as printers, file servers, scanners, and internet are easily sharable among computers.

The LANs are composed of inexpensive networking and routing equipment. It may contains local servers serving file storage and other locally shared applications. It mostly operates on private IP addresses and does not involve heavy routing. LAN works under its own local domain and controlled centrally. LAN uses either Ethernet or Token-ring technology. Ethernet is most widely employed LAN technology and uses Star topology, while Token-ring is rarely seen. LAN can be wired, ireless, or in both forms at once.

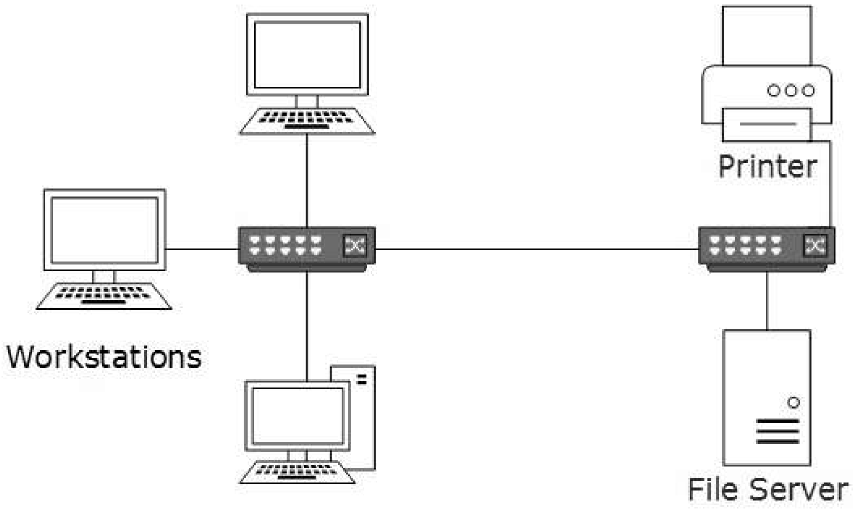


Figure 2.2. Local Area Network

2.4 METROPOLITAN AREA NETWORK (MAN)

The Metropolitan Area Network (MAN) generally expands throughout a city such as cable TV network. It can be in the form of Ethernet, Token-ring, ATM, or Fiber Distributed Data Interface (FDDI). Metro Ethernet is a service which is provided by ISPs. This service enables its users to expand their Local Area Networks. For example, MAN can help an organization to connect all of its offices in a city. Backbone of MAN is high-capacity and high-speed fiber optics. MAN works in between Local Area Network and Wide Area Network. MAN provides uplink for LANs to WANs or internet.

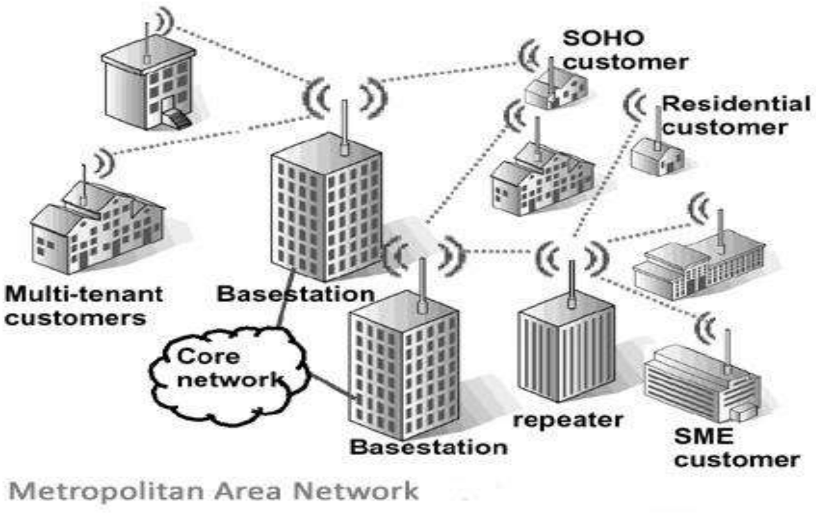


Figure 2.3. Metropolitan Area Network

2.5 WIDE AREA NETWORK (WAN)

As the name suggests, the Wide Area Network (WAN) covers a wide area which may span across provinces and even a whole country. Generally, telecommunication networks are Wide Area Network. These networks provide connectivity to MANs and LANs. Since they are equipped with very high speed backbone, WANs use very expensive network equipment. WAN may use advanced technologies such as Asynchronous Transfer Mode (ATM), Frame Relay, and Synchronous Optical Network (SONET). WAN may be managed by multiple administrations.

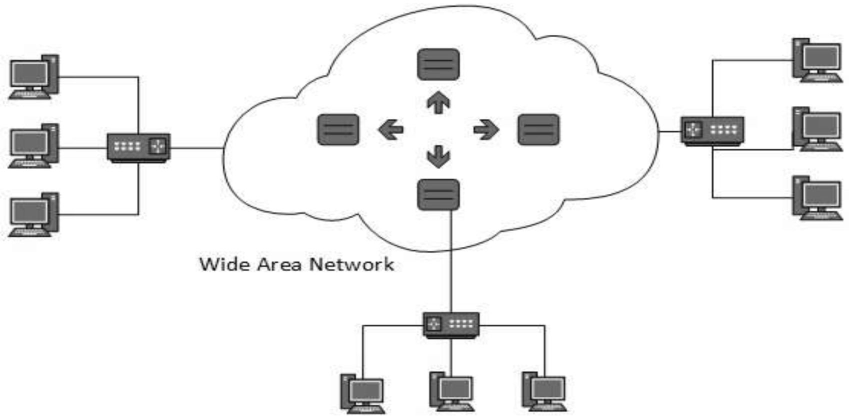


Figure 2.4. Wide Area Network

2.6 INTERNETWORK

A network of networks is called an internetwork, or simply the internet. It is the largest network in existence on this planet. The internet hugely connects all WANs and it can have connection to LANs and Home networks. Internet uses TCP/IP protocol suite and uses IP as its addressing protocol. Present day, Internet is widely implemented using IPv4. Because of shortage of address spaces, it is gradually migrating from IPv4 to IPv6.

Internet enables its users to share and access enormous amount of information worldwide. It uses WWW, FTP, email services, audio, and video streaming etc. At huge level, internet works on Client-Server model. Internet uses very high speed backbone of fiber optics. To inter-connect various continents, fibers are laid under sea known to us as submarine communication cable. Internet is widely deployed on World Wide Web services using HTML linked pages and is accessible by client software known as Web Browsers. When a user requests a page using some web browser located on some Web Server anywhere in the world, the Web Server responds with the proper HTML page.

Table 2.1. Comparison of LAN, MAN, WAN

S No.	LAN	MAN	WAN
1.	Stands for Local Area Network.	Stands for Metropolitan Area Network.	Stands for Wide Area Network.
2.	It is a group of computer and other network devices which are connected together.	It is a larger network of computers and other network devices which are connected together and usually spans over several buildings or large geographical area.	It is a group of computers and other network devices which are connected together and which is not restricted to geographical location.
3.	All the devices that are part of LANs are situated within multiple buildings or offices	All the devices that are part of MAN span across buildings or small town.	All the devices which are part of WAN and have no geographical boundaries.
4.	It has very high speed.	It has lower speed compared to LAN.	The speed is based on geographical location of the servers. It connects several LANs.
5.	Connection speed can be 1Mbps or 100Mbps or 1000Mbps.	Connection speed can be 10Mbps of 100Mbps.	Connection speed can be 10Mbps or 100Mbps.
6.	It uses guided media.	It either uses guided media or unguided media.	It uses guided media or unguided media. Its long distance communications, which may or may not be provided by public packet network.
7.	It has smaller coverage range within the house or office premises.	It has distance coverage and data rate higher than LAN, but less than WAN.	It has larger coverage than LAN and MAN.
8.	Example: Used by desktop and laptops for sharing the common resources such as printer, hard disk etc.	Example: Local Cable TV system.	Example: Telephone system, Internet

2.7 OSI LAYERS

The International Standards Organization (ISO) developed the Open Systems Interconnection (OSI) model. The OSI model describes a networking framework to implement protocols in seven layers. This OSI model is consists of seven layers for making communication. In this model, layers 1-4 are considered the lower layers, and mostly concern themselves with moving data around. Layers 5-7, called the upper layers, contain application-level data. Networks operate on one basic principle: "pass it on." Each layer is organized and defined with specific function or very specific job, and then passes the data onto the next layer.

2.7.1 The Seven Layers of the OSI Model

In the OSI model, control is passed from one layer to the next, starting at the application layer (Layer 7) in one station, and proceeding to the bottom layer, over the channel to the next station and back up the hierarchy. The OSI model takes the task of inter-networking and divides that up into what is referred to as a vertical stack that consists of the following 7 layers.

i). Layer 1- Physical Layer
ii). Layer 2 - Data Link Layer
iii). Layer 3 - Network Layer
iv). Layer 4 - Transport Layer
v). Layer 5 - Session Layer
vi). Layer 6 - Presentation Layer
vii). Layer 7 – Application Layer

i). Physical (Layer 1)

OSI Model, Layer 1 conveys the bit stream - electrical impulse, light or radio signal — through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects. Fast Ethernet, RS232, and ATM are protocols with physical layer components.

Example: Layer 1 Physical examples include Ethernet, FDDI, B8ZS, V.35, V.24, RJ45.

ii). Data Link (Layer 2)

At OSI Model, Layer 2, data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sub layer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.

Example: Layer 2 Data Link examples include PPP, FDDI, ATM, IEEE 802.5/ 802.2, IEEE 802.3/802.2, HDLC, Frame Relay.

iii). Network (Layer 3)

Layer 3 provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing.

Example: Layer 3 Network examples include AppleTalk DDP, IP, IPX.

iv). Transport (Layer 4)

OSI Model, Layer 4, provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer.

Example: Layer 4 Transport examples include SPX, TCP, UDP.

v). Session (Layer 5)

This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.

Example: Layer 5 Session examples include NFS, NetBios names, RPC, SQL.

vi). Presentation (Layer 6)

This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.

Example: Layer 6 Presentation examples include encryption, ASCII, EBCDIC, TIFF, GIF, PICT, JPEG, MPEG, MIDI.

vii). Application (Layer 7)

OSI Model, Layer 7, supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services. Telnet and FTP are applications that exist entirely in the application level. Tiered application architectures are part of this layer.

Example: Layer 7 Application examples include WWW browsers, NFS, SNMP, Telnet, HTTP, FTP.

The OSI reference model is conceptual framework for understanding relationships. The purpose of the OSI reference model is to guide vendors and developers so the digital communication products and software programs they create can interoperate, and to facilitate a clear framework that describes the functions of a networking or telecommunication system. Most vendors involved in telecommunications make an attempt to describe their products and services in relation to the OSI model. And although it is useful for guiding discussion and evaluation, OSI is rarely actually implemented as-is. That's because few network products or standard tools keep related functions together in well-defined layers, as is the case in the OSI model. The TCP/IP protocol suite, which defines the internet, does not map cleanly to the OSI model. The architecture of OSI model is shown in figure 2.5.

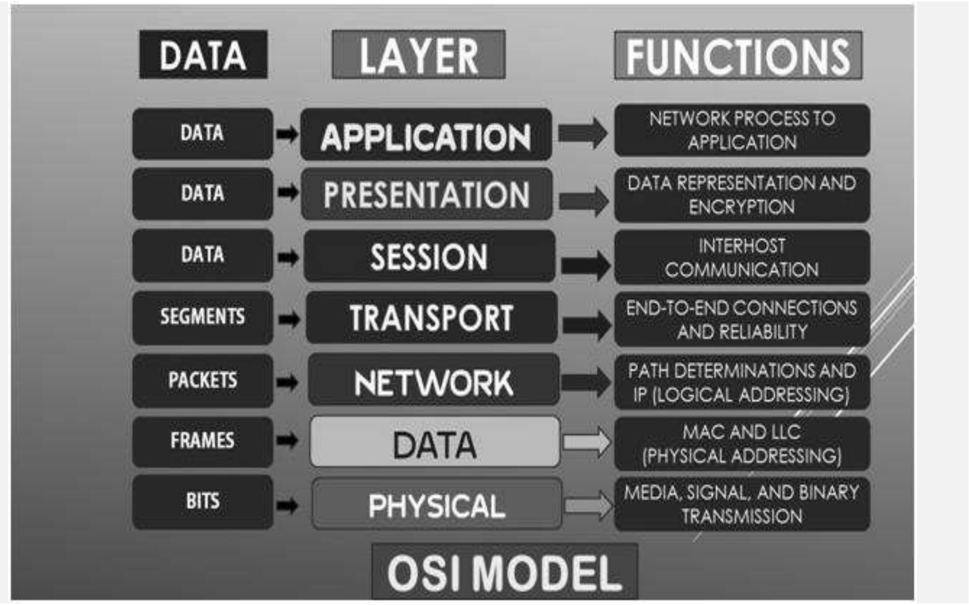

DATA
LAYER
FUNCTIONS
DATA
APPLICATION
NETWORK PROCESS TO APPLICATION
DATA
PRESENTATION
DATA REPRESENTATION AND ENCRYPTION
DATA
SESSION
INTERHOST COMMUNICATION
SEGMENTS
TRANSPORT
END-TO-END CONNECTIONS AND RELIABILITY
PACKETS
NETWORK
PATH DETERMINATIONS AND IP (LOGICAL ADDRESSING)
FRAMES
DATA
MAC AND LLC (PHYSICAL ADDRESSING)
BITS
PHYSICAL
MEDIA, SIGNAL, AND BINARY TRANSMISSION
OSI MODEL

Figure 2.5. OSI Reference Model

2.7.2 Working Method of the OSI Model

The IT professionals use OSI to model or trace how data is sent or received over a network. This model breaks down data transmission over a series of seven layers, each of which is responsible for performing specific tasks concerning sending and receiving data. The main object of OSI is that the process of communication between two endpoints in a network can be divided into seven distinct groups of related functions, or layers. Each communicating user or program is on a device that can provide those seven layers of function.

In this architecture, each layer serves the layer above it and, in turn, is served by the layer below it. So, in a given message between users, there will be a flow of data down through the layers in the source computer, across the network, and then up through the layers in the receiving computer. Only the application layer, at the top of the stack, doesn't provide services to a higher-level layer. The seven layers of function are provided by a combination of applications, operating systems, network card device drivers and networking hardware that enable a system to transmit a signal over a network Ethernet or fiber optic cable or through Wi-Fi or other wireless protocols.

2.8 CHECK YOUR PROGRESS QUESTIONS

1. How to classify the computer networks?
2. State the role of protocols.
3. Define Fiber Optic Network.

2.9 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. The computer network is classified with respect to distance or geographical structure. can be as small as distance between your mobile phone and its Bluetooth headphone and as large as the internet itself, covering the whole geographical world. For example PAN, LAN, MAN, WAN and Internetwork.

2. Network protocol is consists of set of rules or procedures to safeguard the format and meaning of information while transmit through public media like internet.

3. The Fiber optics is utilized for LANs and in addition to for long-haul transmission, although tapping into it is more complex than connecting to an Ethernet. One way around the problem is to realize that a ring network is really just a collection of point-to-point links.

2.10 SUMMARY

The computer networks are having numerous applications in various fields. The computer networks are classified into PAN, LAN, MAN, WAN and Internetworks by means of their own characteristics, technologies, transmission speeds and positions. Network protocols are used to send and receive the data or information in a secured manner. The International Standards Organization (ISO) developed the Open Systems Interconnection (OSI) model. It divides network communication into seven layers. In this model, layers 1-4 are considered the lower layers, and mostly concern themselves with moving data around. Layers 5-7, called the upper layers, contain application-level data.

2.11 KEY WORDS

- **Personal Area Network (PAN):**A Personal Area Network (PAN) is smallest network which is very personal to a user. This may include Bluetooth enabled devices or infra-red enabled devices. PAN has connectivity range up to 10 meters.
- **Local Area Network (LAN) :**A computer network spanned inside a building and operated under single administrative system is generally termed as Local Area Network (LAN). Usually, LAN covers an organization offices, schools, colleges or universities. Number of systems connected in LAN may vary from at least as two to as much as 16 million.
- **Metropolitan Area Network (MAN):** The Metropolitan Area Network (MAN) generally expands throughout a city such as cable TV network. It can be in the form of Ethernet, Token-ring, ATM, or Fiber Distributed Data Interface (FDDI). Metro Ethernet is a service which is provided by ISPs.
- **Wide Area Network (WAN):** The Wide Area Network (WAN) covers a wide area which may span across provinces and even a whole country. Generally, telecommunication networks are Wide Area Network. These networks provide connectivity to MANs and LANs.

2.12 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Define Internetworks.
2. Mention the importance of FTP.
3. What is meant by HTTP?

Long Answer Questions

1. Explain about Computer Network Topology
2. Illustrate about OSI Reference Model.

2.13 FURTHER READINGS

1. Wetteroth, OSI Reference Model for Telecommunications.
2. Palais, Fiber Optic Communication, 3rd ed.
3. Sarikaya, "Packet Mode in Wireless Networks: Overview of Transition to Third Generation".
4. Data and Computer Communications, William Stallings, 8th Edition, Prentice Hall, 2007.

UNIT -3 PHYSICALLAYER

Structure

3.0 INTRODUCTION

In this unit, we will gain knowledge about physical layer of computer networks. The physical layer defines mechanical, electrical and timing interfaces to the network. In addition to that digital and analog signal and its significance are explained.

3.1 OBJECTIVES

After going through this unit, you will understand and express about

- Digital and Analog Transmission
- Asynchronous and Synchronous Transmission
- Guided Transmission Media
- Unguided Transmission Media

3.2 ANALOG AND DIGITAL SIGNALS PERFORMANCE

3.2.1 Analog and Digital Transmission

Data is transmitted from one point to another point by means of electrical signals that may be in digital and analog form. In analog signal the transmission power varies over a continuous range with respect to sound, light and radio waves. On the other hand, a digital signal may assume only discrete set of values within a given range. Analog signal is measured in Volts and its frequency is in Hertz (Hz). A digital signal is a sequence of voltage represented in binary form. When digital data are to be sent over an analog form the digital signal must be converted to analog form. The diagram of analog signal and digital signal are shown in fig. 3.1.

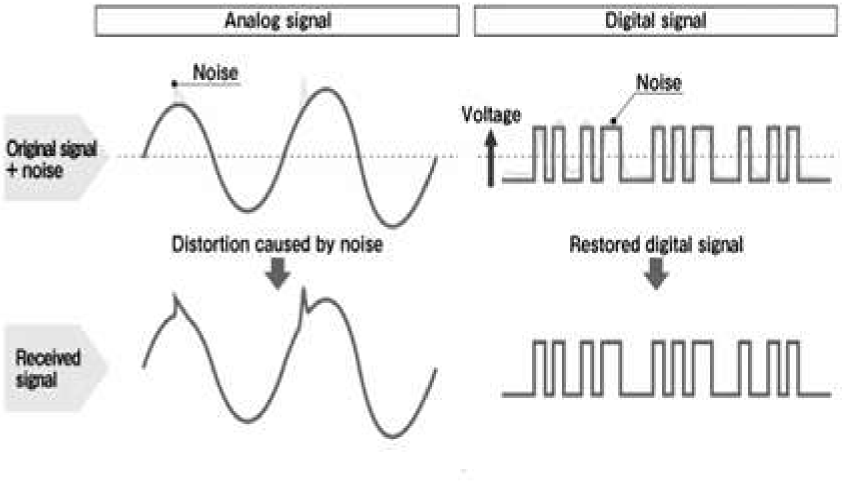


Figure 3.1.Analog and Digital Transmission

3.2.2 Asynchronous and Synchronous Transmission

Asynchronous and synchronous communication refers to methods by which signals are transferred in computing technology. These signals allow computers to transfer data between components within the computer or between the computer and an external network.

Data transmission through a medium can be either asynchronous or synchronous. In asynchronous transmission data is transmitted character by character as you go on typing on a keyboard. The saved data is transmitted block by block. Each block can contain many characters.

Synchronous transmission is well suited for remote communication between a computer and related devices like card reader and printers. The Asynchronous Transmission and Synchronous Transmission systems are shown in figure 3.2.

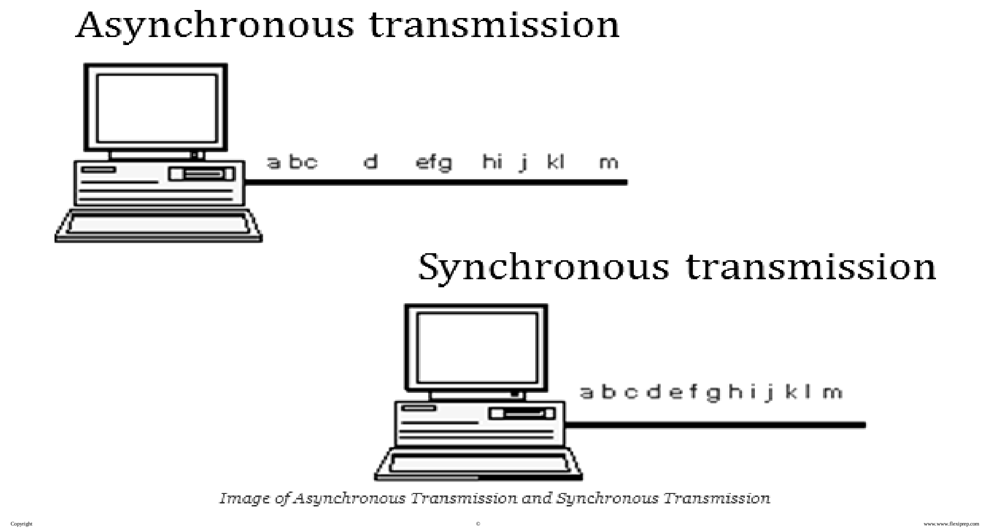

Asynchronous transmission
a bc d efg hi j kl m
Synchronous transmission
a b c d e f g h i j k l m
Image of Asynchronous Transmission and Synchronous Transmission
Copyright
©

Figure 3.2. Asynchronous and Synchronous Transmission

3.3 TRANSMISSION MEDIA

The media, over which the information between two computer systems is sent, called transmission media. Transmission media comes in two forms.

- **Guided Media**

All communication wires or cables are guided media, such as UTP, coaxial cables, and fiber Optics. In this media, the sender and receiver are directly connected and the information is send (guided) through it.

- **Unguided Media**

Wireless or open air space is said to be unguided media, because there is no connectivity between the sender and receiver. Information is spread over the air, and anyone including the actual recipient may collect the information.

3.3.1 Guided Transmission Media

The purpose of the physical layer is to transport a raw bit stream from one machine to another. Various physical media can be used for the actual 1transmission. Each one has its own niche in terms of bandwidth, delay, cost, and ease of installation and maintenance. Media are roughly grouped into guided media, such as copper wire and fiber optics, and unguided media, such as radio and lasers through the air. The various guided transmission media systems are presented in the following sections .

a). Magnetic Media

One of the most common ways to transport data from one computer to another is to write them onto magnetic tape or removable media (e.g., recordable DVDs), physically transport the tape or disks to the destination machine, and read them back in again. Although this method is not as sophisticated as using a geosynchronous communication satellite, it is often more cost effective, especially for applications in which high bandwidth or cost per bit transported is the key factor. A simple calculation will make this point clear. An industry standard Ultrium tape can hold 200 gigabytes. A box 60 x 60 x 60 cm can hold about 1000 of these tapes, for a total capacity of 200 terabytes, or 1600 terabits (1.6 petabits). A box of tapes can be delivered anywhere in the United States in 24 hours by Federal Express and other companies.

The effective bandwidth of this transmission is 1600 terabits/86,400 sec, or 19 Gbps. If the destination is only an hour away by road, the bandwidth is increased to over 400 Gbps. No computer network can even approach this. For a bank with many gigabytes of data to be backed up daily on a second machine (so the bank can continue to function even in the face of a major flood or earthquake), it is likely that no other transmission technology can even begin to approach magnetic tape for performance. Of course, networks are getting faster, but tape densities are increasing, too. If we now look at cost, we get a similar picture. The cost of an Ultrium tape is around $40 when bought in bulk. A tape can be reused at least ten times, so the tape cost is maybe $4000 per box per usage. Add to this another $1000 for shipping (probably much less), and we have a cost of roughly $5000 to ship 200 TB. This amounts to shipping a gigabyte for fewer than 3 cents. No network can beat that. The moral of the story is: Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway

b). Twisted Pair

Although the bandwidth characteristics of magnetic tape are excellent, the delay characteristics are poor. Transmission time is measured in minutes or hours, not milliseconds. For many applications an on-line connection is needed. One of the oldest and still most common transmission media is twisted pair. A twisted pair consists of two insulated copper wires, typically about 1 mm thick. The wires are twisted together in a helical form, just like a DNA molecule. Twisting is done because two parallel wires constitute a fine antenna.

When the wires are twisted, the waves from different twists cancel out, so the wire radiates less effectively. The most common application of the twisted pair is the telephone system. Nearly all telephones are connected to the telephone company (telco) office by a twisted pair. Twisted pairs can run several kilometres without amplification, but for longer distances, repeaters are needed. When many twisted pairs run in parallel for a substantial distance, such as all the wires coming from an apartment building to the telephone company office, they are bundled together and encased in a protective sheath. The pairs in these bundles would interfere with one another if it were not for the twisting. In parts of the world where telephone lines run on poles above ground, it is common to see bundles several centimetres in diameter.

Twisted pairs can be used for transmitting either analog or digital signals. The bandwidth depends on the thickness of the wire and the distance travelled, but several megabits/sec can be achieved for a few kilometres in many cases. Due to their adequate performance and low cost, twisted pairs are widely used and are likely to remain so for years to come. Twisted pair cabling comes in several varieties, two of which are important for computer networks. Category 3 twisted pairs consist of two insulated wires gently twisted together. Four such pairs are typically grouped in a plastic sheath to protect the wires and keep them together. Prior to about 1988, most office buildings had one category 3 cable running from a central wiring closet on each floor into each office. This scheme allowed up to four regular telephones or two multiline telephones in each office to connect to the telephone company equipment in the wiring closet.

Starting around 1988, the more advanced category 5 twisted pairs were introduced. They are similar to category 3 pairs, but with more twists per centimetre, which results in less crosstalk and a better-quality signal over longer distances, making them more suitable for high-speed computer communication. Up-and-coming categories are 6 and 7, which are capable of handling signals with bandwidths of 250 MHz and 600 MHz, respectively (versus a mere 16 MHz and 100 MHz for categories 3 and 5, respectively). All of these wiring types are often referred to as UTP (Unshielded Twisted Pair), to contrast them with the bulky, expensive, shielded twisted pair cables IBM introduced in the early 1980s, but which have not proven popular outside of IBM installations. Twisted pair cabling is illustrated in Fig. 3.3.

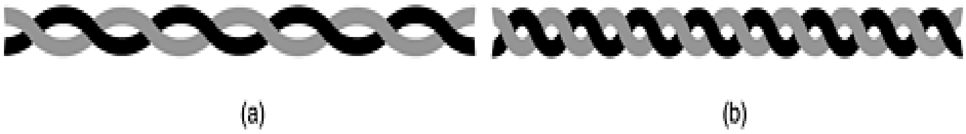


Figure 3.3. (a) Category 3 UTP. (b) Category 5 UTP.

c). Coaxial Cable

Another common transmission medium is the coaxial cable (known to its many friends as just "coax" and pronounced "co-ax"). It has better shielding than twisted pairs, so it can span longer distances at higher speeds. Two kinds of coaxial cable are widely used. One kind, 50-ohm cable, is commonly used when it is intended for digital transmission from the start. The other kind, 75-ohm cable, is commonly used for analog transmission and cable television but is becoming more important with the advent of Internet over cable. This distinction is based on historical, rather than technical, factors (e.g., early dipole antennas had an impedance of 300 ohms, and it was easy to use existing 4:1 impedance matching transformers).

A coaxial cable consists of a stiff copper wire as the core, surrounded by an insulating material. The insulator is encased by a cylindrical conductor, often as a closely-woven braided mesh. The outer conductor is covered in a protective plastic sheath. A cutaway view of a coaxial cable is shown in Fig. 3-4.

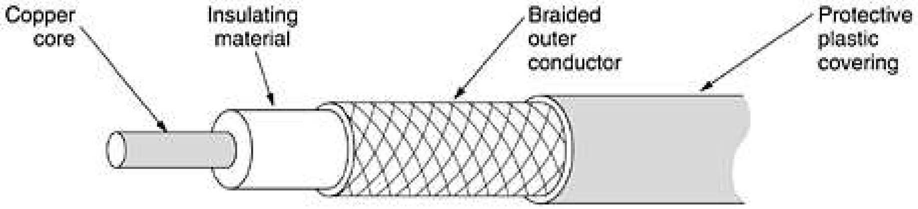


Figure 3.4. Coaxial Cable.

d). Fiber Optics

Many people in the computer industry take enormous pride in how fast computer technology is improving. The original (1981) IBM PC ran at a clock speed of 4.77 MHz. Twenty years later, PCs could run at 2 GHz, a gain of a factor of 20 per decade. Not too bad. In the same period, wide area data communication went from 56 kbps (the ARPANET) to 1 Gbps (modern optical communication), a gain of more than a factor of 125 per decade, while at the same time the error rate went from 10-5 per bit to almost zero.

Furthermore, single CPUs are beginning to approach physical limits, such as speed of light and heat dissipation problems. In contrast, with current fiber technology, the achievable bandwidth is certainly in excess of 50,000 Gbps (50 Tbps) and many people are looking very hard for better technologies and materials. The current practical signalling limit of about 10 Gbps is due to our inability to convert between electrical and optical signals any faster, although in the laboratory, 100 Gbps has been achieved on a single fiber. In the race between computing and communication, communication won.

The full implications of essentially infinite bandwidth (although not at zero cost) have not yet sunk in to a generation of computer scientists and engineers taught to think in terms of the low Nyquist and Shannon limits imposed by copper wire. The new conventional wisdom should be that all computers are hopelessly slow and that networks should try to avoid computation at all costs, no matter how much bandwidth that wastes. In this section we will study fiber optics to see how that transmission technology works.

An optical transmission system has three key components: the light source, the transmission medium, and the detector. Conventionally, a pulse of light indicates a 1 bit and the absence of light indicates a 0 bit. The transmission medium is an ultra-thin fiber of glass. The detector generates an electrical pulse when light falls on it. By attaching a light source to one end of an optical fiber and a detector to the other, we have a unidirectional data transmission system that accepts an electrical signal, converts and transmits it by light pulses, and then reconverts the output to an electrical signal at the receiving end.

This transmission system would leak light and be useless in practice except for an interesting principle of physics. When a light ray passes from one medium to another, for example, from fused silica to air, the ray is refracted (bent) at the silica/air boundary, as shown in Fig. 3.5(a). Here we see a light ray incident on the boundary at an angle a1 emerging at an angle b1. The amount of refraction depends on the properties of the two media (in particular, their indices of refraction). For angles of incidence above a certain critical value, the light is refracted back into the silica; none of it escapes into the air. Thus, a light ray incident at or above the critical angle is trapped inside the fiber, as shown in Fig. 3.5(b), and can propagate for many kilometers with virtually no loss.

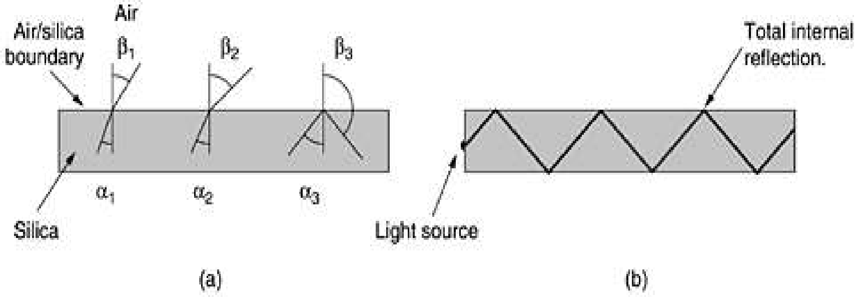


Figure 3.5.(a) Three examples of a light ray from inside a silicafiber impinging on the air/silica boundary at different angles. (b) Light trapped by total internal reflection.

The sketch of Fig. 3.5(b) shows only one trapped ray, but since any light ray incident on the boundary above the critical angle will be reflected internally, many different rays will be bouncing around at different angles. Each ray is said to have a different mode, so a fiber having this property is called a multimode fiber. However, if the fiber's diameter is reduced to a few wavelengths of light, the fiber acts like a wave guide, and the light can propagate only in a straight line, without bouncing, yielding a single-mode fiber. Single-mode fibers are more expensive but are widely used for longer distances. Currently available single-mode fibers can transmit data at 50 Gbps for 100 km without amplification. Even higher data rates have been achieved in the laboratory for shorter distances.

e). Transmission of Light through Fiber

Optical fibers are made of glass, which, in turn, is made from sand, an inexpensive raw material available in unlimited amounts. Glassmaking was known to the ancient Egyptians, but their glass had to be no more than 1 mm thick or the light could not shine through. Glass transparent enough to be useful for windows was developed during the Renaissance. The glass used for modern optical fibers is so transparent that if the oceans were full of it instead of water, the seabed would be as visible from the surface as the ground is from an airplane on a clear day. The attenuation of light through glass depends on the wavelength of the light (as well as on some physical properties of the glass). For the kind of glass used in fibers, the attenuation is shown in Fig. 3.6. in decibels per linear kilometer of fiber. The attenuation in decibels is given by the formula :

$$\text{Attenuation in decibels} = 10 \log_{10} \frac{\text{transmitted power}}{\text{received power}}$$

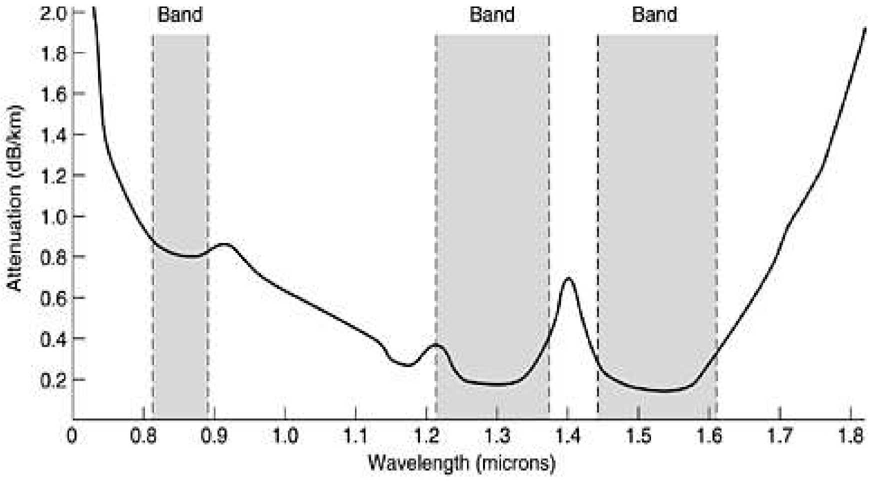


Figure 3.6. Attenuation of Light through Fiber in the Infrared Region.

For example, a factor of two loss gives an attenuation of 10 log10 2 = 3 dB. The figure shows the near infrared part of the spectrum, which is what is used in practice. Visible light has slightly shorter wavelengths, from 0.4 to 0.7 microns (1 micron is 10-6 meters). The true metric purist would refer to these wavelengths as 400 nm to 700 nm, but we will stick with traditional usage. Three wavelength bands are used for optical communication. They are centered at 0.85, 1.30, and 1.55 microns, respectively. The last two have good attenuation properties (less than 5 percent loss per kilometer). The 0.85 micron band has higher attenuation, but at that wavelength the lasers and electronics can be made from the same material (gallium arsenide). All three bands are 25,000 to 30,000 GHz wide.

Light pulses sent down a fiber spread out in length as they propagate. This spreading is called chromatic dispersion. The amount of it is wavelength dependent. One way to keep these spread-out pulses from overlapping is to increase the distance between them, but this can be done only by reducing the signaling rate. Fortunately, it has been discovered that by making the pulses in a special shape related to the reciprocal of the hyperbolic cosine, nearly all the dispersion effects cancel out, and it is possible to send pulses for thousands of kilometers without appreciable shape distortion. These pulses are called solitons. A considerable amount of research is going on to take solitons out of the lab and into the field.

f). Fiber Cables

Fiber optic cables are similar to coax, except without the braid. Figure 3.7(a) shows a single fiber viewed from the side. At the center is the glass core through which the light propagates. In multimode fibers, the core is typically 50 microns in diameter, about the thickness of a human hair. In single-mode fibers, the core is 8 to 10 microns.

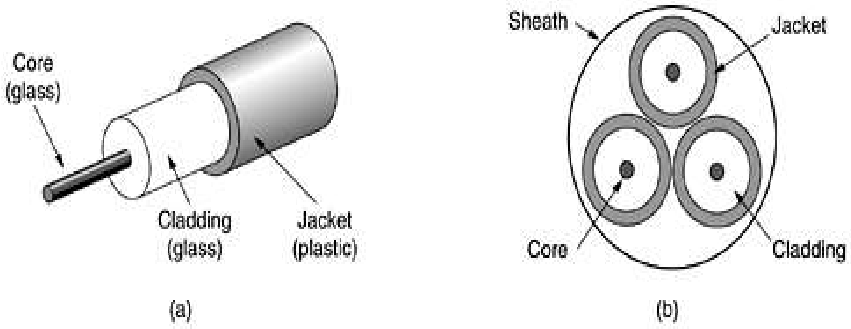


Figure 3.7.(a) Side view of a single fiber. (b) End view of a sheath with three fibers.

The core is surrounded by a glass cladding with a lower index of refraction than the core, to keep all the light in the core. Next come as thin plastic jacket to protect the cladding. Fibers are typically grouped in bundles, protected by an outer sheath. Figure 3-7(b) shows a sheath with three fibers. Terrestrial fiber sheaths are normally laid in the ground within a meter of the surface, where they are occasionally subject to attacks by backhoes or gophers. Near the shore, transoceanic fiber sheaths are buried in trenches by a kind of seaplow. In deep water, they just lie on the bottom, where they can be snagged by fishing trawlers or attacked by giant squid.

Fibers can be connected in three different ways. First, they can terminate in connectors and be plugged into fiber sockets. Connectors lose about 10 to 20 percent of the light, but they make it easy to reconfigure systems. Second, they can be spliced mechanically. Mechanical splices just lay the two carefully-cut ends next to each other in a special sleeve and clamp them in place. Alignment can be improved by passing light through the junction and then making small adjustments to maximize the signal. Mechanical splices take trained personnel about 5 minutes and result in a 10 percent light loss. Third, two pieces of fiber can be fused (melted) to form a solid connection. A fusion splice is almost as good as a single drawn fiber, but even here, a small amount of attenuation occurs. For all three kinds of splices, reflections can occur at the point of the splice, and the reflected energy can interfere with the signal.

Two kinds of light sources are typically used to do the signalling, LEDs (Light Emitting Diodes) and semiconductor lasers. They have different properties, as shown in Table.3.1. They can be tuned in wavelength by inserting Fabry-Perot or Mach-Zehnder interferometers between the source and the fiber. Fabry-Perot interferometers are simple resonant cavities consisting of two parallel mirrors. The light is incident perpendicular to the mirrors. The length of the cavity selects out those wavelengths that fit inside an integral number of times. Mach-Zehnder interferometers separate the light into two beams. The two beams travel slightly different distances. They are recombined at the end and are in phase for only certain wavelengths.

Table 3.1. Comparison of Semiconductor Diodes and LEDs as Light Sources.

S.No.	Basis for Comparison	LED	Semiconductor Laser
1.	Data Rate	Low	High
2.	Fiber Type	Multimode	Multimode or Single Mode
3.	Distance	Short	Long
4.	Lifetime	Long Life	Short Life
5.	Temperature Sensitivity	Minor	Substantial
6.	Cost	Low Cost	Expensive

The receiving end of an optical fiber consists of a photodiode, which gives off an electrical pulse when struck by light. The typical response time of a photodiode is 1 nsec, which limits data rates to about 1 Gbps. Thermal noise is also an issue, so a pulse of light must carry enough energy to be detected. By making the pulses powerful enough, the error rate can be made arbitrarily small.

g). Fiber Optic Networks

Fiber optics can be used for LANs as well as for long-haul transmission, although tapping into it is more complex than connecting to an Ethernet. One way around the problem is to realize that a ring network is really just a collection of point-to-point links, as shown in Fig. 3.8. The interface at each computer passes the light pulse stream through to the next link and also serves as a T junction to allow the computer to send and accept messages.

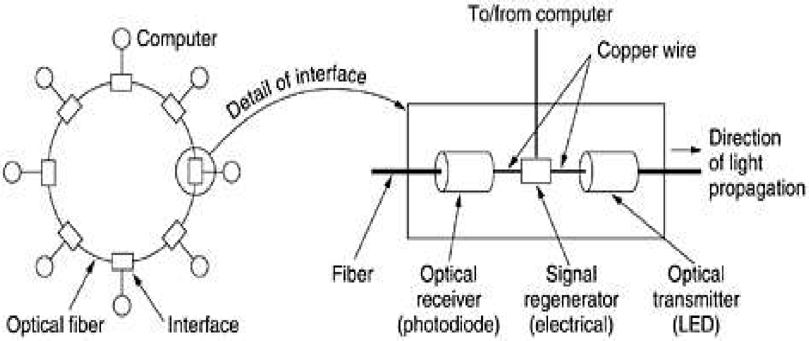

Computer
Detail of interface
To/from computer
Copper wire
Direction of light propagation
Fiber
Optical receiver (photodiode)
Signal regenerator (electrical)
Optical transmitter (LED)
Optical fiber
Interface

***Figure 3.8. Fiber Optic Ring with Active Repeaters*.**

There are two types of interfaces are used. A passive interface consists of two taps fused onto the main fiber. One tap has an LED or laser diode at the end of it (for transmitting), and the other has a photodiode (for receiving). The tap itself is completely passive and is thus extremely reliable because a broken LED or photodiode does not break the ring. It just takes one computer off-line. The other interface type is active repeater shown in Fig. 3.8. The incoming light is converted to an electrical signal, regenerated to full strength if it has been weakened, and retransmitted as light. The interface with the computer is an ordinary copper wire that comes into the signal regenerator. Purely optical repeaters are now being used, too. These devices do not require the optical to electrical to optical conversions, which mean they can operate at extremely high bandwidths.

If an active repeater fails, the ring is broken and the network goes down. On the other hand, since the signal is regenerated at each interface, the individual computer-to-computer links can be kilometers long, with virtually no limit on the total size of the ring. The passive interfaces lose light at each junction, so the number of computers and total ring length are greatly restricted. A ring topology is not the only way to build a LAN using fiber optics. It is also possible to have hardware broadcasting by using the passive star construction of Fig. 3.9.

In this design, each interface has a fiber running from its transmitter to a silica cylinder, with the incoming fibers fused to one end of the cylinder. Similarly, fibers fused to the other end of the cylinder are run to each of the receivers. Whenever an interface emits a light pulse, it is diffused inside the passive star to illuminate all the receivers, thus achieving broadcast. In effect, the passive star combines all the incoming signals and transmits the merged result on all lines. Since the incoming energy is divided among all the outgoing lines, the number of nodes in the network is limited by the sensitivity of the photodiodes.

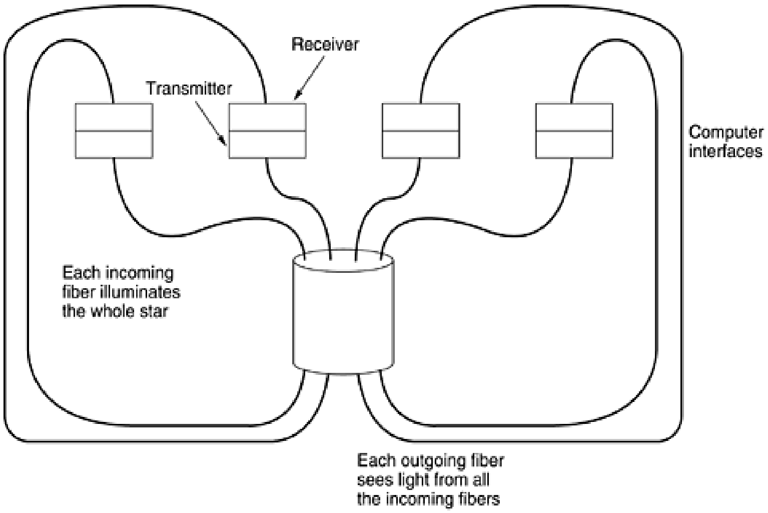

Receiver
Transmitter
Computer interfaces
Each incoming fiber illuminates the whole star
Each outgoing fiber sees light from all the incoming fibers

Figure 3.9. A Passive Star Connection in a Fiber Optics Network.

h). *Comparison of Fiber Optics and Copper Wire*

It is instructive to compare fiber to copper. Fiber has many advantages. To start with, it can handle much higher bandwidths than copper. This alone would require its use in high-end networks. Due to the low attenuation, repeaters are needed only about every 50 km on long lines, versus about every 5 km for copper, a substantial cost saving. Fiber also has the advantage of not being affected by power surges, electromagnetic interference, or power failures. Nor is it affected by corrosive chemicals in the air, making it ideal for harsh factory environments. Oddly enough, telephone companies like fiber for a different reason: it is thin and lightweight. Many existing cable ducts are completely full, so there is no room to add new capacity. Removing all the copper and replacing it by fiber empties the ducts, and the copper has excellent resale value to copper refiners who see it as very high grade ore. Also, fiber is much lighter than copper. One thousand twisted pairs 1 km long weigh 8000 kg. Two fibers have more capacity and weigh only 100 kg, which greatly reduces the need for expensive mechanical support systems that must be maintained. For new routes, fiber wins hands down due to its much lower installation cost.

Finally, fibers do not leak light and are quite difficult to tap. These properties give fiber excellent security against potential wire tappers. On the downside, fiber is a less familiar technology requiring skills not all engineers have, and fibers can be damaged easily by being bent too much. Since optical transmission is inherently unidirectional, two-way communication requires either two fibers or two frequency bands on one fiber. Finally, fiber interfaces cost more than electrical interfaces. Nevertheless, the future of all fixed data communication for distances of more than a few meters is clearly with fiber.

3.3.2 Unguided Media

Our age has given rise to information junkies: people who need to be on-line all the time. For these mobile users, twisted pair, coax, and fiber optics are of no use. They need to get their hits of data for their laptop, notebook, shirt pocket, palmtop, or wristwatch computers without being tethered to the terrestrial communication infrastructure. The wireless communication has many other important applications besides providing connectivity to users who want to surf the Web from the beach. Some people believe that the future holds only two kinds of communication: fiber and wireless. All fixed (i.e., non-mobile) computers, telephones, faxes, and so on will use fiber, and all mobile ones will use wireless.

Wireless has advantages for even fixed devices in some circumstances. For example, if running a fiber to a building is difficult due to the terrain (mountains, jungles, swamps, etc.), wireless may be better. It is noteworthy that modern wireless digital communication began in the Hawaiian Islands, where large chunks of Pacific Ocean separated the users and the telephone system was inadequate.

a). The Electromagnetic Spectrum

When electrons move, they create electromagnetic waves that can propagate through space (even in a vacuum). These waves were predicted by the British physicist James Clerk Maxwell in 1865 and first observed by the German physicist Heinrich Hertz in 1887. The number of oscillations per second of a wave is called its frequency, f, and is measured in Hz (in honor of Heinrich Hertz). The distance between two consecutive maxima (or minima) is called the wavelength, which is universally designated by the Greek letter l (lambda). When an antenna of the appropriate size is attached to an electrical circuit, the electromagnetic waves can be broadcast efficiently and received by a receiver some distance away. All wireless communication is based on this principle.

In vacuum, all electromagnetic waves travel at the same speed, no matter what their frequency. This speed, usually called the speed of light, c, is approximately 3 x 108 m/sec, or about 1 foot (30 cm) per nanosecond. (A case could be made for redefining the foot as the distance light travels in a vacuum in 1 nsec rather than basing it on the shoe size of some long-dead king.) In copper or fiber the speed slows to about 2/3 of this value and becomes slightly frequency dependent. The speed of light is the ultimate speed limit. No object or signal can ever move faster than it. The fundamental relation between f, l, and c (in vacuum) is

$$\lambda * f = C \qquad (3.1)$$

Since c is a constant, if we know f, we can find l, and vice versa. As a rule of thumb, when l is in meters and f is in MHz, If 300 for example, 100-MHz waves are about 3 meters long, 1000-MHz waves are 0.3-meters long, and 0.1-meter waves have a frequency of 3000 MHz.

The electromagnetic spectrum is shown in Fig. 3.10. The radio, microwave, infrared, and visible light portions of the spectrum can all be used for transmitting information by modulating the amplitude, frequency, or phase of the waves. Ultraviolet light, X-rays, and gamma rays would be even better, due to their higher frequencies, but they are hard to produce and modulate, do not propagate well through buildings, and are dangerous to living things. The bands listed at the bottom of Fig. 3.10 are the official ITU names and are based on the wavelengths, so the LF band goes from 1 km to 10 km (approximately 30 kHz to 300 kHz). The terms LF, MF, and HF refer to low, medium, and high frequency, respectively. Clearly, when the names were assigned, nobody expected to go above 10 MHz, so the higher bands were later named the Very, Ultra, Super, Extremely, and Tremendously High Frequency bands. Beyond that there are no names, but Incredibly, Astonishingly, and Prodigiously high frequency (IHF, AHF, and PHF) would sound nice.

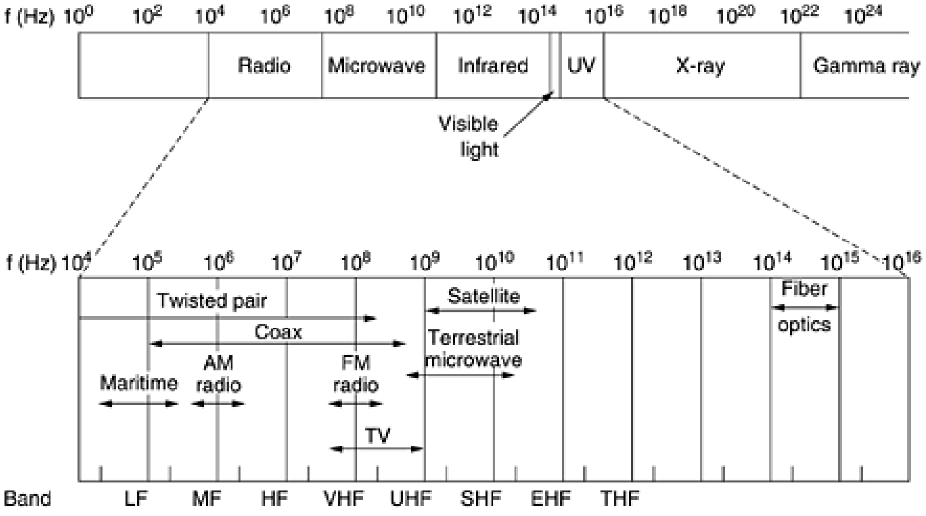


Figure3.10 The Electromagnetic Spectrum and Its Uses for Communication.

3.4 CHECK YOUR PROGRESS QUESTIONS

1. State about Analog Signal.
2. Define Digital Signal.
3. What is meant by Transmission Media?

3.5 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. The analog signal means the transmission power varies over a

NOTES

2. Continuous range with respect to sound, light and radio waves. Analog signal is measured in Volts and its frequency is in Hertz (Hz).

3. The digital signal means only discrete set of values within a given range. A digital signal is a sequence of voltage represented in binary form.

4. The media, over which the information between two computer systems is sent, called transmission media.

3.6 SUMMARY

The physical layer is the basis of all networks. Nature imposes two fundamental limits on all channels, and these determine their bandwidth. These limits are the Nyquist limit, which deals with noiseless channels, and the Shannon limit, which deals with noisy channels. Transmission media can be guided or unguided. The principal guided media are twisted pair, coaxial cable, and fiber optics. Unguided media include radio, microwaves, infrared, and lasers through the air. An up-and-coming transmission system is satellite communication, especially LEO systems. A key element in most wide area networks is the telephone system. Its main components are the local loops, trunks, and switches. Local loops are analog, twisted pair circuits, which require modems for transmitting digital data. ADSL offers speeds up to 50 Mbps by dividing the local loop into many virtual channels and modulating each one separately.

3.7 KEY WORDS

- **Physical Layer:**The physical layer defines mechanical, electrical and timing interfaces to the network.

- **Guided Media:** All communication wires or cables are guided media, such as UTP, coaxial cables, and fiber Optics. In this media, the sender and receiver are directly connected and the information is send (guided) through it.

- **Unguided Media:** Wireless or open air space is said to be unguided media, because there is no connectivity between the sender and receiver. Information is spread over the air, and anyone including the actual recipient may collect the information.

3.8 SELF ASSESSMENT QUESTIONS AND EXERCISES

Small Questions

1. Define the use of coaxial cable.
2. Mention about Fiber Optic Networks.
3. What is magnetic media?

Long Answer Questions

1. Discuss about Asynchronous and Synchronous Transmission.
2. Explain about guided and unguided media.

3.9 FURTHER READINGS

1. Computer Networks, 3rd Edition, Andrew S Tanenbaum, Pearson Education, 2010.
2. Data and Computer Communications, William Stallings, 8th Edition, Prentice Hall, 2007.
3. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.Bellamy, Digital Telephony.
4. Palais, Fiber Optic Communication, 3rd ed.
5. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.

BLOCK 2: DATA LINK LAYER

UNIT 4: DATA LINK LAYER

Structure

4.0 INTRODUCTION

In this unit, you will learn about functions of data link layer, design issues with data link layer, error detection and correction methods. Also illustrates about flow and error control algorithms and block coding system. According to data link layer, the packet is passed across the interface to it from the network layer as pure data, whose every bit is to be delivered to the destination's network layer. The reality that the destination's network layer may interpret part of the packet as a header is of no concern to the data link layer.

4.1 OBJECTIVES

- Subsequent to going through this unit, you will elucidate about data link layer.
- Error Detection and Correction techniques are given in detailed manner.
- Cyclic Redundancy Check
- Framing process is utilized for the data packets transmission.

4.2 INTRODUCTION TO DATA LINK LAYER

Data link layer performs the most reliable node to node delivery of data. It forms frames from the packets that are received from network layer and gives it to physical layer. It also synchronizes the information which is to be transmitted over the data. Error controlling is easily done. The encoded data are then passed to physical. Error detection bits are used by the data link layer. It also corrects the errors. Outgoing messages are assembled into frames. Then the system waits for the acknowledgements to be received after the transmission. It is reliable to send message.

The main task of the **data link layer** is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into **data frames** (typically a few hundred or few thousand bytes) and transmits the frames sequentially. If the service is reliable, the receiver confirms correct receipt of each frame by send back an **acknowledgement frame**.

4.2.1 Functions of Data Link Layer

i). **Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.

ii). **Physical Addressing:** The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.

iii). **Flow Control:** A flow control mechanism to avoid a fast transmitter from running a slow receiver by buffering the extra bit is provided by flow control. This prevents traffic jam at the receiver side.

iv). **Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames is also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

v). **Access Control:** Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.

4.2.2 Design Issues with Data Link Layer

i). The issue that arises in the data link layer (and most of the higher layers as well) is how to keep a fast transmitter from drowning a slow receiver in data. Some traffic regulation mechanism is often needed to let the transmitter know how much buffer space the receiver has at the moment. Frequently, the flow regulation and the error handling are integrated.

ii). Broadcast networks have an additional issue in the data link layer: How to control access to the shared channel. A special sublayer of the data link layer, the Medium Access Control (MAC) sub layer, deals with this problem.

4.3 ERROR DETECTION AND CORRECTION

The telephone system has three parts: the switches, the interoffice trunks, and the local loops. The first two are now almost entirely digital in most developed countries. The local loops are still analog twisted copper pairs and will continue to be so for years due to the enormous expense of replacing them. While errors are rare on the digital part, they are still common on the local loops. Furthermore, wireless communication is becoming more common, and the error rates here are orders of magnitude worse than on the interoffice fiber trunks. The transmission errors are prolonged for many years and take measures how to solve such problems.

Consequently of the physical processes that generate them, errors on some media (e.g., radio) tend to come in bursts rather than singly. Having the errors come in bursts has both advantages and disadvantages over isolated single-bit errors. On the advantage side, computer data are always sent in blocks of bits. Suppose that the block size is 1000 bits and the error rate is 0.001 per bit. If errors were independent, most blocks would contain an error. If the errors came in bursts of 100 however, only one or two blocks in 100 would be affected, on average. The disadvantage of burst errors is that they are much harder to correct than are isolated errors.

4.4. BLOCK CODING

In 1950 the American mathematician Richard Hamming was established block code after that named as "Hamming code". A block code is used to convert software code or an algorithm into any particular form so that errors, if any, in the code can be minimized. Block code can also be applied in the domains of telecommunications, information theory and coding theory. The main idea is to encode a message for a recipient in such a way that the recipient is able to address errors, if any, in the message with the help of the encoding.

The following codes are examples for block codes; they are Reed–Solomon codes, Hamming codes, Hadamard codes, Expander codes, Golay codes, and Reed–Muller codes. These examples also belong to the class of linear codes, and hence they are called linear block codes. More particularly, these codes are known as algebraic block codes, or cyclic block codes, because they can be generated using Boolean polynomials.

The Algebraic block codes are typically hard-decoded using algebraic decoders. **Block coding** are utilized to detect errors and re-transmission of the signal. It is normally referred to as mB/nB coding as it replaces each m-bit data group with an n-bit data group (where n>m). Thus, its adds extra bits (redundancy bits) which helps in synchronization at receiver's and sender's end and also providing some kind of error detecting capability.

It normally involves three steps: division, substitution, and combination. In the division step,a sequence of bits is divided into groups of m-bits. In the substitution step, we substitute an m-bit group for an n-bit group. Finally, the n-bit groups are combined together to form a stream which has more bits than the original bits.

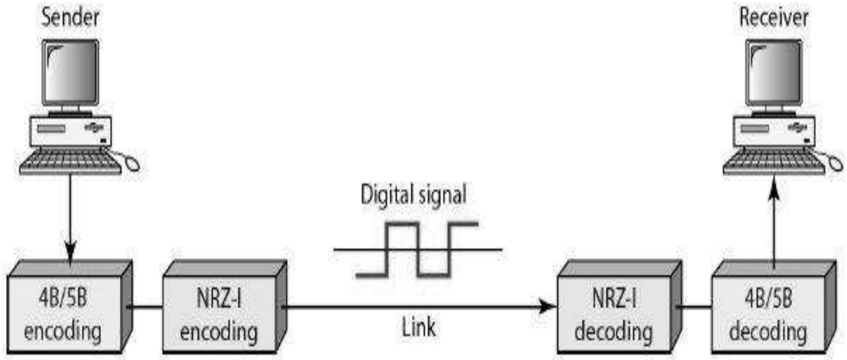


Figure4. 1. Block Coding

- **Examples of mB/nB Coding: 4B/5B (Four Binary/Five Binary)**

This coding scheme is used in combination with NRZ-I. The problem with NRZ-I was that it has a synchronization problem for long sequences of zeros. So, to overcome it we substitute the bit stream from 4-bit to 5-bit data group **before encoding it with NRZ-I**. So that it does not have a long stream of zeros. The block-coded stream does not have more than three consecutive zeros. At the receiver, the NRZ-I encoded digital signal is first decoded into a stream of bits and then decoded again to remove the redundancy bits.

- **Drawback** – Though 4B/5B encoding solves the problem of synchronization,it increases the signal rate of NRZ-L.Moreover,it does not solve the DC component problem of NRZ-L.The encoding scheme is offered in table 4.1.

Table 4.1.Encoding Scheme

S.No.	4-bit Data	5-bit code	S.No.	4-bit Data	5-bit code
1	0000	11110	9	1000	10010
2	0001	01001	10	1001	10011
3	0010	10100	11	1010	10110
4	0011	10101	12	1011	10111
5	0100	01010	13	1100	11010
6	0101	01011	14	1101	11011
7	0110	01110	15	1110	11100
8	0111	01111	16	1111	11101

- **8B/10B(eightbinary/tenbinary):**

This encoding is similar to 4B/5B encoding except that a group of 8 bits of data is now substituted by a 10-bit code and it provides greater error detection capability than 4B/5B.It is actually a combination of 5B/6B and 3B/4B encoding. The most five significant bits of a 10-bit block is fed into the 5B/6B encoder; the least 3 significant bits is fed into a 3B/4B encoder. The split is done to simplify the mapping table.

4.5. CYCLIC REDUNDANCY CHECK

The Cyclic Redundancy Check (CRC) is an error detection mechanism in which a special number is appended to a block of data in order to detect any changes introduced during storage or transmission. The CRC is recalculated on retrieval or reception and compared to the value originally transmitted, which can reveal certain types of error. For example, a single corrupted bit in the data results in a one-bit change in the calculated CRC, but multiple corrupt bits may cancel each other out.

A CRC is derived using a more complex algorithm than the simple CHECKSUM, involving MODULO ARITHMETIC (hence the 'cyclic' name) and treating each input word as a set of coefficients for a polynomial.

i). CRC is more powerful than VRC and LRC in detecting errors.
ii). It is not based on binary addition like VRC and LRC. Rather it is based on binary division.

iii). At the sender side, the data unit to be transmitted IS divided by a predetermined divisor (binary number) in order to obtain the remainder. This remainder is called CRC.

iv). The CRC has one bit less than the divisor. It means that if CRC is of n bits, divisor is of n+ 1 bit.

v). The sender appends this CRC to the end of data unit such that the resulting data unit becomes exactly divisible by predetermined divisor i.e. remainder becomes zero.

vi). At the destination, the incoming data unit i.e. data + CRC is divided by the same number (predetermined binary divisor).

vii). If the remainder after division is zero then there is no error in the data unit & receiver accepts it.

viii). If remainder after division is not zero, it indicates that the data unit has been damaged in transit and therefore it is rejected.

ix). This technique is more powerful than the parity check and checksum error detection.

x). CRC is based on binary division. A sequence of redundant bits called CRC or CRC remainder is appended at the end of a data unit such as byte.

Requirements of CRC:

A CRC will be valid if and only if it satisfies the following requirements:

a) It should have exactly one less bit than divisor.
b) Appending the CRC to the end of the data unit should result in the bit sequence which is exactly divisible by the divisor.

The various steps followed in the CRC method are

a) A string of n as is appended to the data unit. The length of predetermined divisor is n+ 1.
b) The newly formed data unit *i.e.* original data + string of n as are divided by the divisor using binary division and remainder is obtained. This remainder is called CRC.

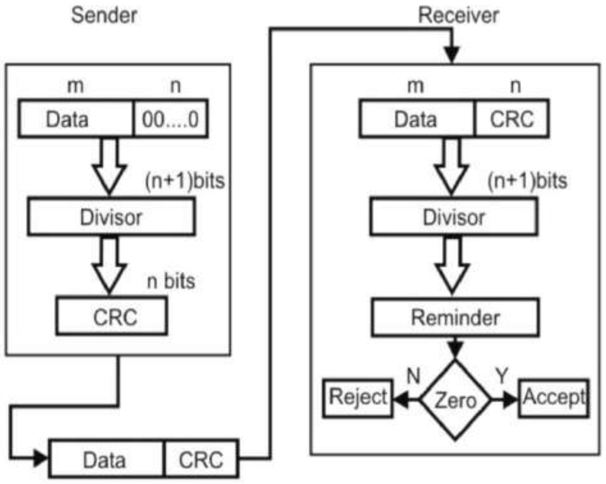


Figure 4.2. Process of Cyclic Redundancy Check

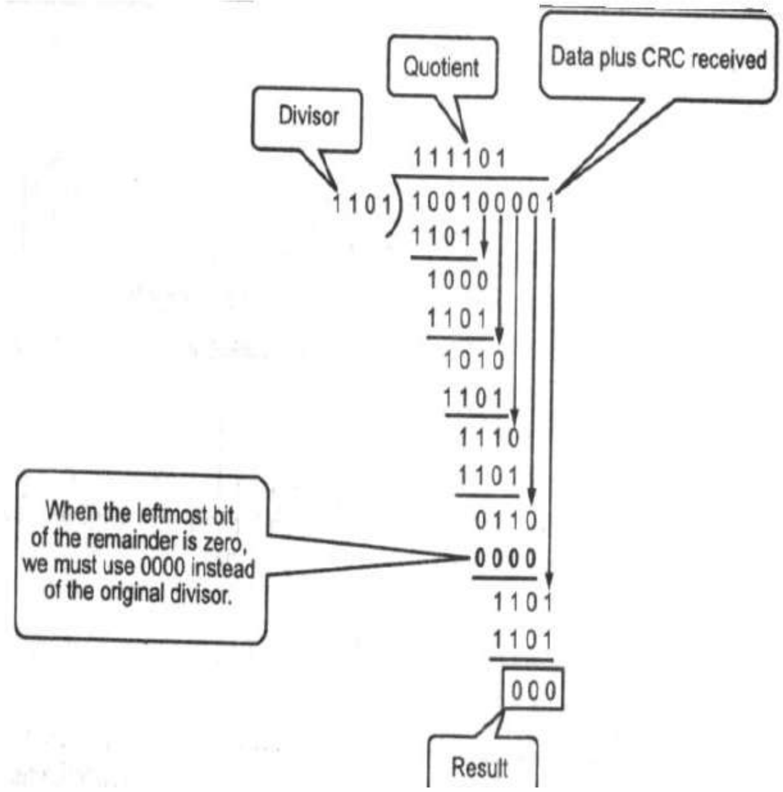


Figure 4.3. CRC Generation

4.6. FRAMING

The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another. The Data Link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame. Framing is a function of the data link layer. Framing is a point-to-point connection between two computers or devices consists of a wire in which data is transmitted as a stream of bits. However, these bits must be framed into visible blocks of information. It provides a way for a sender to transmit a set of bits that are meaningful to the receiver. Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures. Frames have headers that contain information such as error-checking codes.The Data Link layer frame includes:

i). **Data** - The packet from the Network layer.

ii). **Header** - Contains control information, such as addressing, and is located at the beginning of the PDU.

iii). **Trailer** - Contains control information added to the end of the PDU.

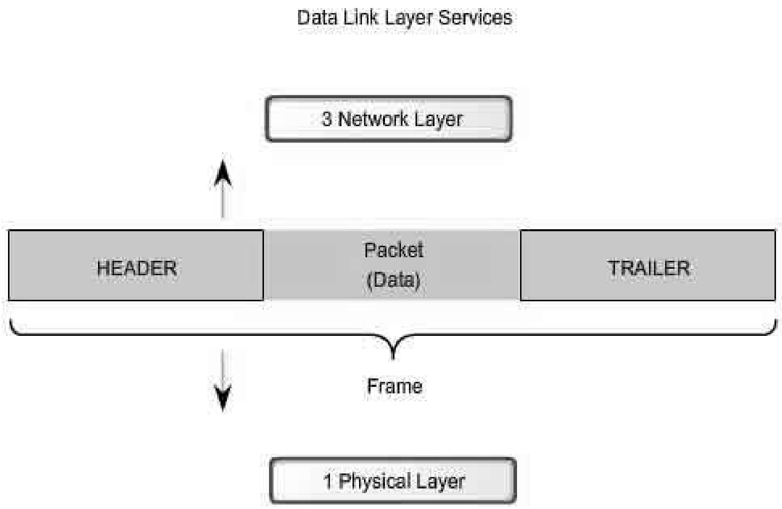


Figure 4.4.Framing in Data link Layer

At data link layer, it extracts message from sender and provide it to receiver by providing sender’s and receiver’s address. The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.

a). Problems in Framing –

i). **Detecting start of the frame:** When a frame is transmitted, every station must be able to detect it. Station detect frames by looking out for special sequence of bits that marks the beginning of the frame i.e. SFD (Starting Frame Delimeter).

ii). **How do station detect a frame:** Every station listen to link for SFD pattern through a sequential circuit. If SFD is detected, sequential circuit alerts station. Station checks destination address to accept or reject frame.

iii). **Detecting end of frame:** When to stop reading the frame.

b). Types of framing – There are two types of framing:

i). **Fixed size –** The frame is of fixed size and there is no need to provide boundaries to the frame, length of the frame itself acts as delimiter.

- **Drawback:** It suffers from internal
- fragmentation if data size is less than frame size
- **Solution:** Padding

ii). **Variable size –** In this there is need to define end of frame as well as beginning of next frame to distinguish. This can be done in two ways:

➢ **Length field –** We can introduce a length field in the frame to indicate the length of the frame. Used in **Ethernet(802.3)**. The problem with this is that sometimes the length field might get corrupted.

- **End Delimiter (ED)** – We can introduce an ED (pattern) to indicate the end of the frame. Used in **Token Ring**. The problem with this is that ED can occur in the data.

4.7. Flow and Error Control

a). Flow Control:

Flow control coordinates that amount of data that can be sent before receiving acknowledgement.

i). It is one of the most important duties of the data link layer.
ii). Flow control tells the sender how much data to send.
iii). It makes the sender wait for some sort of an acknowledgment (ACK) before continuing to send more data.
iv). *Flow Control Techniques: Stop-and-wait, and Sliding Window*

b). Error Control:

Error control in the data link layer is based on ARQ (automatic repeat request), which is the retransmission of data.

i). The term error control refers to methods of error detection and retransmission.
ii). Anytime an error is detected in an exchange, specified frames are retransmitted. This process is called ARQ.

To ensure reliable communication, there needs to exist flow control (managing the amount of data the sender sends), and error control (that data arrives at the destination error free).

i). Flow and error control needs to be done at several layers.
ii). For node-to-node links, flow and error control is carried out in the data-link layer.
iii). For end-point to end-point, flow and error control is carried out in the transport layer.

c). Flow & Error control: Error Detection and ARQ (error detection with retransmissions) must be combined with methods that intelligently limit the number of ‘outstanding’ (unACKed) frames.

4.8 CHECK YOUR PROGRESS QUESTIONS

1. State the role of Data link layer.
2. Define physical address.
3. What is CRC?

4.9 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. The foremost assignment of the data link layer is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer. It accomplishes this task by having the sender break up the input data into data frames and transmits the frames sequentially.

2. The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.

3. The Cyclic Redundancy Check (CRC) is an error detection mechanism in which a special number is appended to a block of data in order to detect any changes introduced during storage or transmission. The CRC is recalculated on retrieval or reception and compared to the value originally transmitted, which can reveal certain types of error.

4.10 SUMMARY

The responsibility of the data link layer is to convert the raw bit stream offered by the physical layer into a stream of frames for use by the network layer. Various farming methods are used, including character count, byte stuffing and bit stuffing. Data link protocols can provide error control to retransmit damaged or lost frames. To prevent a fast sender from overrunning a slow receiver, the data link protocol can also provide flow control.

4.11 KEY WORDS

- **Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.

- **Flow Control:** A flow control mechanism to avoid a fast transmitter from running a slow receiver by buffering the extra bit is provided by flow control. This prevents traffic jam at the receiver side.

- **Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

- **Access Control:** Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.

4.12 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Define Error.
2. Differentiate Error Control and Flow Control.
3. What is meant by Framing?

Long Answer Questions

1. Discuss about Cyclic Redundancy Check.

2. Frames of 1000 bits are sent over a 1-Mbps channel using a geostationary satellite whose propagation time from the earth is 270 msec. Acknowledgements are always piggybacked onto data frames. The headers are very short. Three-bit sequence numbers are used. What is the maximum achievable channel utilization for a. (a) Stop-and-wait (b) Protocol 5 (c) Protocol 6.

3. In protocol 6 the code for frame arrival has a section used for NAKs. This section is invoked if the incoming frame is a NAK and another condition is met. Give a scenario where the presence of this other condition is essential.

4.13 FURTHER READINGS

1. Behrouz A. Forouzan, Data Communications and Networking, Fourth Edition, TataMcGraw Hill, 2009.

2. Gravano, Introduction to Error Control Codes.Computer Networks, 3rd Edition, Andrew S Tanenbaum, Pearson Education, 2010.

3. Data and Computer Communications, William Stallings, 8th Edition, Prentice Hall, 2007.

4. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.

UNIT-5 DATA LINK LAYER PROTOCOLS

Structure

5.0 INTRODUCTION

In this unit, we can have the details about the design principles, algorithms and protocols for data link layer concerning proficient communication between two adjacent machines. In addition to that the data link layer protocols are elucidated in detailed manner with suitable diagrams.

5.1 OBJECTIVES

After studying this unit, we will understand and express about

- Data Link Layer Protocols
- Stop-and-Wait Protocol
- Sliding Window Protocols
- Automatic Repeat reQuest(ARQ) Protocols
- Stop-and-Wait ARQ Protocol
- Go-Back-n ARQ Protocol
- Selective Repeat ARQ Protocol

5.2 INTRODUCTION TO DATA LINK LAYER PROTOCOLS

The Data Link Layer is responsible for transmission of data between two nodes. One important aspect of data link layer is flow control. Flow control refers to a set of procedures used to restrict the amount of data the sender can send before waiting for acknowledgement. The sender has to wait for an acknowledgment of every frame that it sends. Its main functions are Data Link Control and Multiple Access Control.

Data Link control : The data link control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control. For Data link control refer to Stop and Wait ARQ.

Multiple Access Control: If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then multiple stations can access the channel simultaneously.

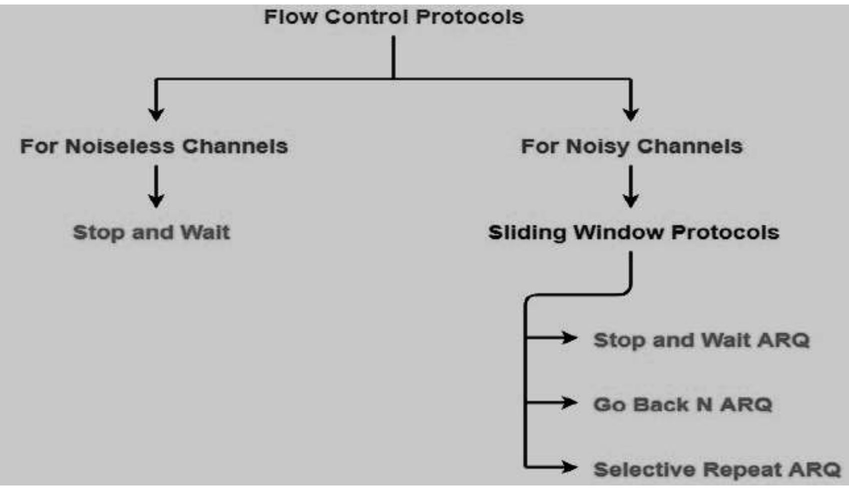


Figure 5.1. Categories of Flow Control Protocols

5.3 STOP AND WAIT PROTOCOL

Only when a acknowledgment has been received is the next frame sent. This process continues until the sender transmits an End of Transmission (EOT) frame. In Stop-and-Wait flow control, the receiver indicates its readiness to receive data for each frame. For every frame that is sent, there needs to be an acknowledgment, which takes a similar amount of propagation time to get back to the sender. Only one frame can be in transmission at a time. This leads to inefficiency if propagation delay is much longer than the transmission delay. The working rule of Simple Stop and Wait protocol is given below:

- **Sender:**

Rule 1) Send one data packet at a time.

Rule 2) Send next packet only after receiving acknowledgement for the previous.

- **Receiver:**

Rule 1) Send acknowledgement after receiving and consuming of data packet.

Rule 2) After consuming packet acknowledgement need to be sent (Flow Control).

a). Characteristics of Stop and Wait Protocol

i). Include re-transmission of data in case of lost or damaged framer.
ii). It is addition to the basic flow control mechanism with re-transmissions.
iii). Sender sends an information frame to receiver.
iv). Sender waits for an ACK before sending the next frame.
v). Receiver sends an ACK if frame is correctly received.
vi). If no ACK arrives within time-out, sender will resend the frame.
Time-out period >Rround trip time
vii). If an error is discovered in a data frame, indicating that it has been corrupted in transit, a NAK frame is returned. NAK frames, which are numbered, tell the sender to retransmit the last frame sent.

b). Advantages of Stop and Wait:

i). It's simple and each frame is checked and acknowledged well.

c). Disadvantages of Stop and Wait:

i). Only one frame can be in transmission at a time.
ii). It is inefficient, if the distance between devices is long. Reason is propagation delay is much longer than the transmission delay.
iii). The time spent for waiting acknowledgements between each frame can add significant amount to the total transmission time.

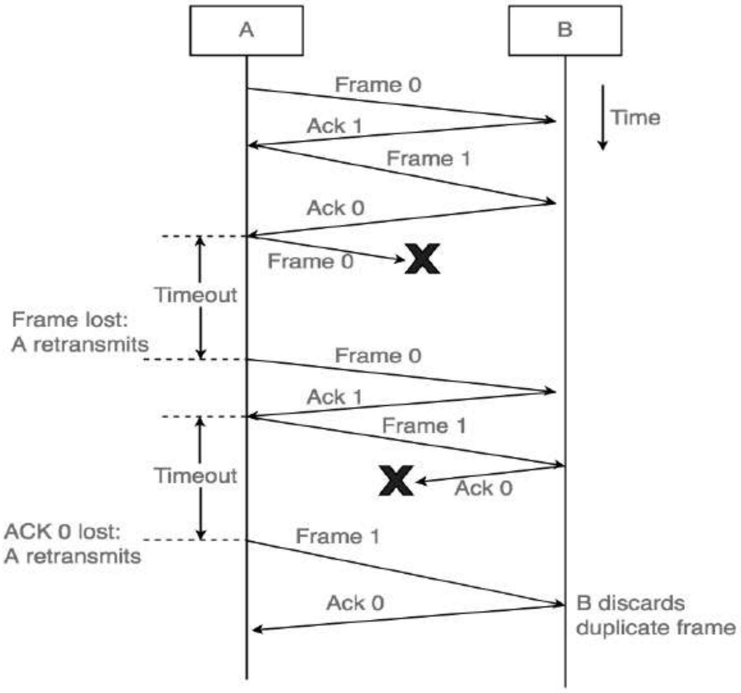


Figure 5.2.Stop and Wait Flow control

5.4 SLIDING WINDOW PROTOCOL

The **sliding window protocol** is a feature of packet-based data transmission **protocols**. By placing limits on the number of packets that can be transmitted or received at any given time, a **sliding window protocol** allows an unlimited number of packets to be communicated using fixed-size sequence numbers.

It works by having the sender and receiver have a “window” of frames.

i). Each frame has to be numbered in relation to the sliding window. For a window of size n, frames get a number from 0 to n - 1. Subsequent frames get a number mod n.

ii). The sender can send as many frames as would fit into a window.

iii). The receiver, upon receiving enough frames, will respond with an acknowledgment of all frames up to a certain point in the window. It is called slide.

iv). This window can hold frames at either end and provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgement.

v). To cover retransmission of lost or damaged frames, some features are added to the basic flow control mechanism of sliding window.

vi). A Sender may send multiple frames as allowed by the window size.

vii). The sending device keeps copies of all transmitted frames, until they have been acknowledged. .

viii). In addition to ACK frames, the receiver has the option of returning a NAK frame, if the data have been received damaged. NAK frame tells the sender to retransmit a damaged frame.

ix). Here, both ACK and NAK frames must be numbered for identification.

x). ACK frames carry the number of next frame expected.

xi). NAK frames on the other hand, carry the number of the damaged frame itself.

xii). If the last ACK was numbered 3, an ACK 6 acknowledges the receipt of frames 3, 4 and 5 as well.

xiii). If data frames 4 and 5 are received damaged, both NAK 4 and NAK 5 must be returned.

xiv). Like stop and wait ARQ, the sending device in sliding window ARQ is equipped with a timer to enable it to handle lost acknowledgements.

xv). Sliding window ARQ is two types: Go-back-n ARQ, and Selective Reject ARQ.

xvi). There are two ACK processing methods in sliding windows:

- **Selective ACK:** The **ACK N** message acknowledges **only** the frame with sequence number **N.**

- **Cumulative ACK** : The **ACK N** message acknowledges **all** frames with sequence number **<= N.**

For example, if n = 8, the frames are numbered 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1...so on. The size of the window is (n -1) = 7. When the receiver sends an ACK, it includes the number of the next frame it expects to receive. When the receiver sends an ACK containing the number 5, it means all frames upto number 4 have been received.

If the window size is sufficiently large the sender can continuously transmit packets:

- If W >= (2a+1), sender can transmit continuously. **(Efficiency =1)**

- If W < (2a+1), sender can transmit W frames every (2a+1) time units. **(Efficiency = W/(1+2a))**

Piggybacking: In bidirectional communications, both parties send & acknowledge data, i.e. both parties implement flow control. Outstanding ACKs are placed in the header of information frames, piggybacking can save bandwidth since the overhead from a data frame and an ACK frame (addresses, CRC, etc) can be combined into just one frame.

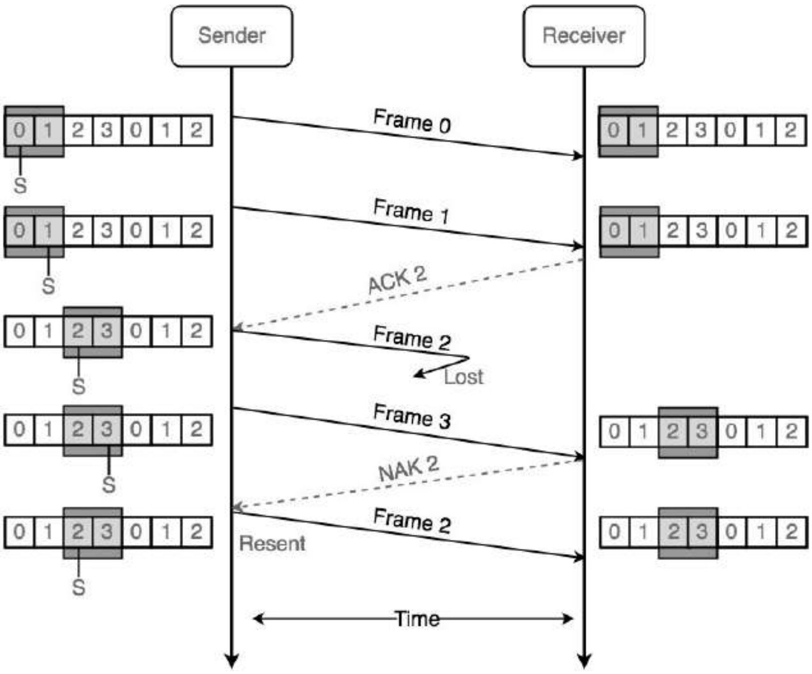


Figure 5.3. Sliding Window Flow Protocol

5.4.1 AUTOMATIC REPEAT REQUEST (ARQ)

Automatic Repeat ReQuest (ARQ), also called Automatic Repeat Query, is an error-control protocol that automatically initiates a call to retransmit any data packet or frame after receiving flawed or incorrect data. When the transmitting device fails to receive an acknowledgement signal to confirm the data has been received, it usually retransmits the data after a predefined timeout and repeats the process a predetermined number of times until the transmitting device receives the acknowledgement.

.

ARQs are often used to assure reliable transmissions over an unreliable service. These protocols reside in the Data Link Layer and in the Transport Layer of the OSI (Open Systems Interconnection) reference model. They are named so because they provide for automatic retransmission of frames that are corrupted or lost during transmission. ARQ is also called Positive Acknowledgement with Retransmission (PAR). ARQs are used to provide reliable transmissions over unreliable upper layer services. They are often used in Global System for Mobile (GSM) communication.

According to ARQ protocols the receiver sends an acknowledgement message back to the sender if it receives a frame correctly. If the sender does not receive the acknowledgement of a transmitted frame before a specified period of time, i.e. a timeout occurs, the sender understands that the frame has been corrupted or lost during transit. So, the sender retransmits the frame. This process is repeated until the correct frame is transmitted.

Three types of ARQ protocols are available namely Stop-and-wait ARQ, Go-Back-N ARQ, and Selective Repeat ARQ/Selective Reject ARQ. All three protocols usually use some form of sliding window protocol to tell the transmitter to determine which (if any) packets need to be retransmitted. These protocols reside in the data link or transport layers (layers 2 and 4) of the OSI model.

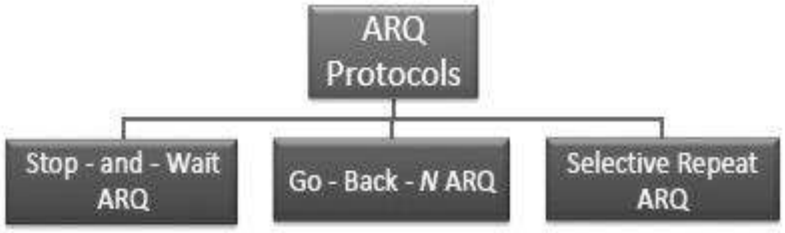


Figure 5.4. Types of ARQ protocols

5.4.2 STOP – AND – WAIT ARQ

The Stop – and – wait ARQ protocol offers unidirectional data transmission with flow control and error control mechanisms, appropriate for noisy channels. The sender keeps a copy of the sent frame. It then waits for a finite time to receive a positive acknowledgement from receiver. If the timer expires, the frame is retransmitted. If a positive acknowledgement is received then the next frame is sent.

a). Characteristics of Stop and Wait ARQ:

i). Used in Connection-oriented communication.
ii). It offers error and flow control
iii). It is used in Data Link and Transport Layers
iv). Stop and Wait ARQ mainly implements Sliding Window Protocol concept with Window Size 1.
v). It uses link between sender and receiver as half duplex link
vi). Throughput = 1 Data packet/frame per RTT
vii). If Bandwidth*Delay product is very high, then stop and wait protocol is not so useful. The sender has to keep waiting for acknowledgements before sending the processed next packet.
viii). It is an example for "**Closed Loop OR connection oriented** " protocols
ix). It is an special category of SWP where its window size is 1
x). Irrespective of number of packets sender is having stop and wait protocol requires only 2 sequence numbers 0 and 1.

The Stop and Wait ARQ solves main three problems, but may cause big performance issues as sender always waits for acknowledgement even if it has next packet ready to send. Consider a situation where you have a high bandwidth connection and propagation delay is also high (you are connected to some server in some other country though a high speed connection). To solve this problem, we can send more than one packet at a time with a larger sequence numbers. So Stop and Wait ARQ may work fine where propagation delay is very less for example LAN connections, but performs badly for distant connections like satellite connection.

b). Useful Terms:

- **Propagation Delay:** Amount of time taken by a packet to make a physical journey from one router to another router.

Propagation Delay = (Distance between routers) / (Velocity of propagation)

- RoundTripTime (**RTT**) = 2* Propagation Delay
- TimeOut (**TO**) = 2* RTT
- Time To Live (**TTL**) = 2* TimeOut. (Maximum TTL is 180 seconds)

The three problems of simple Stop and Wait Protocol are resolved by Stop and Wait ARQ that does both error control and flow control.

c). Working of Stop and Wait ARQ:

1) Sender A sends a data frame or packet with sequence number 0. 2) Receiver B, after receiving data frame, sends and acknowledgement with sequence number 1 (sequence number of next expected data frame or packet)

There is only one bit sequence number that implies that both sender and receiver have buffer for one frame or packet only.

5.4.3 GO – BACK – N ARQ

Go- Back- N ARQ provides for sending multiple frames before receiving the acknowledgement for the first frame. It uses the concept of sliding window, and so is also called sliding window protocol. The frames are sequentially numbered and a finite number of frames are sent. If the acknowledgement of a frame is not received within the time period, all frames starting from that frame are retransmitted.

i). The sliding window method using **cumulative ACK** is known as the **Go-Back-N** ARQ protocol.
ii). Receiver window size is 1.
iii). In this method, if one frame is lost or damaged all frames sent, since the last frame acknowledged are retransmitted.
iv). For example, sender may send frames 1,2,3,4 and get an NAK with a value of 2. The NAK acknowledges everything that came before it, and asks for frame 2 (and subsequent frames) to be resent.
v). NAK number refer to the next expected frame number.
vi). Example: In the following figure, frame 2 has an error, then all subsequent frames are discarded. After timeout sender sends all frames from frame 2.

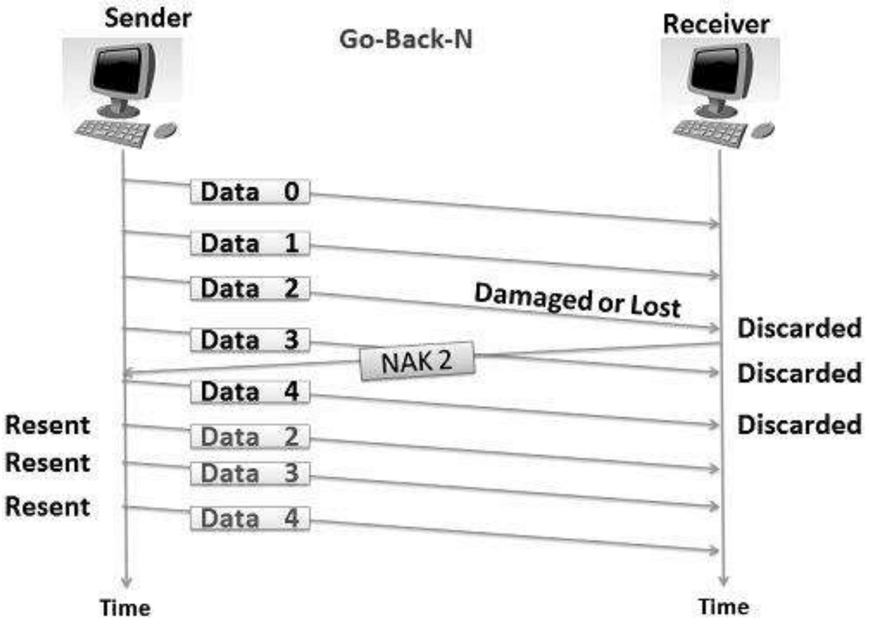


Figure 5.5. Go-Back-N ARQ

a). Damaged/Error Frame :

i). In go-back-n ARQ, The receiver sends the NAK for this frame along with that frame number, that it expects to be retransmitted.

ii). After sending NAK, the receiver discards all the frames that it receives, after a damaged frame.

iii). The receiver does not send any ACK (acknowledgement) for the discarded frames. After the sender receives the NAK for the damaged frame, it retransmits all the frames onwards the frame number referred by NAK.

b). Lost frame:

i). In go-back-n ARQ, Receiver easily detects the loss of a frame as the newly received frame is received out of sequence.

ii). The receiver sends the NAK for the lost frame and then the receiver discards all the frames received after a lost frame.

iii). The receiver does not send any ACK for that discarded frames.

iv). After the sender receives the NAK for the lost frame, it retransmits the lost frame referred by NAK and also retransmits all the frames which it has sent after the lost frame.

c). Lost Acknowledgement :

i). In go-back-n ARQ, If the sender does not receive any ACK or if the ACK is lost or damaged in between the transmission.

ii). The sender waits for the time to run out and as the time run outs, the sender retransmits all the frames for which it has not received the ACK.

iii). The sender identifies the loss of ACK with the help of a timer.

iv). The ACK number, like NAK number, shows the number of the frame, that receiver expects to be the next in sequence.

v). The window size of the receiver is 1 as the data link layer only require the frame which it has to send next to the network layer.

vi). The sender window size is equal to ‘w’. If the error rate is high, a lot of bandwidth is lost wasted.

d). Go Back N ARQ Algorithm

```
N  = window size
Sn = sequence number
Sb = sequence base
Sm = sequence max
ack = ack number
nack = first non acknowledged
Receiver:
Do the following forever:
  Randomly accept or reject packet
  If the packet received and the packet is error free
        Accept packet
        Send a positive ack for packet
  Else
        Refuse packet
        Send a negative ack for packet
Sender:
Sb = 0
Sm = N − 1
ack = 0
Repeat the following steps forever:
  Send packet with ack
  If positively ack is recieved:
ack++
         Transmit a packet where Sb<= ack<= Sm.
         packets are transmitted in order
  Else
Enqueue the nack into the queue
  //check if last packet in the window is sent
  if(ack==Sm)
    if(queue is not empty)
          // start from the first nack packet
nack = queue.front();
         empty the queue
ack = nack
Sm = Sm + (ack − Sb)
Sb = ack
```

5.4.4 SELECTIVE REPEAT ARQ

The Selective Repeat ARQ protocol also provides for sending multiple frames before receiving the acknowledgement for the first frame. However, here only the erroneous or lost frames are retransmitted, while the good frames are received and buffered.

Selective Repeat ARQ overcomes the limitations of Go-Back-N by adding two new features:

i). Receiver window > 1 frame: Out-of-order but error-free frames can be accepted

ii). Retransmission mechanism is modified: Only individual frames are retransmitted.

In this method, only specific damaged or lost frame is retransmitted. Sender only retransmits frames for which a NAK is received. NAK number refers to the frame lost. If a frame is corrupted in transmit, a NAK is returned and the frame is resent out of sequence. The sender needs to maintain all data that hasn’t been acknowledged yet. The receiving device must be able to sort the frames it has and insert the retransmitted frame into its proper place in the sequence. It has advantage that few re-transmissions than go-back-n. But complexity at sender and receiver is involved.

For example,the Frame 2 has an error, so receiver maintains buffer to store the next frames.

a). Damaged Frames :

i). In Selective Repeat ARQ, If a receiver receives a damaged frame, it sends the NAK for the frame in which error or damage is detected.

ii). The NAK number like in go-back-n also indicate the acknowledgement of the previously received frames and error in the current frame.

iii). The receiver keeps receiving the new frames while waiting for the damaged frame to be replaced.

iv). The frames that are received after the damaged frame are not be acknowledged until the damaged frame has been replaced.

b). Lost Frame :

i). As in a selective repeat protocol, a frame can be received out of order and further they are sorted to maintain a proper sequence of the frames.

ii). While sorting, if a frame number is skipped, the receiver recognises that a frame is lost and it sends NAK for that frame to the sender.

iii). After receiving NAK for the lost frame the sender searches that frame in its window and retransmits that frame.

iv). If the last transmitted frame is lost then receiver does not respond and this silence is a negative acknowledgement for the sender.

c). Lost Acknowledgement :

i). In Selective reject, If the sender does not receive any ACK or the ACK is lost or damaged in between the transmission.

ii). The sender waits for the time to run out and as the time run outs, the sender retransmit all the frames for which it has not received the ACK.

iii). The sender identifies the loss of ACK with the help of a timer.

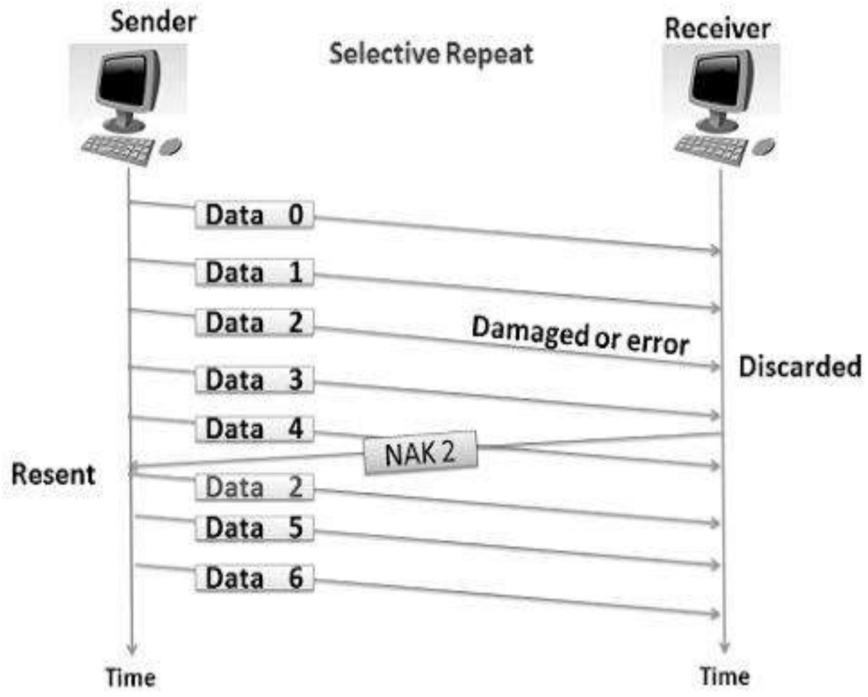


Figure 5.6. Selective Repeat ARQ

d). <u>Selective Repeat ARQ Algorithm</u>

N = window size
Sn = sequence number
Sb = sequence base
Sm = sequence max
ack = ack number
nack = first non acknowledged
Receiver:
Do the following forever:
Randomly accept or reject packet

If the packet received and the packet is error free

```
        Accept packet
        Send a positive ack for packet
  Else
        Refuse packet
        Send a negative ack for packet

Sender:
Sb = 0
Sm = N − 1
ack = 0
Repeat the following steps forever:
          If the packet was not already positively acknowledged by receiver
       Send packet with ack
       If positively ack is recieved:
           Transmit a packet where Sb<= ack<= Sm.
           packets are transmitted in order
       Else
Enqueue the nack into the queue
ack++
  //check if last packet in the window is sent
  if(ack==Sm)
    if(queue is not empty)
       // start from the first nack packet
nack = queue.front();
       empty the queue
ack = nack

Sm = Sm + (ack − Sb)
Sb = ack
```

5.5 CHECK YOUR PROGRESS QUESTIONS

1. Define Piggybacking.
2. Mention about Multiple Access Control.

5.6 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. **Piggybacking:** In bidirectional communications, both parties send & acknowledge data, i.e. both parties implement flow control. Outstanding ACKs are placed in the header of information frames, piggybacking can

save bandwidth since the overhead from a data frame and an ACK frame can be combined into just one frame.

2. **Multiple Access Control:** If there is a dedicated link between the sender and the receiver then data link control layer is sufficient, however if there is no dedicated link present then multiple stations can access the channel simultaneously.

5.7 SUMMARY

The sliding window mechanism is widely used to integrate error control and flow control in a convenient way. Sliding window protocols can be categorized by the size of the sender's window and the size of the receiver's window. When both are equal to 1, the protocol is stop-and-wait. When the sender's window is greater than 1, for example, to prevent the sender from blocking on a circuit with a long propagation delay, the receiver can be programmed either to discard all frames other than the next one in sequence or to buffer out-of-order frames until they are needed.

5.8 KEY WORDS

- **Data Link Layer**: Data Link Layer is utilized to transmit data between two computers.
- **Data Link control:** The data link control is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control.

5.9 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Mention about the demerit of Go back-n protocol.
2. Differentiate ACK and NAK.

Long Answer Questions

1. Discuss about simple stop-and –wait protocol.
2. Illustrate about Selective Repeat ARQ Algorithm.
3. An upper layer packet is split in to 10 frames, each of which has an 80 percent chance of arriving undamaged. If no error control is done by the data link protocol, how many times must the message be sent on average to get the entire thing through?

5.10 FURTHER READINGS

1. Peterson and Davie, Computer Networks: A Systems Approach.
2. Behrouz A. Forouzan, Data Communications and Networking, Fourth Edition, TataMcGraw Hill, 2009.
3. Kleinrock, "On Some Principles of Nomadic Computing and Multi-Access Communications".

UNIT-6. MULTIPLE ACCESS PROTOCOLS

Structure

6.0 INTRODUCTION

In this unit, you will learn about how to allocate the channel while broadcast communication is needed. When only one channel is available, it is very tough to identify who is going first. The multiple access protocols such as Pure ALOHA, Slotted ALOHA, Carrier Sense Multiple Access (CSMA), Carrier Sense Multiple Access with Collision Detection (CSMA/CD), and Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) are playing significant role in solving the channel allocation problem.

6.1 OBJECTIVES

After going through this unit, you will describe about

- Multiple Access Protocols
- Control Access Protocols

NOTES

6.2 INTRODUCTION TO MULTIPLE ACCESS PROTOCOLS

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

i). Data Link Control
ii). Multiple Access Control

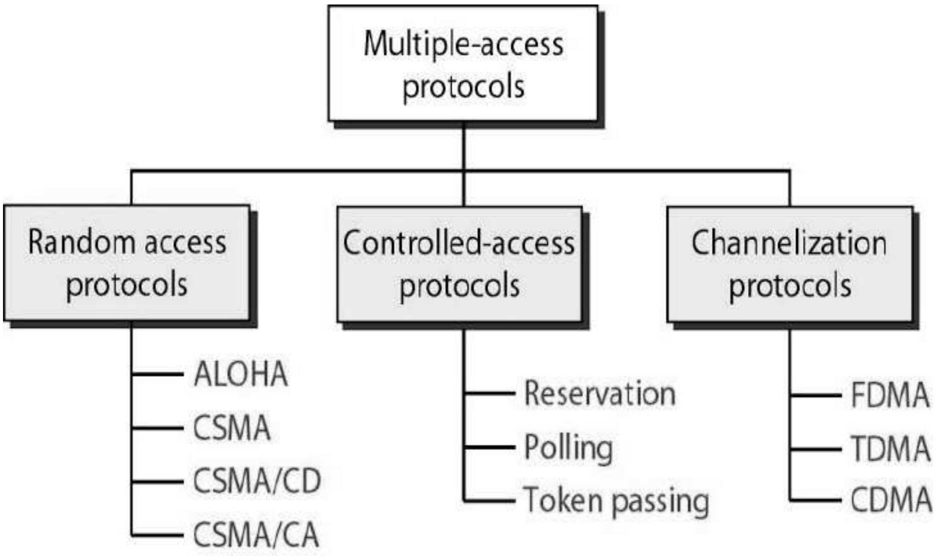


Figure 6. 1. Multiple Access Protocols

6.3 ALOHA

In the 1970s, Norman Abramson and his colleagues at the University of Hawaii devised a new and elegant method to solve the channel allocation problem. Their work has been extended by many researchers since then (Abramson, 1985). Although Abramson's work, called the ALOHA system, used ground-based radio broadcasting, the basic idea is applicable to any system in which uncoordinated users are competing for the use of a single shared channel. We will discuss two versions of ALOHA here: pure and slotted. They differ with respect to whether time is divided into discrete slots into which all frames must fit. Pure ALOHA does not require global time synchronization; slotted ALOHA does.

6.3.1 PURE ALOHA

The basic idea of an ALOHA system is simple: let users transmit frames will be damaged. However, due to the feedback property of broadcasting, a sender can always find out whether its frame was destroyed by listening to the channel, the same way other users do. With a LAN, the feedback is immediate; with a satellite, there is a delay of 270 msec before the sender knows if the transmission was successful. If listening while transmitting is not possible for some reason, acknowledgements are needed. If the frame was destroyed, the sender just waits a random amount of time and sends it again. The waiting time must be random or the same frames will collide over and over, in lockstep. Systems in which multiple users share a common channel in a way that can lead to conflicts are widely known as **contention** systems. A sketch of frame generation in an ALOHA system is given in Fig. 6.2. We have made the frames all the same length because the throughput of ALOHA systems is maximized by having a uniform frame size rather than by allowing variable length frames.

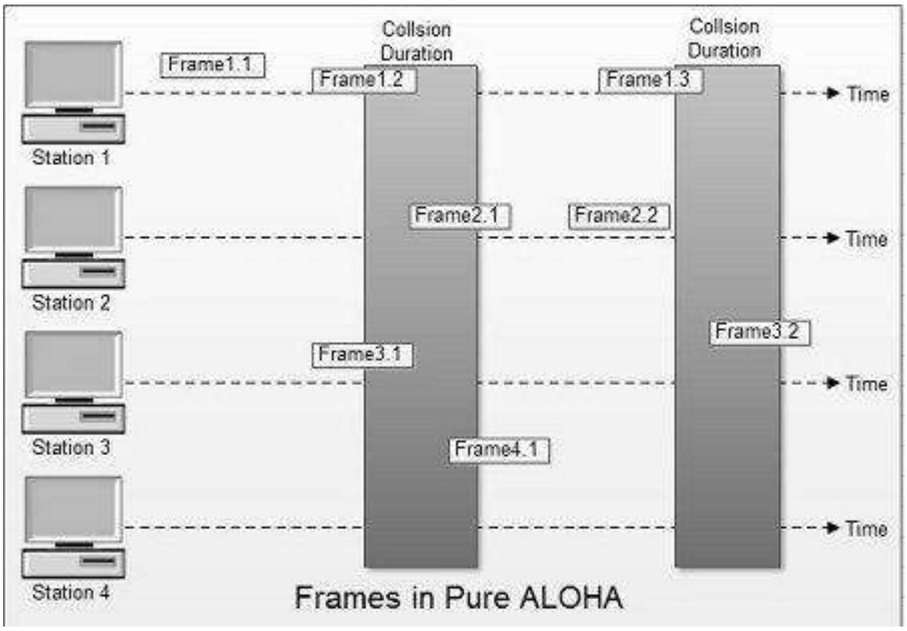


Figure 6.2. In pure ALOHA, frames are transmitted at Completely Arbitrary Times.

Whenever two frames try to occupy the channel at the same time, there will be a collision and both will be garbled. If the first bit of a new frame overlaps with just the last bit of a frame almost finished, both frames will be totally destroyed and both will have to be retransmitted later. The checksum cannot (and should not) distinguish between a total loss and a near miss. Bad is bad.

An interesting question is: What is the efficiency of an ALOHA channel? In other words, what fractions of all transmitted frames escape collisions under these chaotic circumstances? Let us first consider an infinite collection of interactive users sitting at their computers (stations). A user is always in one of two states: typing or waiting. Initially, all users are in the typing state. When a line is finished, the user stops typing, waiting for a response. The station then transmits a frame containing the line and checks the channel to see if it was successful. If so, the user sees the reply and goes back to typing. If not, the user continues to wait and the frame is retransmitted over and over until it has been successfully sent.

Let the "frame time" denote the amount of time needed to transmit the standard, fixed-length frame (i.e., the frame length divided by the bit rate). At this point we assume that the infinite population of users generates new frames according to a Poisson distribution with mean *N* frames per frame time. (The infinite-population assumption is needed to ensure that *N* does not decrease as users become blocked.) If $N > 1$, the user community is generating frames at a higher rate than the channel can handle, and nearly every frame will suffer a collision. For reasonable throughput we would expect $0 < N < 1$.

In addition to the new frames, the stations also generate retransmissions of frames that previously suffered collisions. Let us further assume that the probability of *k* transmission attempts per frame time, old and new combined, is also Poisson, with mean *G* per frame time. Clearly, G_N.At low load (i.e., N_0), there will be few collisions, hence few retransmissions, so G_N. At high load there will be many collisions, so $G > N$. Under all loads, the throughput, *S*, is just the offered load, *G*, times the probability, P_0, of a transmission succeeding—that is, $S = GP_0$, where P_0 is the probability that a frame does not suffer a collision.

A frame will not suffer a collision if no other frames are sent within one frame time of its start, as shown in Fig.6.3. Under what conditions will the shaded frame arrive undamaged? Let *t*be the time required to send a frame. If any other user has generated a frame between time t_0 and $t_0 + t$, the end of that frame will collide with the beginning of the shaded one. In fact, the shaded frame's fate was already sealed even before the first bit was sent, but since in pure ALOHA a station does not listen to the channel before transmitting, it has no way of knowing that another frame was already underway. Similarly, any other frame started between t_0+*t*and t_0 + 2twill bump into the end of the shaded frame.

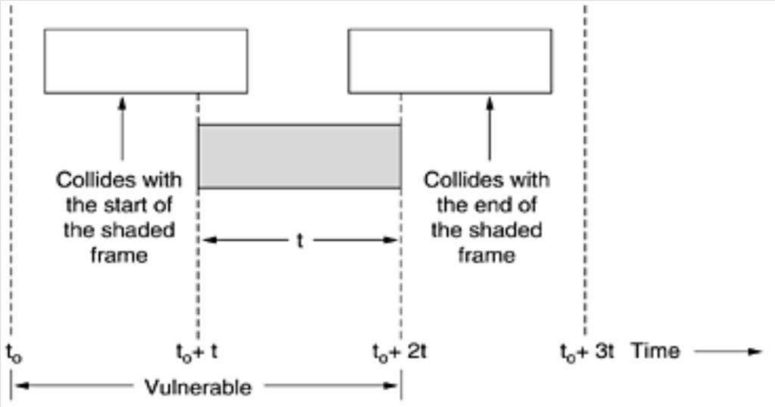

Collides with the start of the shaded frame
Collides with the end of the shaded frame
t
t₀
t₀+ t
t₀+ 2t
t₀+ 3t
Time
Vulnerable

Figure 6.3. Vulnerable period for the Shaded Frame

$$\Pr[k] = \frac{G^k e^{-G}}{k!} \qquad (6.1)$$

So the probability of zero frames is just *e*-*G*. In an interval two frame times long, the mean number of frames generated is 2*G*. The probability of no other traffic being initiated during the entire vulnerable period is thus given by *P*0 = *e* -2*G*. Using *S* = *GP*0, we get

$$S = Ge^{-2G} \qquad (6.2)$$

The relation between the offered traffic and the throughput is shown in Fig. 6-3. The maximum throughput occurs at *G* = 0.5, with *S* = 1/2*e*, which is about 0.184. In other words, the best we can hope for is a channel utilization of 18 percent. This result is not very encouraging, but with everyone transmitting at will, we could hardly have expected a 100 percent success rate.

6.3.2 SLOTTED ALOHA

In 1972, Roberts published a method for doubling the capacity of an ALOHA system. His proposal was to divide time into discrete intervals, each interval corresponding to one frame. This approach requires the users to agree on slot boundaries. One way to achieve synchronization would be to have one special station emit a pip at the start of each interval, like a clock.

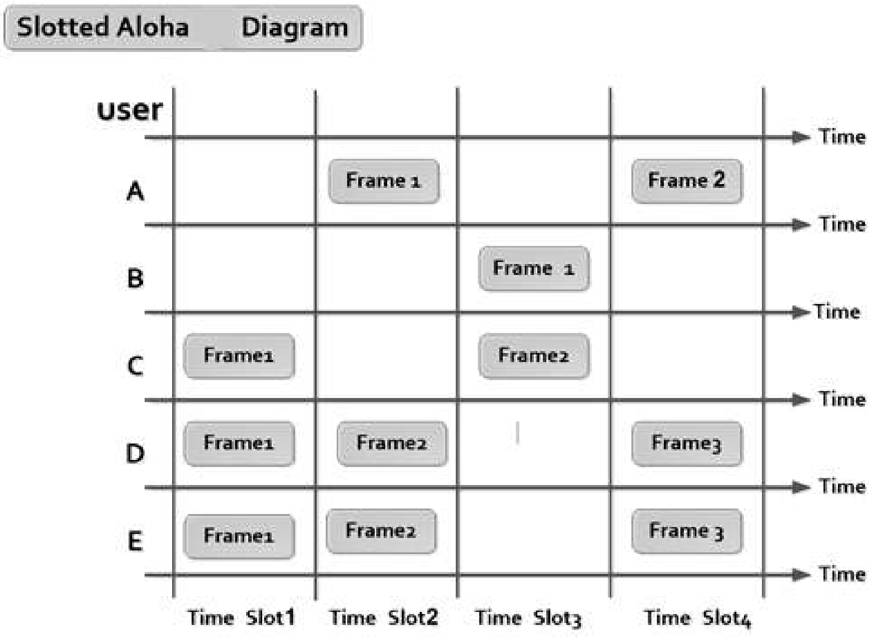


Figure 6.4 . Slotted ALOHA

In Roberts' method, which has come to be known as slotted ALOHA, in contrast to Abramson's pure ALOHA, a computer is not permitted to send whenever a carriage return is typed. Instead, it is required to wait for the beginning of the next slot. Thus, the continuous pure ALOHA is turned into a discrete one. Since the vulnerable period is now halved, the probability of no other traffic during the same slot as our test frame is e-G which leads to equation.

$$S = Ge^{-G} \qquad (6.3)$$

As you can see from Fig. 4-3, slotted ALOHA peaks at *G* = 1, with a throughput of *S* =1/*e* or about 0.368, twice that of pure ALOHA. If the system is operating at *G* = 1, the probability of an empty slot is 0.368. The best we can hope for using slotted ALOHA is 37 percent of the slots empty, 37 percent successes, and 26 percent collisions. Operating at higher values of *G* reduces the number of empties but increases the number of collisions exponentially. To see how this rapid growth of collisions with *G* comes about, consider the transmission of a test frame. The probability that it will avoid a collision is *e-G*, the probability that all the other users are silent in that slot. The probability of a collision is then just 1 - *e-G*. The probability of a transmission requiring exactly *k* attempts, (i.e., *k* - 1 collisions followed by one success) is

$$\boldsymbol{P_k} = e^{-G} * (1 - e^{-G})^{k-1} \qquad (6.4)$$

The expected number of transmissions, E, per carriage return typed is then

$$\mathrm{E} = \sum_{\mathrm{k}=1}^{\infty} \mathrm{k}\,\boldsymbol{P_k} = \sum_{\mathrm{k}=1}^{\infty} \mathrm{ke}^{-\mathrm{G}}\ (1 - e^{-G})^{\mathrm{k}-1} = e^{G} \qquad (6.5)$$

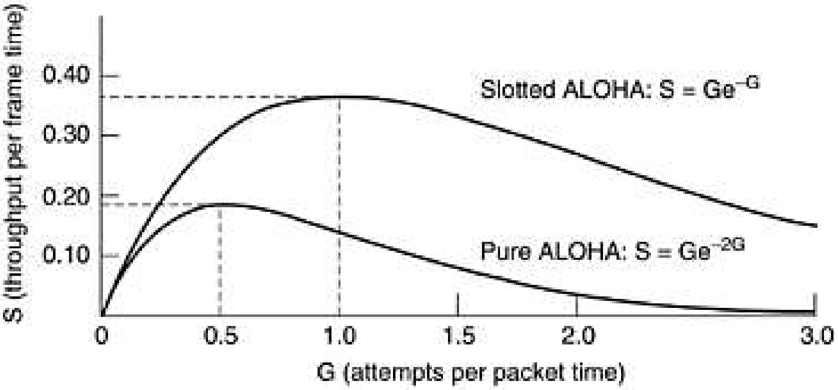


Figure 6.5. Throughput Versus Offered Traffic for ALOHA

As a result of the exponential dependence of E upon G, small increases in the channel load can drastically reduce its performance. Slotted Aloha is important for a reason that may not be initially obvious. It was devised in the 1970s, used in a few early experimental systems, then almost forgotten. When Internet access over the cable was invented, all of a sudden there was a problem of how to allocate a shared channel among multiple competing users, and slotted Aloha was pulled out of the garbage can to save the day. It has often happened that protocols that are perfectly valid fall into disuse for political reasons (e.g., some big company wants everyone to do things its way), but years later some clever person realizes that a long-discarded protocol solves his current problem.

6.4 CARRIER SENSE MULTIPLE ACCESS (CSMA)

The Carrier Sense Multiple Access (CSMA) ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data. If it is idle then it sends data, otherwise it waits till the channel becomes idle. However there is still chance of collision in CSMA due to propagation delay. For example, if station A wants to send data, it will first sense the medium. If it finds the channel idle, it will start sending data. However, by the time the first bit of data is transmitted (delayed due to propagation delay) from station A, if station B requests to send data and senses the medium it will also find it idle and will also send data. This will result in collision of data from station A and B.

The categories of access mode of CSMA are described in the following sections:

a). I-Persistent CSMA

The first carrier sense protocol that we will study here is called 1-persistent CSMA. The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle. When a station has data to send, it first listens to the channel to see if anyone else is transmitting at that moment. If the channel is busy, the station waits until it becomes idle. When the station detects an idle channel, it transmits a frame. If a collision occurs, the station waits a random amount of time and starts all over again. The protocol is called **1-persistent** because the station transmits with a probability of 1 when it finds the channel idle.

b). Non-Persistent CSMA

A second carrier sense protocol is **non-persistent CSMA**. The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle. In this protocol, a conscious attempt is made to be less greedy than in the previous one. Before sending, a station senses the channel. If no one else is sending, the station begins doing so itself. However, if the channel is already in use, the station does not continually sense it for the purpose of seizing it immediately upon detecting the end of the previous transmission. Instead, it waits a random period of time and then repeats the algorithm. Consequently, this algorithm leads to better channel utilization but longer delays than 1-persistent CSMA.

c). P-Persistent CSMA

In **p-persistent CSMA, t**he node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems. It applies to slotted channels and works as follows. When a station becomes ready to send, it senses the channel. If it is idle, it transmits with a probability p. With a probability q = 1 - p, it defers until the next slot. If that slot is also idle, it either transmits or defers again, with probabilities p and q. This process is repeated until either the frame has been transmitted or another station has begun transmitting. In the latter case, the unlucky station acts as if there had been a collision (i.e., it waits a random time and starts again). If the station initially senses the channel busy, it waits until the next slot and applies the above algorithm. Figure 6.6 shows the computed throughput versus offered traffic for all three protocols, as well as for pure and slotted ALOHA.

d). **O-persistent:** Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

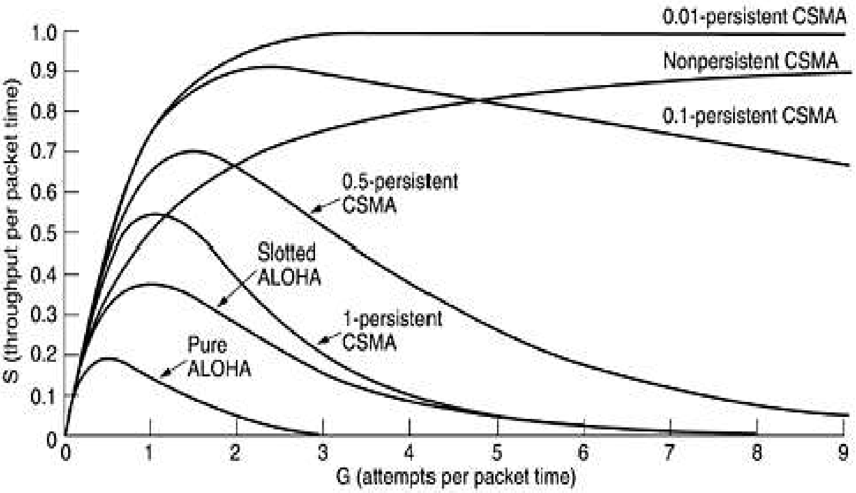


Figure 6.6. Comparison of the Channel Utilization Versus Load for Various Random Access Protocols

6.5 CARRIER SENSE MULTIPLE ACCESS WITH COLLISION DETECTION (CSMA/CD)

In CSMA/CD method, a station monitors the medium after it sends a frame to see if the transmission was successful. If so, the station is finished. If, however, there is a collision, the frame is sent again. Persistent and non-persistent CSMA protocols are clearly an improvement over ALOHA because they ensure that no station begins to transmit when it senses the channel busy. Another improvement is for stations to abort their transmissions as soon as they detect a collision. In other words, if two stations sense the channel to be idle and begin transmitting simultaneously, they will both detect the collision almost immediately. Rather than finish transmitting their frames, which are irretrievably garbled anyway, they should abruptly stop transmitting as soon as the collision is detected. Quickly terminating damaged frames saves time and bandwidth. This protocol, known as CSMA/CD (CSMA with Collision Detection) is widely used on LANs in the MAC sub layer. In particular, it is the basis of the popular Ethernet LAN, so it is worth devoting some time to looking at it in detail.

CSMA/CD, as well as many other LAN protocols, uses the conceptual model of Fig.6.7. At the point marked t0, a station has finished transmitting its frame. Any other station having a frame to send may now attempt to do so. If two or more stations decide to transmit simultaneously, there will be a collision. Collisions can be detected by looking at the power or pulse width of the received signal and comparing it to the transmitted signal.

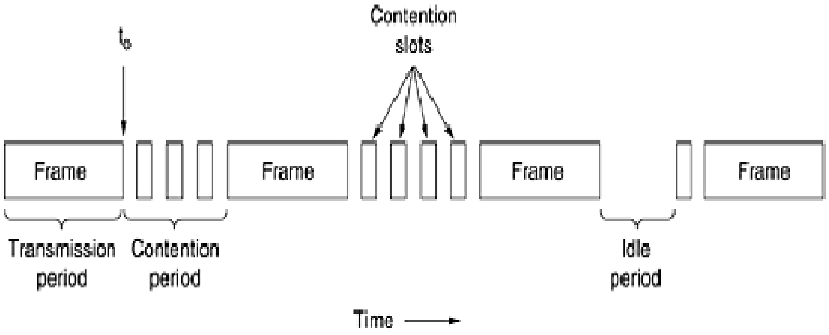


Figure 6.7. CSMA/CD can be in one of three states: Contention, Transmission and Idle.

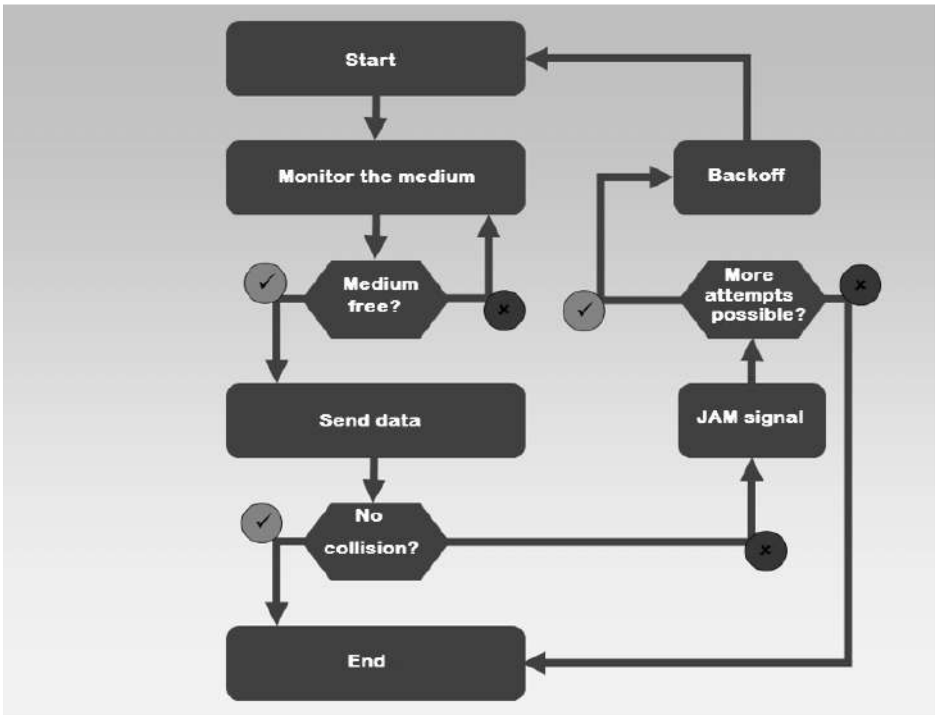


Figure 6.8. CSMA / CD Flowchart

After a station detects a collision, it aborts its transmission, waits a random period of time, and then tries again, assuming that no other station has started transmitting in the meantime. Therefore, our model for CSMA/CD will consist of alternating contention and transmission periods, with idle periods occurring when all stations are quiet (e.g., for lack of work).

6.6 CARRIER SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE (CSMA/CA)

The fundamental idea behind *CSMA/CD* is that a station needs to be able to receive while transmitting to detect a collision. When there is no collision, the station receives one signal: its own signal. When there is a collision, the station receives two signals: its own signal and the signal transmitted by a second station. In order to handle collision problem, 802.11 supports two modes of operation.

a). Distributed Coordination Function (DCF): The DCF does not utilize any kind of central control (in that respect, similar to Ethernet).

b). Point Coordination Function (PCF): The PCF utilize the base station to control all activity in its cell. All implementations must support DCF but PCF is optional.

The 802.11 employs CSMA/CA protocol while DCF is utilized, In this protocol, both physical channel sensing and virtual channel sensing are used. Two methods of operation are supported by CSMA/CA. In the first method, when a station wants to transmit, it senses the channel. If it is idle, it just starts transmitting. It does not sense the channel while transmitting but emits its entire frame, which may well be destroyed at the receiver due to interference there. If the channel is busy, the sender defers until it goes idle and then starts transmitting. If a collision occurs, the colliding stations wait a random time, using the Ethernet binary exponential back off algorithm, and then try again later. The other mode of CSMA/CA operation is based on MACAW and uses virtual channel sensing, as illustrated in Fig. 6.9. In this example, A wants to send to B. C is a station within range of A. Also D is a station within range of B but not within range of A

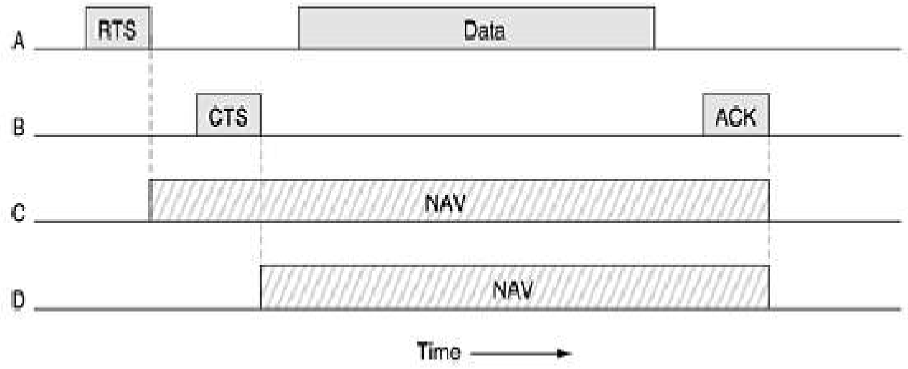


Figure 6.9. The Use of Virtual Channel Sensing Using CSMA/CA.

➢ CONTROLLED ACCESS

In controlled access, the stations consult one another to find which station has the right to send.

a). **Reservation:** In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.

b). **Polling:** Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be madethrough the primary device even when the ultimate destination is a secondary device.The primary device controls the link; the secondary devices follow its instructions. Itis up to the primary device to determine which device is allowed to use the channel ata given time. The primary device, therefore, is always the initiator of a session.

c). **Token Passing:** In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a *predecessor* and a *successor*. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now.

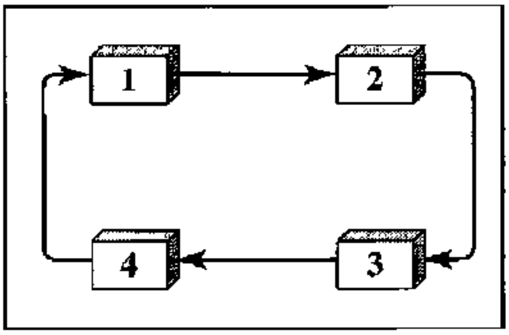


Figure 6.10. Token-Passing Method

6.7 CHECK YOUR PROGRESS QUESTIONS

1. Define Throughput of Channel.
2. What is Token Passing?

6.8 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Throughput of channel is defined as number of data frame transferred per unit time via that channel.
2. In token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a *predecessor* and a *successor.*

6.9 SUMMARY

Some networks have a single channel that is used for all communication. In these networks, the key design issue is the allocation of this channel among the competing stations wishing to use it. Numerous channel allocation algorithms have been devised. When the number of stations is large and variable or the traffic is fairly rupture. The FDM and TDM are poor choices. The ALOHA protocol, with and without slotting, has been proposed as an alternative. ALOHA and its many variants and derivatives have been widely discussed, analysed, and used in real systems. When the state of the channel can be sensed, stations can avoid starting a transmission while another station is transmitting. This technique, carrier sensing, has led to a variety of protocols that can be used on LANs and MANs.

6.10 KEY WORDS

- The **Data Link Control** is responsible for reliable transmission of message over transmission channel by using techniques like framing, error control and flow control.
- **Controlled Access** means the stations consult one another to find which station has the right to send.
- **Polling** works with topologies in which one device is designated as a primary station and the other devices are secondary stations.

6.11 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1) Differentiate Simple ALOHA and Slotted ALOHA.
2) What is meant by Polling?

Long Answer Questions

1. Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.
2. Ten thousand airline reservation stations are competing for the use of a single slotted
3. ALOHA channel. The average station makes 18 requests/hour. A slot is 125 μsec. What is the approximate total channel load?
4. A large population of ALOHA users manages to generate 50 requests/sec, including both originals and retransmissions. Time is slotted in units of 40 msec.

(a) What is the chance of success on the first attempt?
(b) What is the probability of exactly k collisions and then a success?
(c) What is the expected number of transmission attempts needed?

5. Measurements of a slotted ALOHA channel with an infinite number of users show that 10 percent of the slots are idle.

(a) What is the channel load G?

(b) What is the throughput?

(c) Is the channel under loaded or overloaded?

6.12 FURTHER READINGS

1. Azzam and Ransom, Broadband Access Technologies.
2. Bellamy, Digital Telephony.
3. Farserotu and Prasad, "A Survey of Future Broadband Multimedia Satellite Systems, Issues and Trends".
4. Kleinrock, "On Some Principles of Nomadic Computing and Multi-Access Communications".

BLOCK3: NETWORK LAYER

UNIT-7- SWITCHING TECHNIQUES

Structure

7.0 INTRODUCTION

In this unit, you will learn about a variety of transmission media available for telecommunication. The network layer is concerned with getting packets from the source all the way to the destination. Getting to the destination may require making many hops at intermediate routers along the way. In between the switching offices, coaxial cables, microwaves, and especially fiber optics are widely utilized. This function clearly contrasts with that of the data link layer, which has the more modest goal of just moving frames from one end of a wire to the other. Thus, the network layer is the lowest layer that deals with end-to-end transmission. To achieve its goals, the network layer must know about the topology of the communication subnet (i.e., the set of all routers) and choose appropriate paths through it. It must also take care to choose routes to avoid overloading some of the communication lines and routers while leaving others idle.

7.1 OBJECTIVES

After going through this unit you will be able to understand about

- Process of Network Layer and Switching Circuits
- Circuit Switching

- Packet Switching
- Message Switching
- Virtual Circuit and Datagram Subnets

7.2 INTRODUCTION TO NETWORK LAYER

The network layer has to recognize the topology of the communication subnet (set of all routers) and select appropriate paths through it. The network layer provides services to the transport layer at the network layer or transport layer interface. The network Layer controls the operation of the subnet. The main aim of this layer is to deliver packets from source to destination across multiple links (networks). If two computers or systems are connected on the same link, then there is no need for a network layer. It routes the signal through different channels to the other end and acts as a network controller. It also divides the outgoing messages into packets and to assemble incoming packets into messages for higher levels. The network layer services have been designed with the following goals in mind.

i). The services should be independent of the router technology.

ii). The transport layer should be shielded from the number, type, and topology of the routers present.

iii). The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

7.2.1 Design Issues with Network Layer

A key design issue is **determining how packets are routed from source to destination**. Routes can be based on static tables that are wired into the network and rarely changed. They can also be highly dynamic, being determined anew for each packet, to reflect the current network load. If **too many packets** are present in the subnet at the same time, they will get into one another's way, forming **bottlenecks**. The **control of such congestion** also belongs to the network layer. Moreover, the **quality of service** provided (delay, transmit time, jitter, etc) is also a network layer issue.

When a packet has to travel from one network to another to get to its destination, many problems can arise such as:

i). The addressing used by the second network may be different from the first one.

ii). The second one may not accept the packet at all because it is too large. The protocols may differ, and so on. It is up to the network layer to overcome all these problems to allow heterogeneous networks to be interconnected.

7.2.2 Switching Techniques

The telephone system is divided into two principal parts: **outside plant** (the local loops and trunks, since they are physically outside the switching offices) and **inside plant** (the switches), which are inside the switching offices. Here consider the outside plant. Now it is time to examine the inside plant. Switched communication networks are those in which data transferred from source to destination is routed between various intermediate nodes. Switching is the technique by which nodes control or switch data to transmit it between specific points on a network. The figure 7.1 describes concept of switching techniques.

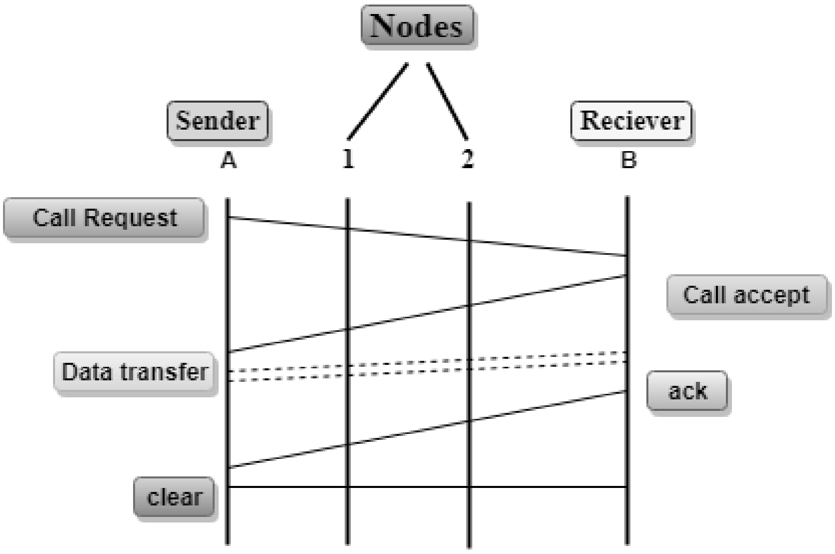


Figure 7.1.Concept of Switching Techniques

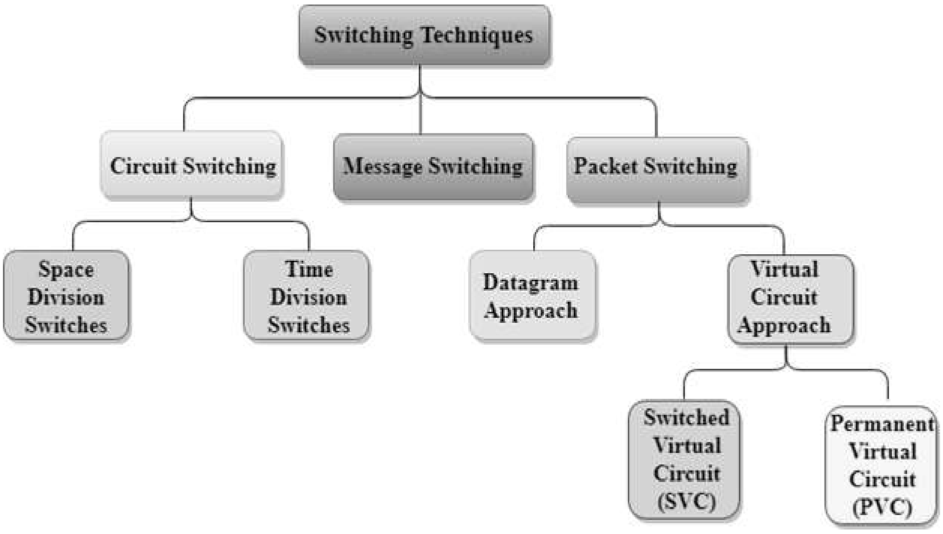


Figure 7.2. Classification of Switching Techniques

The Classification of switching techniques is offered in figure 7.2.

There are 3 common switching techniques:

i). Circuit Switching
ii). Packet Switching
iii). Message Switching

NOTES

7.3. CIRCUIT SWITCHING

The information about circuit switching is given in detailed manner below because that is how the telephone system works. When you or your computer places a telephone call, the switching equipment within the telephone system seeks out a physical path all the way from your telephone to the receiver's telephone. This technique is called circuit switching and is shown schematically in Fig.7.3 (a). Each of the six rectangles represents a carrier switching office (end office, toll office, etc.). In this example, each office has three incoming lines and three outgoing lines. When a call passes through a switching office, a physical connection is (conceptually) established between the line on which the call came in and one of the output lines, as shown by the dotted lines.

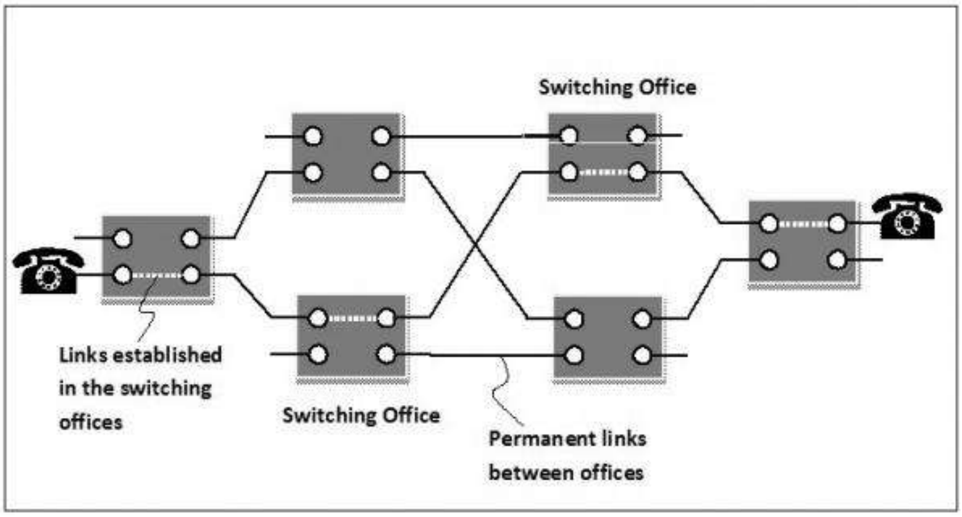

Switching Office
Links established in the switching offices
Switching Office
Permanent links between offices

Figure 7.3.(a) Circuit Switching.

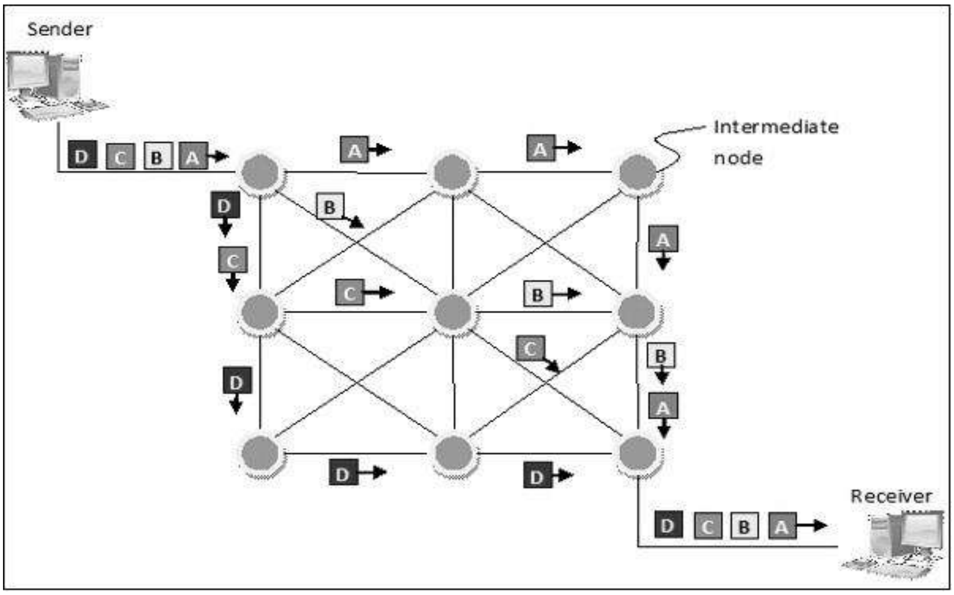


Figure 7.3. (b) Packet Switching.

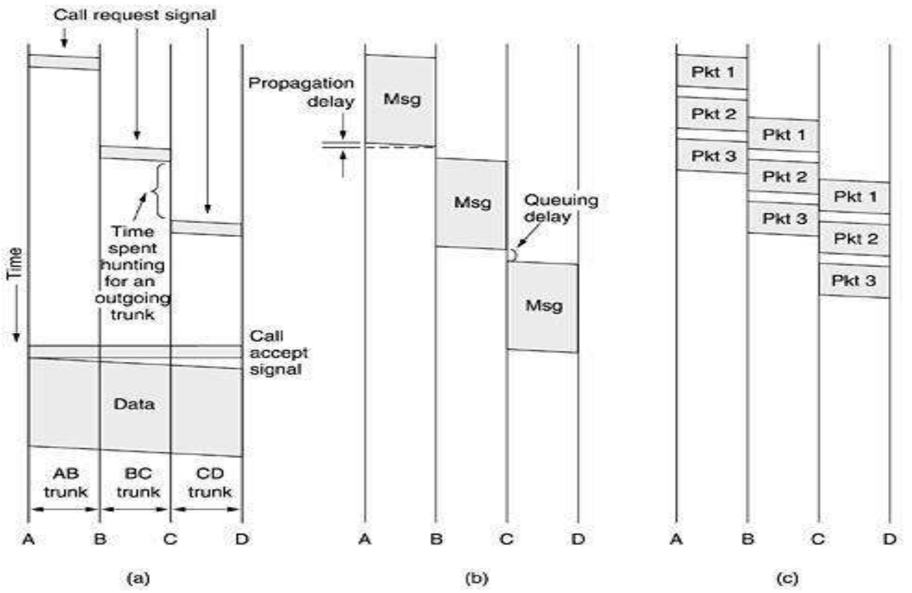


Figure 7.4. Timing of Events in (a) Circuit Switching, (b) Message Switching, (c) Packet Switching

a). Advantages of Circuit Switching:

i). In the case of Circuit Switching technique, the communication channel is dedicated.

ii). It has fixed bandwidth.

b). Disadvantages of Circuit Switching:

i). Once the dedicated path is established, the only delay occurs in the speed of data transmission.

ii). It takes a long time to establish a connection approx 10 seconds during which no data can be transmitted.

iii). It is more expensive than other switching techniques as a dedicated path is required for each connection.

iv). It is inefficient to use because once the path is established and no data is transferred, then the capacity of the path is wasted.

v). In this case, the connection is dedicated therefore no other data can be transferred even if the channel is free.

7.4. PACKET SWITCHING

The packet-switching networks place a tight upper limit on block size, allowing packets to be buffered in router main memory instead of on disk. By making sure that no user can monopolize any transmission line very long (milliseconds), packet-switching networks are well suited for handling interactive traffic. A further advantage of packet switching over message switching is shown in Fig. 7.4.(b) and (c): the first packet of a multi packet message can be forwarded before the second one has fully arrived, reducing delay and improving throughput. For these reasons, computer networks are usually packet switched, occasionally circuit

NOTES

switched, but never message switched. The process of store and forward packet switching is presented in figure 7.5.

+

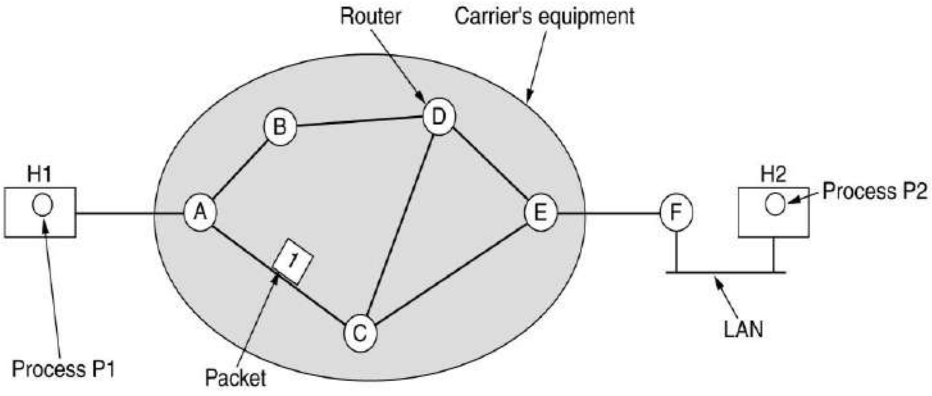


Figure 7.5 Store–and- Forward Packet Switching

Packet switching is more fault tolerant than circuit switching. In fact, that is why it was invented. If a switch goes down, all of the circuits using it are terminated and no more traffic can be sent on any of them. With packet switching, packets can be routed around dead switches. Setting up a path in advance also opens up the possibility of reserving bandwidth in advance. If bandwidth is reserved, then when a packet arrives, it can be sent out immediately over the reserved bandwidth. With packet switching, no bandwidth is reserved, so packets may have to wait their turn to be forwarded. Having bandwidth reserved in advance means that no congestion can occur when a packet shows up (unless more packets show up than expected). On the other hand, when an attempt is made to establish a circuit, the attempt can fail due to congestion. Thus, congestion can occur at different times with circuit switching (at setup time) and packet switching (when packets are sent).

a). Advantages of Packet Switching:

i). **Cost-effective:** In packet switching technique, switching devices do not require massive secondary storage to store the packets, so cost is minimized to some extent. Therefore, we can say that the packet switching technique is a cost-effective technique.

ii). **Reliable:** If any node is busy, then the packets can be rerouted. This ensures that the Packet Switching technique provides reliable communication.

iii). **Efficient:** Packet Switching is an efficient technique. It does not require any established path prior to the transmission, and many users can use the same communication channel simultaneously, hence makes use of available bandwidth very efficiently.

b). Disadvantages of Packet Switching:

i). Packet Switching technique cannot be implemented in those applications that require low delay and high-quality services.

ii). The protocols used in a packet switching technique are very complex and requires high implementation cost.

iii). If the network is overloaded or corrupted, then it requires retransmission of lost packets. It can also lead to the loss of critical information if errors are nor recovered.

Table 7.1. A comparison of Circuit-Switched and Packet- Switched Networks.

BASIS FOR COMPARISON	CIRCUIT SWITCHING	PACKET SWITCHING
Orientation	Connection oriented.	Connectionless.
Purpose	Initially designed for Voice communication.	Initially designed for Data Transmission.
Flexibility	Inflexible, because once a path is set all parts of a transmission follows the same path.	Flexible, because a route is created for each packet to travel to the destination.
Order	Message is received in the order, sent from the source.	Packets of a message are received out of order and assembled at the destination.
Technology/ Approach	Circuit switching can be achieved using two technologies, either Space Division Switching or Time-Division Switching.	Packet Switching has two approaches Datagram Approach and Virtual Circuit Approach.
Layers	Circuit Switching is implemented at Physical Layer.	Packet Switching is implemented at Network Layer.

7.5. MESSAGE SWITCHING

The message switching is an alternative switching strategy in illustrated in Fig.7.6 and timing events of message switching is shown in 7.4(b). When this form of switching is used, no physical path is established in advance between sender and receiver. Instead, when the sender has a block of data to be sent, it is stored in the first switching office (i.e., router) and then forwarded later, one hop at a time. Each block is received in its entirety, inspected for errors, and then retransmitted. A network using this technique is called a **store-and-forward network**. With message switching, there is no limit at all on block size, which means that the routers must have disks to buffer long blocks. It also means that a single block can tie up a router-router line for minutes, rendering message switching useless for interactive traffic. To get around these problems, packet switching was invented.

Message switching was a technique developed as an alternate to circuit switching, before packet switching was introduced. In message switching, end users communicate by sending and receiving *messages* that included the entire data to be shared. Messages are the smallest individual unit. Furthermore, the sender and receiver are not directly connected. There are a number of intermediate nodes transfer data and ensure that the message reaches its destination. Message switched data networks are hence called **hop-by-hop systems**.

They provide 2 distinct and important characteristics:

i). **Store and forward –** The intermediate nodes have the responsibility of transferring the entire message to the next node. Hence, each node must have storage capacity. A message will only be delivered if the next hop and the link connecting it are both available, otherwise it'll be stored indefinitely. A store-and-forward switch forwards a message only if sufficient resources are available and the next hop is accepting data. This is called the store-and-forward property.

ii). **Message delivery –** This implies wrapping the entire information in a single message and transferring it from the source to the destination node. Each message must have a header that contains the message routing information, including the source and destination.

Message switching network consists of transmission links (channels), store-and-forward switch nodes and end stations as shown in the figure 7.6.

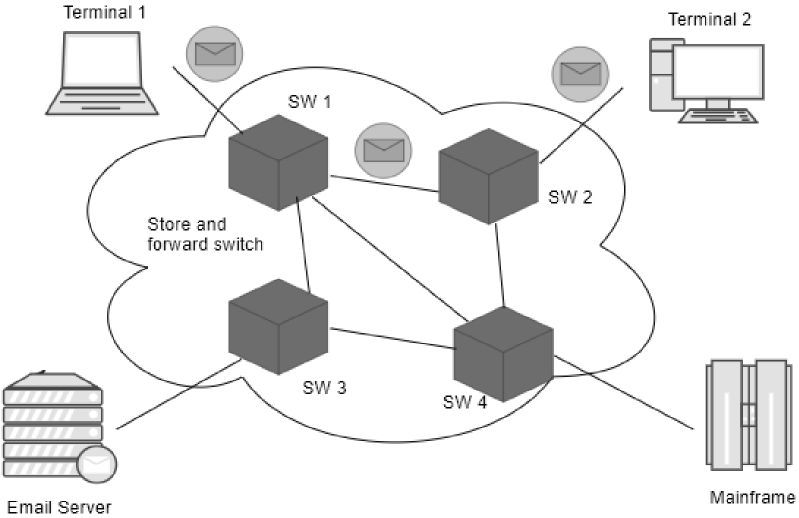


Figure 7.6. Message Switching

a). Characteristics of message switching:

Message switching is advantageous as it enables efficient usage of network resources. Also, because of the store-and-forward capability of intermediary nodes, traffic can be efficiently regulated and controlled. Message delivery as one unit, rather than in pieces, is another benefit. However, message switching has certain disadvantages as well. Since messages are stored indefinitely at each intermediate node, switches require large storage capacity. Also, these are pretty slow. This is because at each node, first there us wait till the entire message is received, then it must be stored and transmitted after processing the next node and links to it depending on availability and channel traffic. Hence, message switching cannot be used for real time or interactive applications like video conference.

b). Applications of message switching:

The store-and-forward method was implemented in telegraph message switching centres. Today, although many major networks and systems are packet-switched or circuit switched networks, their delivery processes can be based on message switching. For example, in most electronic mail systems the delivery process is based on message switching, while the network is in fact either circuit-switched or packet-switched.

c). Advantages of Message Switching

i). Data channels are shared among the communicating devices that improve the efficiency of using available bandwidth.

ii). Traffic congestion can be reduced because the message is temporarily stored in the nodes.

iii). Message priority can be used to manage the network.

iv). The size of the message which is sent over the network can be varied. Therefore, it supports the data of unlimited size.

d). **Disadvantages of Message Switching**

i). The message switches must be equipped with sufficient storage to enable them to store the messages until the message is forwarded.

ii). The Long delay can occur due to the storing and forwarding facility provided by the message switching technique.

7.6. VIRTUAL CIRCUIT AND DATAGRAM SUBNETS

The network layer provides services to the transport layer at the network layer or transport layer interface. Here an important perception is what kind of services the network layer provides to the transport layer. The network layer services have been designed with the following goals in mind.

i). The services should be independent of the router technology.

ii). The transport layer should be shielded from the number, type, and

NOTES

topology of the routers present.

iii). The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

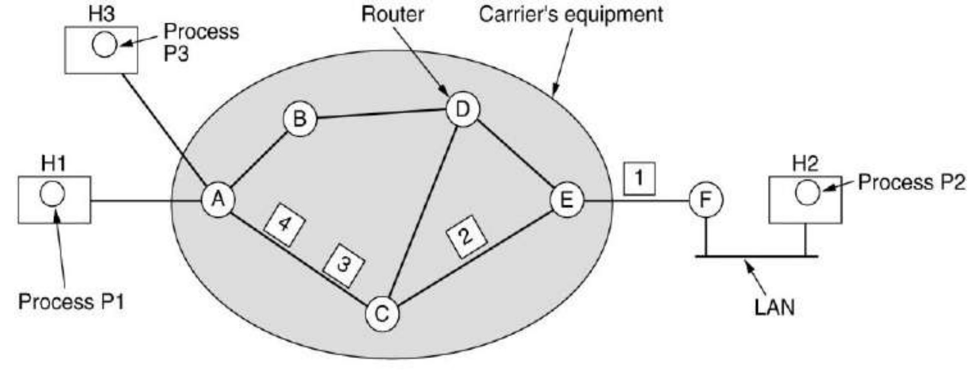
The layer services are classified in to two. They are

a). Connection-Oriented Service

b). Connectionless Service

a). Implementation of Connection-Oriented Service

If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a **Virtual Circuit (VC**), in analogy with the physical circuits set up by the telephone system, and the subnet is called a **Virtual-Circuit Subnet**. In order to establish connection-oriented network service the virtual-circuit subnet is required. The idea behind virtual circuits is to avoid having to choose a new route for every packet sent, as shown in Fig.7.7. 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. For example, consider the situation of Fig. 7.7. Here, host H1 has established connection 1 with host H2. It is remembered as the first entry in each of the routing tables.



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

Figure 7.7 Routing within a Virtual-Circuit Subnet

The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1. Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the subnet 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 1packets 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 is called label switching.

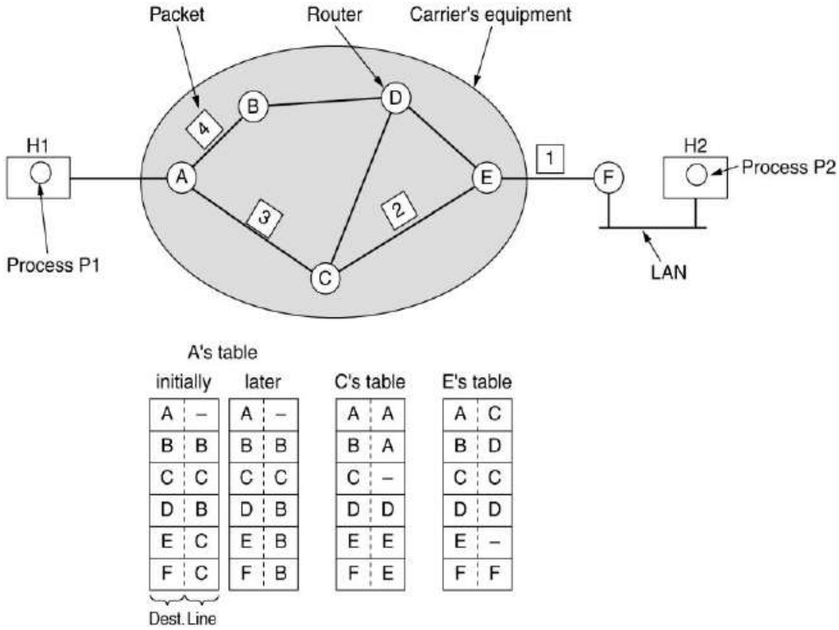
b). Implementation of Connectionless Service

A datagram is primarily used for wireless communication and is self-contained with source and destination addresses written in the header. If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called **Datagram** (in analogy with telegrams) and the subnet is called a **Datagram Subnet**. A datagram is a unit of transfer associated with networking. A datagram has the following characteristics:

i). Data is transmitted from source to destination without guarantee of delivery

ii). Data is frequently divided into smaller pieces and transmitted without a defined route or guaranteed order of delivery.

It is similar to a packet, which is a small piece of data transmitted through a connectionless protocol; but a datagram cannot handle prior or subsequent data communication. Intermediary devices (e.g., routers) automatically lead a datagram to its final network destination per the header's specified address, i.e., a datagram does not follow a predefined transmission route. Thus, the router does not require prior route information. In addition, successful datagram delivery is facilitated through the destination system's third-party application software. The process of routing within a datagram subnet is given in figure 7.8.



A's table

initially	
A	–
B	B
C	C
D	B
E	C
F	C

later	
A	–
B	B
C	C
D	B
E	B
F	B

C's table	
A	A
B	A
C	–
D	D
E	E
F	E

E's table	
A	C
B	D
C	C
D	D
E	–
F	F

Dest. Line

Figure 7.8. Routing within a Datagram Subnet

Table 7.2. Comparison of Virtual Circuit and Datagram Subnets

S.No.	VIRTUAL CIRCUIT	DATAGRAM SUBNETS
1.	VC is connection-oriented simply meaning that there is a reservation of resources like buffers, CPU, bandwidth, etc. for data transfer session.	Datagram is connectionless service. There is no need for reservation of resources as there is no dedicated path for a connection session.
2.	First packet goes and reserves resources for the subsequent packets which as a result follow the same path for the whole connection time.	All packets are free to go to any path on any intermediate router which is decided on the go by dynamically changing routing tables on routers.
3.	Since all the packets are going to follow the same path, a global header is required only for the first packet of the connection and other packets generally don't require global headers.	Since every packet is free to choose any path, all packets must be associated with a header with proper information about the source and the upper layer data.
4.	The data follows a particular dedicated path, packets reach in order to the destination.	The connectionless property makes data packets reach the destination in any order.
5.	VC is having high reliability, guarantee for all the packets will reach to the Destination.	Datagram networks are having less reliability. Some packet may discarded.
6.	VC is not cost-efficient for implementation to communicate	Datagram is cost-efficient for implementation to communicate.
7.	It is used by the Asynchronous Transfer Mode (ATM) Network, which is used for the Telephone calls.	It is generally used the IP network, which is used for Data services like Internet.

7.7 CHECK YOUR PROGRESS QUESTIONS

1. Define store-and- forward method.
2. What is Virtual Circuit?
3. Mention about Datagram Service.
4. Define bottleneck Problem.

7.8 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Store-and-forward means the intermediate nodes have the responsibility of transferring the entire message to the next node. Hence, each node must have storage capacity. A message will only be delivered if the next hop and the link connecting it are both available, otherwise it'll be stored indefinitely. A store-and-forward switch

forwards a message only if sufficient resources are available and the next hop is accepting data. This is called the store-and-forward property.

2. If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a virtual circuit, in analogy with the physical circuits set up by the telephone system, and the subnet is called a virtual-circuit subnet.
3. If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams and the subnet is called a datagram subnet.
4. When making data transmission between two nodes, If too many packets are present in the subnet at the same time, they will get into one another's way, forming bottlenecks problem

7.9 SUMMARY

The main function of the network layer is routing packets from the source machine to the destination machine. In most subnets, packets will require multiple hops to make the journey. 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. In large networks, there can be multiple paths from sender to receiver. The switching technique will decide the best route for data transmission. Switching technique is used to connect the systems for making one-to-one communication. There are different switching techniques available. Circuit switching is a switching technique that establishes a dedicated path between sender and receiver. The packet switching is a switching technique in which the message is sent in one goes, but it is divided into smaller pieces, and they are sent individually. Message Switching is a switching technique in which a message is transferred as a complete unit and routed through intermediate nodes at which it is stored and forwarded.

7.10 KEY WORDS

- **Switching** is the technique by which nodes control or switch data to transmit it between specific points on a network. Switched communication networks are those in which data transferred from source to destination is routed between various intermediate nodes.

- **Routing algorithm** is part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

- **Message delivery means** wrapping the entire information in a single message and transferring it from the source to the destination node. Each message must have a header that contains the message routing information, including the source and destination.

7.11 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1) Define data Packet.
2) Differentiate Circuit switching and packet switching.
3) What is meant by Message Switching?

Long Answer Questions

1. Explain about Circuit Witching.
2. Discuss about Packet Switching.
3. Write notes on Virtual Circuit and Datagram Subnets.

7.12 FURTHER READINGS

1. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.
2. Puzmanova, Routing and Switching: Time of Convergence?
3. Stevens, TCP/IP Illustrated, Vol. 1.
4. Computer Networks, 3rd Edition, Andrew S Tanenbaum, Pearson Education, 2010.
5. Data and Computer Communications, 8th Edition, William Stallings, Prentice Hall.
6. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008

UNIT-8 ROUTING ALGORITHM

Structure

8.0 INTRODUCTION

In this unit, we can have the idea about the purpose of routing algorithm is to make decisions for routers concerning the best path for data. Routing algorithm make decisions concerning the data path taken from one network to another. It acts as a traffic officer of the router. Routing algorithm is a formula that is stored in routers memory. Different routing algorithms use different methods to determine the best path. For example, a distance vector algorithm calculates a graph of all available routes by having each point or node determine the "cost" of travelling to each immediate neighbour. This information is collected for every node to create a distance table; which is used to determine the best path to from any one node to another.

8.1 OBJECTIVE

After study this unit we can have the knowledge about the following:

- Introduction to Routing algorithm
- Optimality Principle
- Static routing
- Shortest Path Routing
- Flooding

- Flow Based Routing
- Dynamic Routing
- Distance Vector Routing
- Link State Routing

8.2 INTRODUCTION TO ROUTING ALGORITHM

The algorithm that manages the tables and makes the routing decisions is called the **Routing Algorithm**. The foremost function of the network layer is routing packets from the source machine to the destination machine. In most subnets, 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. The algorithms that choose the routes and the data structures that they use are a major area of network layer design.

The routing algorithms are playing vital role in network communication system and are part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on. If the subnet uses datagram internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously-established route. The latter case is sometimes called **Session Routing** because a route remains in force for an entire user session (e.g., a login session at a terminal or a file transfer). 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.

Routing algorithms are classified in to two major categories namely Non-adaptive and Adaptive.

a). **Non-adaptive algorithms** do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J (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.

b). **Adaptive algorithms**, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive 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., every ΔT sec, when the load changes or when the topology 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, both static and dynamic.

8.2.1 Optimality Principle

Optimality means finding optimal route among various routes without regard to network topology or traffic. 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 Jr1 and the rest of the route r2. If a route better than r2 existed from J to K, it could be concatenated with r1 to improve the route from I to K, contradicting our statement that r1r2 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. 8.1, where the distance metric is the number of hops. Note that a sink tree is not necessarily unique; other trees with the same path lengths may exist. The goal of all routing algorithms is to discover and use the sink trees for all routers.

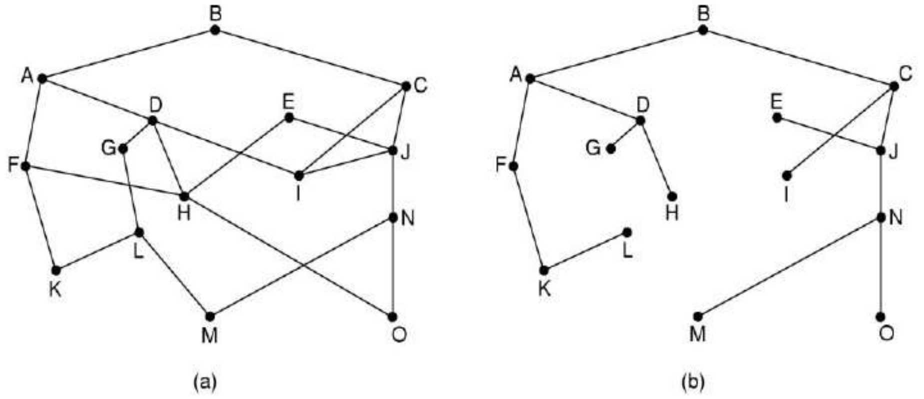


Figure 8.1. (a) A subnet. (b) A sink tree for router B

While a sink tree is definitely 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.

8.3 STATIC ROUTING

The static routing is most secure way of routing which reduces overhead from network resources. Static routing occurs when you, the network administrator manually add or configure routes on each router interface with IP addresses. This is no simple task, especially when you are administering a large network. Static routes are most often used to connect to a specific network or to provide a Gateway of Last Resort for a stub network. It is useful where numbers of route are limited. Like other routing methods static routing also has its pros and cons. The architecture of static routing is given in figure 8.2.

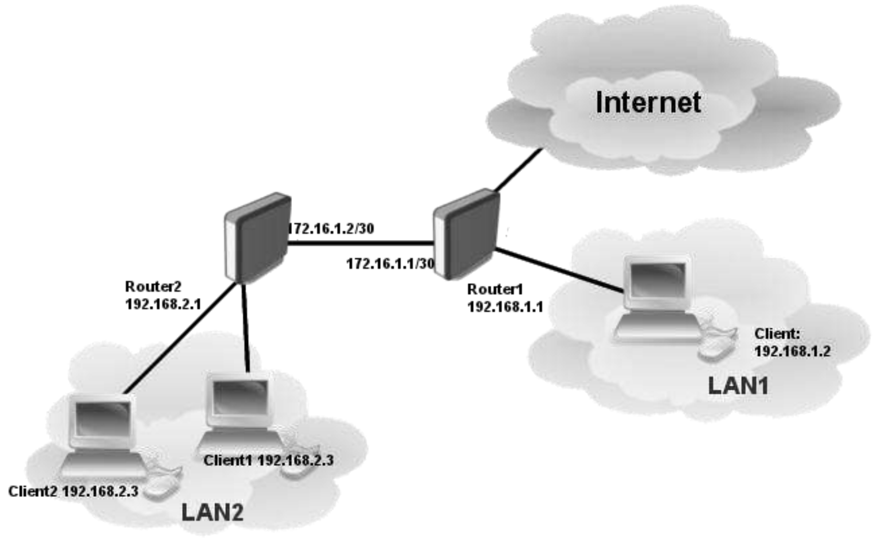


Figure 8.2. **Static Routing**

- **Advantage of Static Routing**

i). It is easy to implement.
ii). It is most secure way of routing, since no information is shared with other routers.
iii). It puts no overhead on res static routing sources such as CPU or memory.

- **Disadvantage of static routing**

i). It is suitable only for small network.
ii). If a link fails it cannot reroute the traffic.

8.4 SHORTEST PATH ROUTING

In routing process, the feasible routing algorithms uses many forms because it is simple and easy to understand. The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line (often called a link). To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph. The perception 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. 8.3 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).

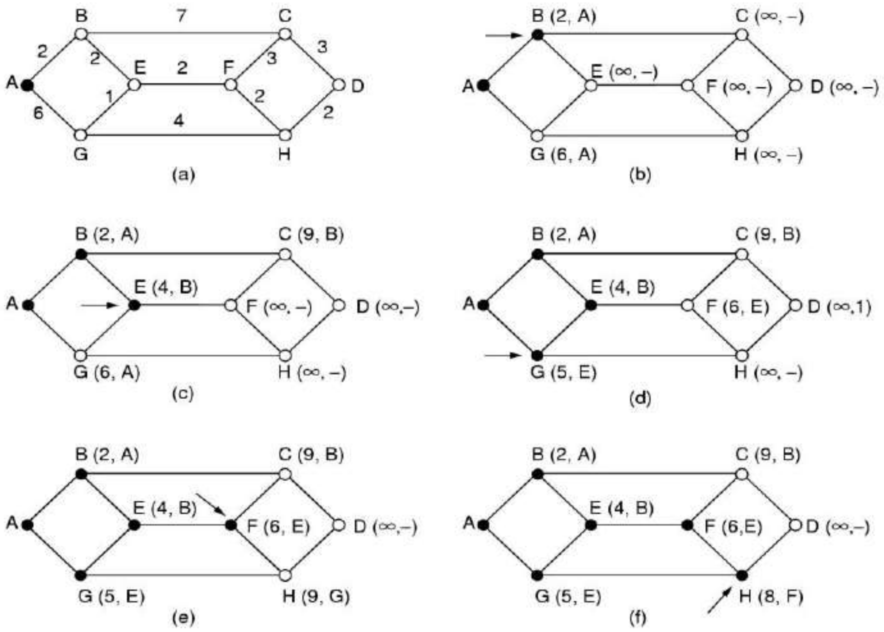


Figure 8.3. The first five steps used in computing the shortest path from A to D. The arrows indicate the working node.

However, many other metrics besides hops and physical distance are also possible. For example, each arc could be labelled with the mean queuing and transmission delay for some standard test packet as determined by hourly test runs. With this graph labelling, the shortest path is the fastest path rather than the path with the fewest arcs or kilometres. In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, 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). Each node is labelled (in parentheses) with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labelled 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 labelling algorithm works, look at the weighted, undirected graph of Fig. 8.3(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 relabelled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. Having examined each of the nodes adjacent to A, we examine all the tentatively labelled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. 8.3(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 relabelled.

8.5 FLOODING

The flooding is a simple routing technique or static algorithm utilized in computer networks for routing process where a source or node sends packets through every outgoing link. In Flooding static algorithm, every incoming packet is sent through every outgoing link except the one it arrived on. Flooding, which is similar to broadcasting, occurs when source packets (without routing data) are transmitted to all attached network nodes. Because flooding uses every path in the network, the shortest path is also used. The flooding algorithm is easy to implement. Flooding is used in bridging and in systems such as Usenet and peer-to-peer file sharing and as part of some routing protocols, including OSPF, DVMRP, and those used in ad-hoc wireless networks (WANETs).

There are generally two types of flooding available, uncontrolled flooding and controlled flooding.

a). **Uncontrolled flooding** is the fatal law of flooding. All nodes have neighbours and route packets indefinitely. More than two neighbours create a broadcast storm.

b). **Controlled flooding** has its own two algorithms to make it reliable, SNCF (Sequence Number Controlled Flooding) and RPF (Reverse Path Forwarding). In SNCF, the node attaches its own address and sequence number to the packet, since every node has a memory of addresses and sequence numbers. If it receives a packet in memory, it drops it immediately while in RPF, the node will only send the packet forward. If it is received from the next node, it sends it back to the sender.

There are several variants of flooding algorithms. Most work roughly as follows:

i). Each node acts as both a transmitter and a receiver.
ii). Each node tries to forward every message to every one of its neighbors except the source node.

This results in every message eventually being delivered to all reachable parts of the network. Algorithms may need to be more complex than this, since, in some case, precautions have to be taken to avoid wasted duplicate deliveries and infinite loops, and to allow messages to eventually expire from the system.

Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process. One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop, with the packet being discarded when the counter reaches zero. Ideally, the hop counter should be initialized to the length of the path from source to destination. If the sender does not know how long the path is, it can initialize the counter to the worst case, namely, the full diameter of the subnet.

The substitute technique for damming the flood is to keep track of which packets have been flooded, to avoid sending them out a second time. Achieve this goal is to have the source router put a sequence number in each packet it receives from its hosts. Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded. To prevent the list from growing without bound, each list should be augmented by a counter, k, meaning that all sequence numbers through k have been seen. When a packet comes in, it is easy to check if the packet is a duplicate; if so, it is discarded. Furthermore, the full list below k is not needed, since k effectively summarizes it.

A variant of flooding called selective flooding partially addresses these issues by only sending packets to routers in the same direction. In selective flooding the routers don't send every incoming packet on every line but only on those lines which are going approximately in the right direction. For example, in military applications, where large numbers of routers may be blown to bits at any instant, the tremendous robustness of flooding is highly desirable. In distributed database applications, it is sometimes necessary to update all the databases concurrently, in which case flooding can be useful. In wireless networks, all messages transmitted by a station can be received by all other stations within its radio range, which is, in fact, flooding, and some algorithms utilize this property. The process of flooding static algorithm is shown in fig. 8.4.

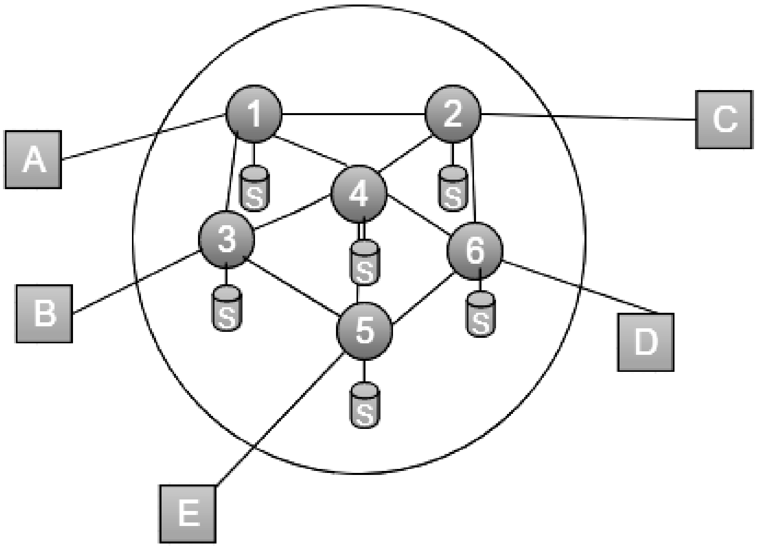


Figure 8.4.Flooding Static Algorithm

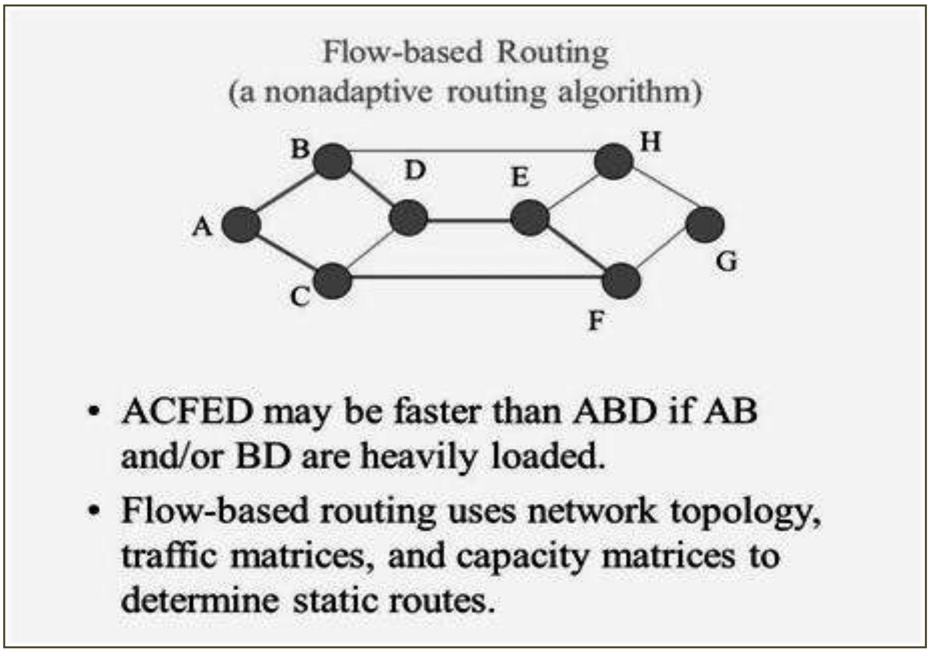
Table 8.1.Comparison of Routing and Flooding

Routing Algorithm	Flooding Algorithm
Routing table is required	Routing table is not required
May give shortest path	Always gives shortest path
Offer less reliability	Offer more reliability
Traffic density is less	Traffic density is more
No duplicate packets are present	Duplicate packets are present

8.6 FLOW BASED ROUTING

Flow routing is a network routing technology that takes variations in the flow of data into account to increase routing efficiency. The increased efficiency helps avoid excessive latency and jitter for streaming data, such as VoIP (voice over IP) or video. Rather than routing individual packets, a flow router observes and evaluates flows to gather statistics, including source, destination, amount of traffic "in flight," and stream duration. A flow is a single meaningful end-to-end activity over the network. This evaluation permits the router to prioritize traffic, deliver on quality of service (QoS) requirements, and keep flows from consuming more than some pre-allotted portion of network resources. A flow router evaluates traffic flows in real time, based on an ID, route, and time of receipt and rate of flow, to keep streaming traffic moving as quickly as possible. By contrast, conventional (Layer 3) IP routing does not differentiate between packets. Conventional routing uses a best-effort technique to ensure delivery of incoming traffic to the proper destination on a packet-by-packet basis, and is not sensitive to timing and data rate requirements for streaming data such as voice, video, multimedia and IPTV.

The industry analysts indicate that high-end conventional routers can achieve many of the same advantages touted for flow routing using ancillary queuing, deep-packet inspection, rate shaping and policing, and selective packet discard methods. Nevertheless, flow routing's unique ability to accommodate streaming data types makes it interesting and potentially valuable, especially for networks where such data consumes an appreciable portion of overall bandwidth and resources. The process of flow based routing is shown in fig. 8.5.



Flow-based Routing
(a nonadaptive routing algorithm)

- ACFED may be faster than ABD if AB and/or BD are heavily loaded.
- Flow-based routing uses network topology, traffic matrices, and capacity matrices to determine static routes.

Figure 8.5. Flow Based Routing

8.7 DYNAMIC ROUTING

Dynamic routing protocols have been used in networks since the early 1980s. The first version of RIP was released in 1982, but some of the basic algorithms within the protocol were used on the ARPANET as early as 1969. Dynamic routing is a networking technique that provides optimal data routing. Routing protocols determine the best path to each network, which is then added to the routing table. One of the primary benefits of using a dynamic routing protocol is that routers exchange routing information whenever there is a topology change. This exchange allows routers to automatically learn about new networks and also to find alternate paths if there is a link failure to a current network.

Unlike static routing, dynamic routing enables routers to select paths according to real-time logical network layout changes. In dynamic routing, the routing protocol operating on the router is responsible for the creation, maintenance and updating of the dynamic routing table. In static routing, all these jobs are manually done by the system administrator.

NOTES

Dynamic routing uses multiple algorithms and protocols. The most popular are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). Companies, organizations, and even service providers use an IGP on their internal networks. IGPs include RIP, EIGRP, OSPF, and IS-IS. Exterior Gateway Protocols (EGP): Used for routing between autonomous systems. The classification of dynamic routing algorithm is shown in fig. 8.6.

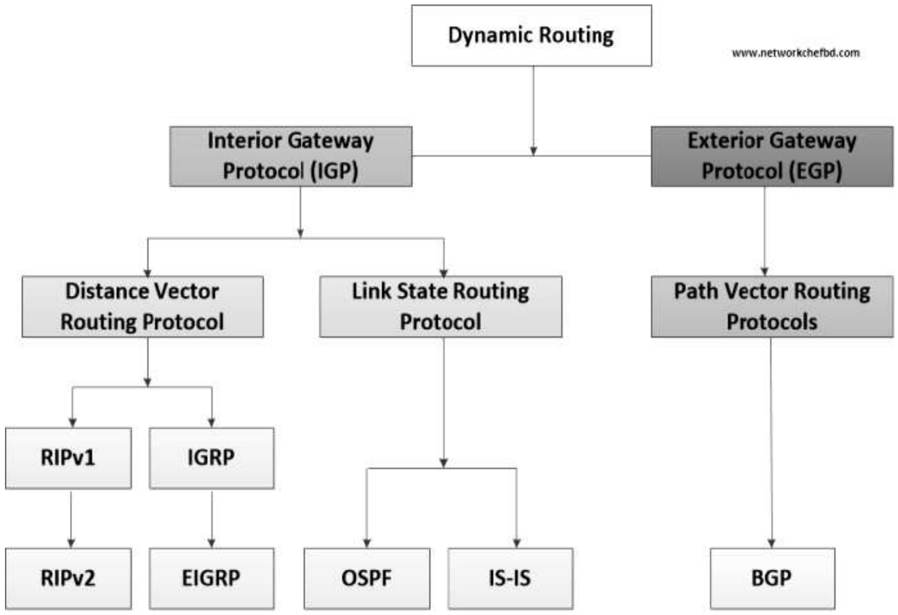

Dynamic Routing
www.networkchefbd.com
Interior Gateway Protocol (IGP)
Exterior Gateway Protocol (EGP)
Distance Vector Routing Protocol
Link State Routing Protocol
Path Vector Routing Protocols
RIPv1
IGRP
RIPv2
EIGRP
OSPF
IS-IS
BGP

Figure 8.6.Classification of Dynamic Routing Algorithm

- **Purpose of Dynamic Routing Protocols:**

A routing protocol is a set of processes, algorithms, and messages that are used to exchange routing information and populate the routing table with the routing protocol's choice of best paths. The purpose of a routing protocol includes

i). Discovering remote networks
ii). Maintaining up-to-date routing information
iii). Choosing the best path to destination networks
iv). Having the ability to find a new best path if the current path is no longer available
v). The components of a routing protocol are as follows:
vi). Data structures: Some routing protocols use tables or databases for their operations. This information is kept in RAM.
vii). Algorithm: An algorithm is a finite list of steps used in accomplishing a task. Routing protocols use algorithms for processing routing information and for best-path determination.
viii). Routing protocol messages: Routing protocols use various types of messages to discover neighbouring routers, exchange routing information, and do other tasks to learn and maintain accurate information about the network.

The major advantages of dynamic routing over static routing are scalability and adaptability. A dynamically routed network can grow more quickly and larger, and is able to adapt to changes in the network topology brought about by this growth or by the failure of one or more network components. Dynamic Routing Protocol Advantages Dynamic routing protocols provide several advantages, which will be discussed in this section. In many cases, the complexity of the network topology, the number of networks, and the need for the network to automatically adjust to changes require the use of a dynamic routing protocol. Before examining the benefits of dynamic routing protocols in more detail, you need to consider the reasons why you would use static routing. Dynamic routing certainly has several advantages over static routing; however, static routing is still used in networks today. In fact, networks typically use a combination of both static and dynamic routing. The differences between static and dynamic routing are given in table 8.2.

Table 8.2. Difference between Static and Dynamic Routing

BASIS	STATIC ROUTING	DYNAMIC ROUTING
Configuration	Manual	Automatic
Routing table building	Routing locations are hand-typed	Locations are dynamically filled in the table.
Routes	User defined	Routes are updated according to change in topology.
Routing algorithms	Doesn't employ complex routing algorithms.	Uses complex routing algorithms to perform routing operations.
Implemented in	Small networks	Large networks
Link failure	Link failure obstructs the rerouting.	Link failure doesn't affect the rerouting.
Security	Provides high security.	Less secure due to sending broadcasts and multicasts.
Routing protocols	No routing protocols are indulged in the process.	Routing protocols such as RIP, EIGRP, etc are involved in the routing process.
Additional resources	Not required	Needs additional resources to store the information.

NOTES

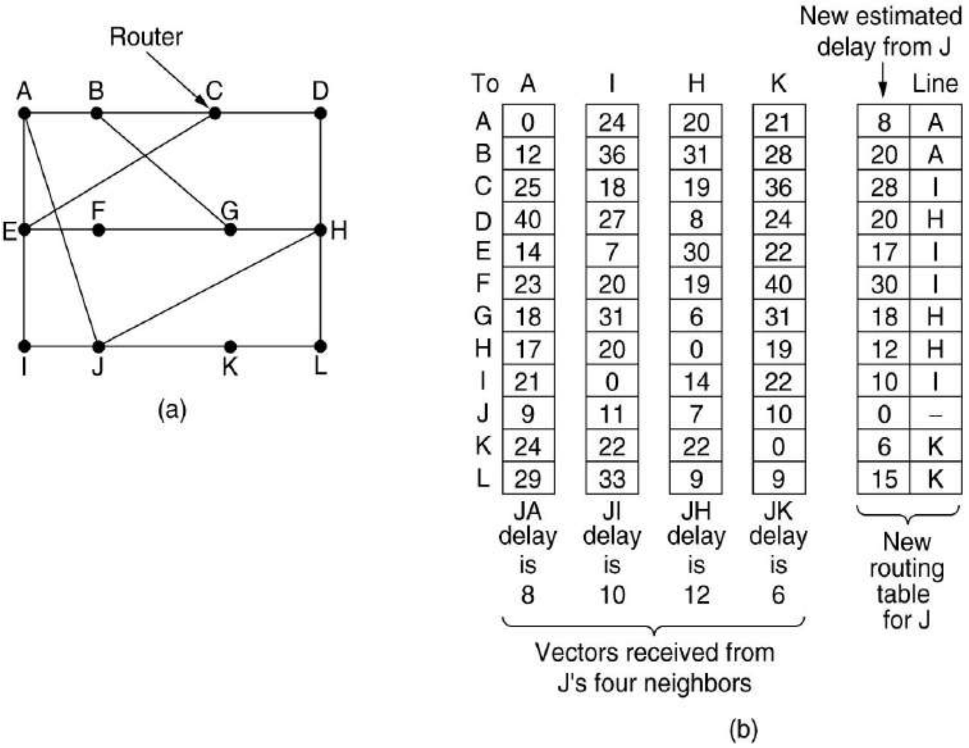
8.8 DISTANCE VECTOR ROUTING

The contemporary computer networks usually utilize dynamic routing algorithms rather than the static ones described above because static algorithms do not take the current network load into account. Two dynamic algorithms in particular, distance vector routing and link state routing, are the most popular. In this section we will look at the former algorithm. In the following section we will study the latter algorithm. Distance vector routing algorithms operate by having each router maintain a table (i.e, a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbors. The distance vector routing algorithm is sometimes called by other names, most commonly the distributed Bellman-Ford routing algorithm and the Ford-Fulkerson algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts: the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar. The router is assumed to know the "distance" to each of its neighbours. If the metric is hops, the distance is just one hop. If the metric is queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just timestamps and sends back as fast as it can.

As an example, assume that delay is used as a metric and that the router knows the delay to each of its neighbours. Once every T msec each router sends to each neighbour a list of its estimated delays to each destination. It also receives a similar list from each neighbour. Imagine that one of these tables has just come in from neighbour X, with Xi being X's estimate of how long it takes to get to router i. If the router knows that the delay to X is m msec, it also knows that it can reach router i via X in Xi + m msec. By performing this calculation for each neighbour, a router can find out which estimate seems the best and use that estimate and the corresponding line in its new routing table. Note that the old routing table is not used in the calculation.

This updating process is shown in Fig. 8.7 Part (a) shows a subnet. The first four columns of part (b) show the delay vectors received from the neighbours of router J. A claims to have a 12-msec delay to B, a 25-msec delay to C, a 40-msec delay to D, etc. Suppose that J has measured or estimated its delay to its neighbours, A, I, H, and K as 8, 10, 12, and 6 msec, respectively.


Router
A
B
C
D
E
F
G
H
I
J
K
L
(a)
New estimated delay from J
Vectors received from J's four neighbors
New routing table for J
(b)

To	A	I	H	K	Line	
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K
	JA delay is 8	JI delay is 10	JH delay is 12	JK delay is 6		

Figure 8.7. (a) A subnet. (b) Input from A, I, H, K, and the new Routing Table for J.

In this given diagram, identify that how J computes its new route to router G. It knows that it can get to A in 8 msec, and A claims to be able to get to G in 18 msec, so J knows it can count on a delay of 26 msec to G if it forwards packets bound for G to A. Similarly, it computes the delay to G via I, H, and K as 41 (31 + 10), 18 (6 + 12), and 37 (31 + 6) msec, respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to G is 18 msec and that the route to use is via H. The same calculation is performed for all the other destinations, with the new routing table shown in the last column of the figure.

8.9. LINK STATE ROUTING

In order to avoid the lengthy delay metric and the count-to-infinity problem, the new algorithm known as link state routing was introduced. The link state routing method is trouble-free and can be stated as five parts. Each router must do the following:

a). Discover its neighbours and learn their network addresses.
b). Measure the delay or cost to each of its neighbours.
c). Construct a packet telling all it has just learned.
d). Send this packet to all other routers.
e). Compute the shortest path to every other router.

In effect, the complete topology and all delays are experimentally measured and distributed to every router. Then Dijkstra's algorithm can be run to find the shortest path to every other router. Below we will consider each of these five steps in more detail.

a). Learning about the Neighbours

When a router is booted, its first task is to learn who its neighbours are. It accomplishes this goal by sending a special HELLO packet on each point-to-point line. The router on the other end is expected to send back a reply telling who it is. These names must be globally unique because when a distant router later hears that three routers are all connected to F, it is essential that it can determine whether all three mean the same F. When two or more routers are connected by a LAN, the situation is slightly more complicated.

The Fig. 8.8(a) illustrates a LAN to which three routers, A, C, and F, are directly connected. Each of these routers is connected to one or more additional routers, as shown. One way to model the LAN is to consider it as a node itself, as shown in Fig. 8.8(b). Here we have introduced a new, artificial node, N, to which A, C, and F are connected. The fact that it is possible to go from A to C on the LAN is represented by the path ANC here.

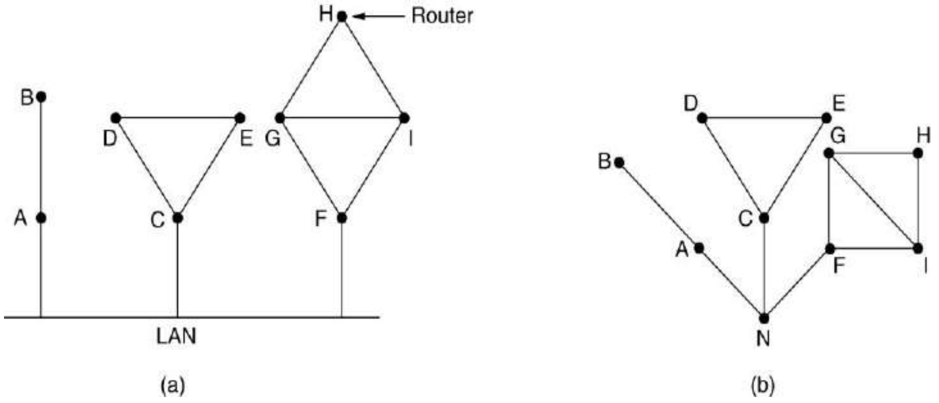


Figure 8.8. (a) Nine routers and a LAN. (b) A graph model of (a).

b). Measuring Line Cost

The link state routing algorithm requires each router to know, or at least have a reasonable estimate of, the delay to each of its neighbours. The most direct way to determine this delay is to send over the line a special ECHO packet that the other side is required to send back immediately. By measuring the round-trip time and dividing it by two, the sending router can get a reasonable estimate of the delay. For even better results, the test can be conducted several times, and the average used. Of course, this method implicitly assumes the delays are symmetric, which may not always be the case. An interesting issue is whether to take the load into account when measuring the delay. To factor the load in, the round-trip timer must be started when the ECHO packet is queued. To ignore the load, the timer should be started when the ECHO packet reaches the front of the queue.

Arguments can be made both ways. Including traffic-induced delays in the measurements means that when a router has a choice between two lines with the same bandwidth, one of which is heavily loaded all the time and one of which is not, the router will regard the route over the unloaded line as a shorter path. This choice will result in better performance. Unfortunately, there is also an argument against including the load in the delay calculation. Consider the subnet of Fig. 8.9, which is divided into two parts, East and West, connected by two lines, CF and EI.

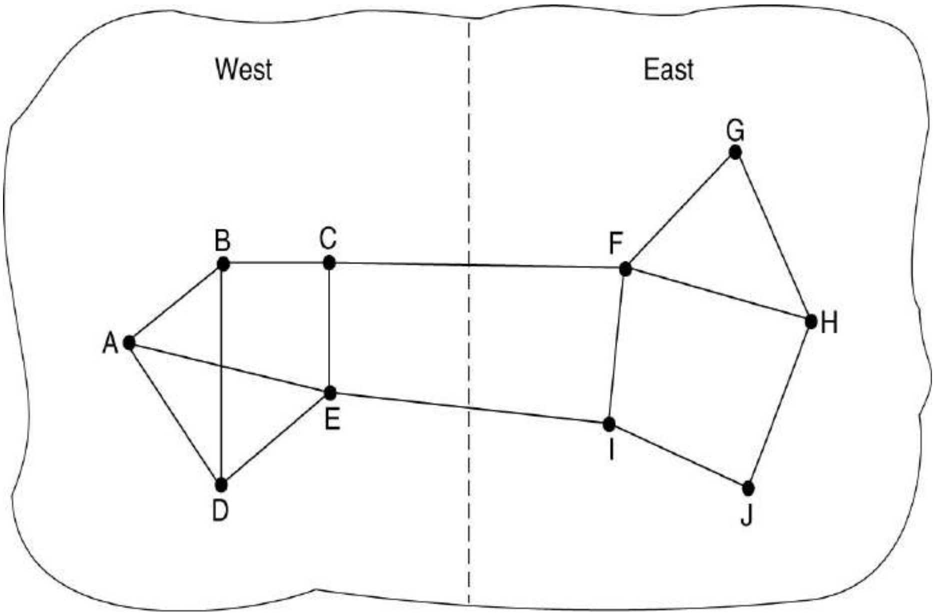
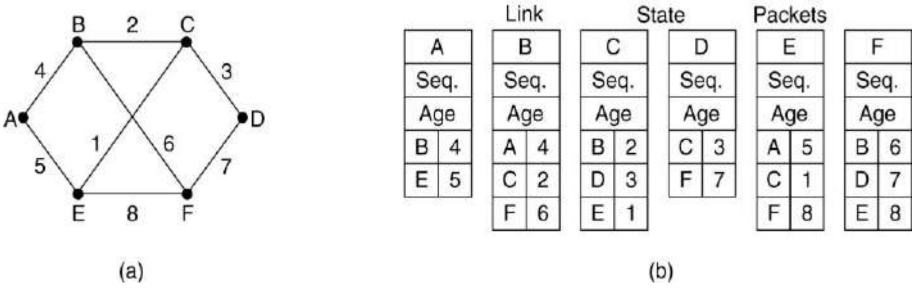


Figure 8.9. A subnet in which the East and West parts are connected by two lines.

Identify that the most of the traffic between East and West is using line CF, and as a result, this line is heavily loaded with long delays. Including queuing delay in the shortest path calculation will make EI more attractive. After the new routing tables have been installed, most of the East-West traffic will now go over EI, overloading this line. Consequently, in the next update, CF will appear to be the shortest path. As a result, the routing tables may oscillate wildly, leading to erratic routing and many potential problems. If load is ignored and only bandwidth is considered, this problem does not occur. Alternatively, the load can be spread over both lines, but this solution does not fully utilize the best path. Nevertheless, to avoid oscillations in the choice of best path, it may be wise to distribute the load over multiple lines, with some known fraction going over each line.

c). Building Link State Packets

Once the information needed for the exchange has been collected, the next step is for each router to build a packet containing all the data. The packet starts with the identity of the sender, followed by a sequence number and age (to be described later), and a list of neighbors. For each neighbor, the delay to that neighbor is given. An example subnet is given in Fig. 8.10 (a) with delays shown as labels on the lines. The corresponding link state packets for all six routers are shown in Fig. 8.10 (b). Building the link state packets is easy. The hard part is determining when to build them. One possibility is to build them periodically, that is, at regular intervals. Another possibility is to build them when some significant event occurs, such as a line or neighbour going down or coming back up again or changing its properties appreciably.



A	B	C	D	E	F
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.
Age	Age	Age	Age	Age	Age
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
	F 6	E 1		F 8	E 8

Figure 8.10. (a) A subnet. (b) The link state packets for this subnet.

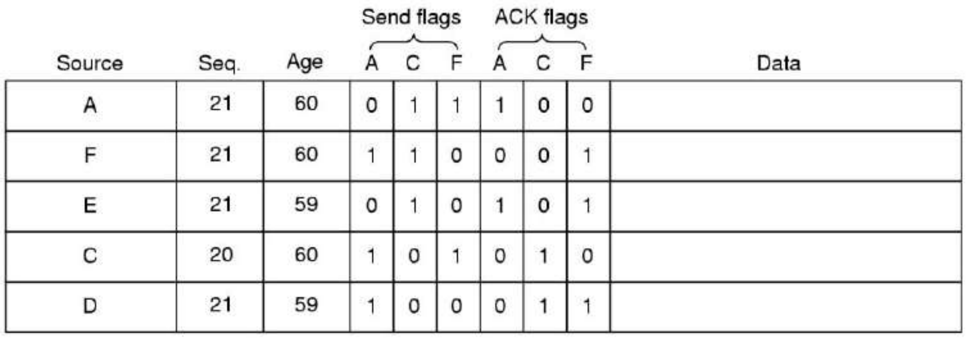
d). Distributing the Link State Packets

The trickiest part of the algorithm is distributing the link state packets reliably. As the packets are distributed and installed, the routers getting the first ones will change their routes. Consequently, the different routers may be using different versions of the topology, which can lead to inconsistencies, loops, unreachable machines, and other problems.First we will describe the basic distribution algorithm. Later we will give some refinements. The fundamental idea is to use flooding to distribute the link state packets. To keep the flood in check, each packet contains a sequence number that is incremented for each new packet sent. Routers keep track of all the (source router, sequence) pairs they see. When a new link state packet comes in, it is checked against the list of packets already seen. If it is new, it is forwarded on all lines except the one it arrived on. If it is a duplicate, it is discarded. If a packet with a sequence number lower than the highest one seen so far ever arrives, it is rejected as being obsolete since the router has more recent data.

This algorithm has a few problems, but they are manageable. First, if the sequence numbers wrap around, confusion will reign. The solution here is to use a 32-bit sequence number. With one link state packet per second, it would take 137 years to wrap around, so this possibility can be ignored. Second, if a router ever crashes, it will lose track of its sequence number. If it starts again at 0, the next packet will be rejected as a duplicate. Third, if a sequence number is ever corrupted and 65,540 is received instead of 4 (a 1-bit error), packets 5 through 65,540 will be rejected as obsolete, since the current sequence number is thought to be 65,540.

The solution to all these problems is to include the age of each packet after the sequence number and decrement it once per second. When the age hits zero, the information from that router is discarded. Normally, a new packet comes in, say, every 10 sec, so router information only times out when a router is down (or six consecutive packets have been lost, an unlikely event). The Age field is also decremented by each router during the initial flooding process, to make sure no packet can get lost and live for an indefinite period of time (a packet whose age is zero is discarded).When a link state packet comes in to a router for flooding, it is not queued for transmission immediately. Instead it is first put in a holding area to wait a short while. If another link state packet from the same source comes in before the first packet is transmitted, their sequence numbers are compared. If they are equal, the duplicate is discarded. If they are different, the older one is thrown out. To guard against errors on the router-router lines, all link state packets are acknowledged. When a line goes idle, the holding area is scanned in round-robin order to select a packet or acknowledgement to send.

The data structure used by router B for the subnet shown in Fig. 8.10 (a) is depicted in Fig. 8.11. Each row here corresponds to a recently-arrived, but as yet not fully-processed, link state packet. The table records where the packet originated, its sequence number and age, and the data. In addition, there are send and acknowledgement flags for each of B's three lines (to A, C, and F, respectively). The send flags mean that the packet must be sent on the indicated line. The acknowledgement flags mean that it must be acknowledged there.



			Send flags			ACK flags			
Source	Seq.	Age	A	C	F	A	C	F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Figure 8.11. The packet buffer for router B in Fig. 8.10.

In Fig. 8.11, the link state packet from A arrives directly, so it must be sent to C and F and acknowledged to A, as indicated by the flag bits. Similarly, the packet from F has to be forwarded to A and C and acknowledged to F. However, the situation with the third packet, from E, is different. It arrived twice, once via EAB and once via EFB. Consequently, it has to be sent only to C but acknowledged to both A and F, as indicated by the bits. If a duplicate arrives while the original is still in the buffer, bits have to be changed. For example, if a copy of C's state arrives from F before the fourth entry in the table has been forwarded, the six bits will be changed to 100011 to indicate that the packet must be acknowledged to F but not sent there.

e). Computing the New Routes

Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph because every link is represented. Every link is, in fact, represented twice, once for each direction. The two values can be averaged or used separately. Now Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations. The results of this algorithm can be installed in the routing tables, and normal operation resumed. For a subnet with n routers, each of which has k neighbors, the memory required to store the input data is proportional to kn. For large subnets, this can be a problem. Also, the computation time can be an issue. However, in many practical situations, link state routing works well. However, problems with the hardware or software can wreak havoc with this algorithm . For example, if a router claims to have a line it does not have or forgets a line it does have, the subnet graph will be incorrect. If a router fails to forward packets or corrupts them while forwarding them, trouble will arise. Finally, if it runs out of memory or does the routing calculation wrong, bad things will happen. As the subnet grows into the range of tens or hundreds of thousands of nodes, the probability of some router failing occasionally becomes non-negligible. The trick is to try to arrange to limit the damage when the predictable happens.

Table 8.3. Comparison of Distance Vector Routing and Link State Routing

BASIS FOR COMPARISON	DISTANCE VECTOR ROUTING	LINK STATE ROUTING
Algorithm	Bellman ford	Dijsktra
Network view	Topology information from the neighbour point of view	Complete information on the network topology
Best path calculation	Based on the least number of hops	Based on the cost
Updates	Full routing table	Link state updates
Updates frequency	Periodic updates	Triggered updates
CPU and memory	Low utilisation	Intensive
Simplicity	High simplicity	Requires a trained network administrator
Convergence time	Moderate	Fast
Updates	On broadcast	On multicast
Hierarchical structure	No	Yes
Intermediate Nodes	No	Yes

8.10 CHECK YOUR PROGRESS QUESTIONS

1. What is meant by Router?
2. State the role of Routing Protocol.
3. Define Flooding.

8.11 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. A **Router** is a networking device that forwards data packets between computer networks. A packet is typically forwarded from one router to another router through the networks that constitute an internetwork (e.g. the Internet) until it reaches its destination node.

2. The **Routing Protocol** is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination.

3. The **Flooding** static algorithm is used in computer networks for routing process. In Flooding static algorithm, every incoming packet is sent through every outgoing link except the one it arrived on. Flooding is used in bridging and in systems such as Usenet and peer-to-peer file sharing and as part of some routing protocols, including OSPF, DVMRP, and those used in ad-hoc wireless networks.

8.12 SUMMARY

In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted. A routing algorithm is a set of step-by-step operations used to direct Internet traffic efficiently. When a packet of data leaves its source, there are many different paths it can take to its destination. The routing algorithm is used to determine mathematically the best path to take. Alike to a wired router, a wireless router connects directly to a modem via a cable for receiving Internet data packets. Contrasting a wired router, which establishes a wired local area network (LAN), a wireless router establishes a wireless local area network (WLAN). The most common standard for WLAN is known as Wi-Fi. Whether the network layer provides datagram service or virtual circuit service, the main work of the network layer is to provide the best route. The routing protocol is performing this work.

8.13 KEY WORDS

- **Routing** is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm.
- **Session Routing** means a route remains in force for an entire user session such as login session at a terminal or a file transfer. One process is handling 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.
- **Non-adaptive algorithm** do not base their routing decisions on measurements or estimates of the current traffic and topology. This procedure is sometimes called static routing.
- **Adaptive algorithm** means change their routing decisions to reflect changes in the topology, and usually the traffic as well.

8.14 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Mention about Optimality Principle.
2. Define Static Routing.
3. State the importance of Shortest Path Routing.
4. Differentiate Routing and Flooding.
5. Compare Distance Vector Routing and Link State Routing.

Long Answer Questions

1. Discuss about Flow Based Routing.
2. Explain about Dynamic Routing and its categories.
3. Illustrate about Distance Vector Routing.
4. Elucidate about Link State Routing.

8.15 FURTHER READINGS

1. Data and Computer Communications, 8th Edition, William Stallings, Prentice Hall.
2. Computer Networks, 3rd Edition, Andrew S Tanenbaum, Pearson Education, 2010.
3. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.
4. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.

UNIT-9: OTHER ROUTING ALGORITHMS

Structure

9.0 INTRODUCTION

In this unit, the network layer is responsible for receiving data packets from the source all the way to destination or end-to-end transmission. In order to transferring data packets from source to destination the intermediate routers are playing significant role. There are numerous routing algorithms available for performing routing process such as Hierarchical Routing, Broadcast Routing, Multicast Routing and Congestion Control Algorithms.

9.1 OBJECTIVES

After go through this unit, we will elucidate about

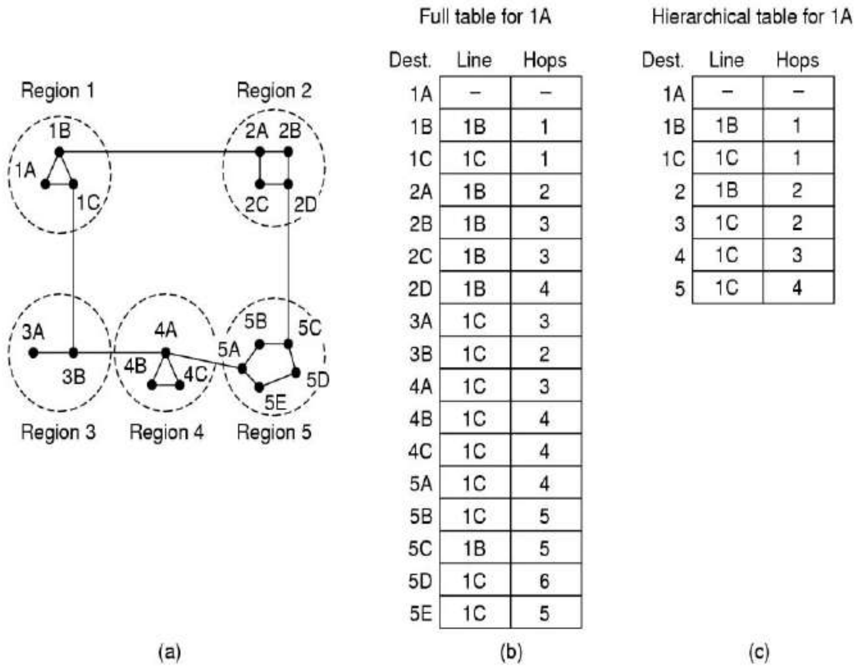
- Hierarchical Routing
- Broadcast Routing
- Multicast Routing
- Congestion Control Algorithms

9.2. HIERARCHICAL ROUTING

As networks grow in size, the router routing tables grow proportionally. Not only is router memory consumed by ever-increasing tables, but more CPU time is needed to scan them and more bandwidth is needed to send status reports about them. At a certain point the network may grow to the point where it is no longer feasible for every router to have an entry for every other router, so the routing will have to be done hierarchically, as it is in the telephone network.

When hierarchical routing is used, the routers are divided into what we will call **Regions**, with each router knowing all the details about how to route packets to destinations within its own region, but knowing nothing about the internal structure of other regions. When different networks are interconnected, it is natural to regard each one as a separate region in order to free the routers in one network from having to know the topological structure of the other ones.

For huge networks, a two-level hierarchy may be insufficient; it may be necessary to group the regions into clusters, the clusters into zones, the zones into groups, and so on, until we run out of names for aggregations. Figure 9.1 gives a quantitative example of routing in a two-level hierarchy with five regions. The full routing table for router 1A has 17 entries, as shown in Fig. 9.1(b). When routing is done hierarchically, as in Fig. 9.1(c), there are entries for all the local routers as before, but all other regions have been condensed into a single router, so all traffic for region 2 goes via the 1B -2A line, but the rest of the remote traffic goes via the 1C -3B line. Hierarchical routing has reduced the table from 17 to 7 entries. As the ratio of the number of regions to the number of routers per region grows, the savings in table space increase.



Full table for 1A

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

Hierarchical table for 1A

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

Figure 9.1.Hierarchical Routing

Unfortunately, these gains in space are not free. There is a penalty to be paid, and this penalty is in the form of increased path length. For example, the best route from 1A to 5C is via region 2, but with hierarchical routing all traffic to region 5 goes via region 3, because that is better for most destinations in region 5. When a single network becomes very large, an interesting question is: How many levels should the hierarchy have? For example, consider a subnet with 720 routers. If there is no hierarchy, each router needs 720 routing table entries. If the subnet is partitioned into 24 regions of 30 routers each, each router needs 30 local entries plus 23 remote entries for a total of 53 entries. If a three-level hierarchy is chosen, with eight clusters, each containing 9 regions of 10 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own cluster, and 7 entries for distant clusters, for a total of 25 entries. Kamoun and Kleinrock (1979) discovered that the optimal number of levels for an N router subnet is ln N, requiring a total of e ln N entries per router. They have also shown that the increase in effective mean path length caused by hierarchical routing is sufficiently small that it is usually acceptable.

9.3. BROADCAST ROUTING

The process of Sending a packet to all destinations simultaneously is called **broadcasting**; various methods have been proposed for doing it. In some applications, hosts need to send messages to many or all other hosts. For example, a service distributing weather reports, stock market updates, or live radio programs might work best by broadcasting to all machines and letting those that are interested read the data. One broadcasting method that requires no special features from the subnet is for the source to simply send a distinct packet to each destination. Not only is the method wasteful of bandwidth, but it also requires the source to have a complete list of all destinations. In practice this may be the only possibility, but it is the least desirable of the methods.

Flooding is another obvious candidate. Although flooding is ill-suited for ordinary point-to-point communication, for broadcasting it might rate serious consideration, especially if none of the methods described below are applicable. The problem with flooding as a broadcast technique is the same problem it has as a point-to-point routing algorithm: it generates too many packets and consumes too much bandwidth. A third algorithm is multi destination routing. If this method is used, each packet contains either a list of destinations or a bit map indicating the desired destinations.

When a packet arrives at a router, the router checks all the destinations to determine the set of output lines that will be needed. The router generates a new copy of the packet for each output line to be used and includes in each packet only those destinations that are to use the line. In effect, the destination set is partitioned among the output lines. After a sufficient number of hops, each packet will carry only one destination and can be treated as a normal packet. The multi destination routing is like separately addressed packets, except that when several packets must follow the same route, one of them pays full fare and the rest ride free. The fig 9.2 shows the architecture of broadcasting system.

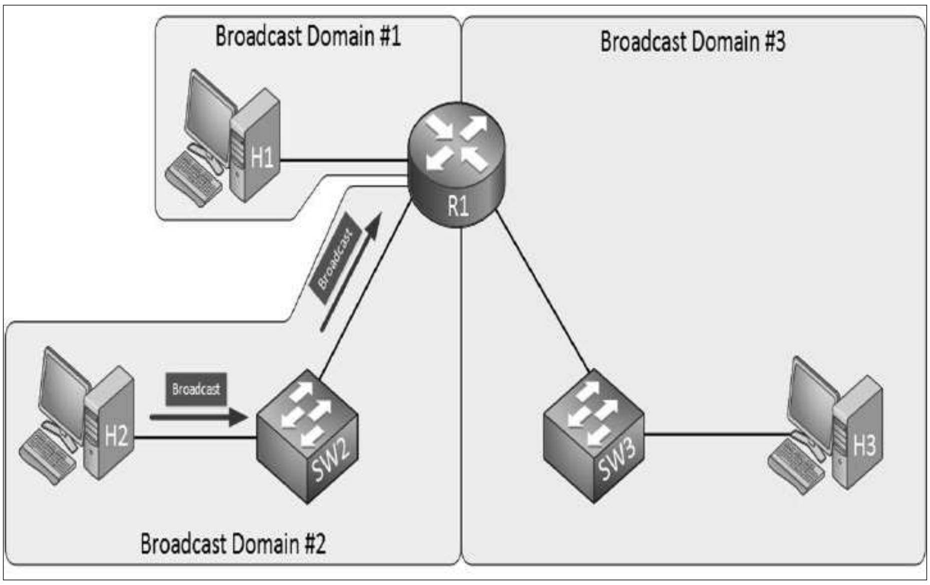


Figure 9.2. Broadcast Routing

A fourth broadcast algorithm makes explicit use of the sink tree for the router initiating the broadcast or any other convenient spanning tree for that matter. A **spanning tree** is a subset of the subnet that includes all the routers but contains no loops. If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet onto all the spanning tree lines except the one it arrived on. This method makes excellent use of bandwidth, generating the absolute minimum number of packets necessary to do the job. The only problem is that each router must have knowledge of some spanning tree for the method to be applicable. Sometimes this information is available (e.g., with link state routing) but sometimes it is not (e.g., with distance vector routing).

Our last broadcast algorithm is an attempt to approximate the behavior of the previous one, even when the routers do not know anything at all about spanning trees. The idea, called **Reverse Path Forwarding**, is remarkably simple once it has been pointed out. When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line that is normally used for sending packets to the source of the broadcast. If so, there is an excellent chance that the broadcast packet itself followed the best route from the router and is therefore the first copy to arrive at the router. This being the case, the router forwards copies of it onto all lines except the one it arrived on. If, however, the broadcast packet arrived on a line other than the preferred one for reaching the source, the packet is discarded as a likely duplicate.

An example of reverse path forwarding is shown in Fig. 9.3. Part (a) shows a subnet, part (b) shows a sink tree for router I of that subnet, and part (c) shows how the reverse path algorithm works. On the first hop, I send packets to F, H, J, and N, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to I (assuming that the preferred path falls along the sink tree) and is so indicated by a circle around the letter. On the second hop, eight packets are generated, two by each of the routers that received a packet on the first hop. As it turns out, all eight of these arrive at previously unvisited routers, and five of these arrive along the preferred line. Of the six packets generated on the third hop, only three arrive on the preferred path (at C, E, and K); the others are duplicates. After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly. The principal advantage of reverse path forwarding is that it is both reasonably efficient and easy to implement. It does not require routers to know about spanning trees, nor does it have the overhead of a destination list or bit map in each broadcast packet as does multi destination addressing. Nor does it require any special mechanism to stop the process, as flooding does (either a hop counter in each packet and a priori knowledge of the subnet diameter, or a list of packets already seen per source).

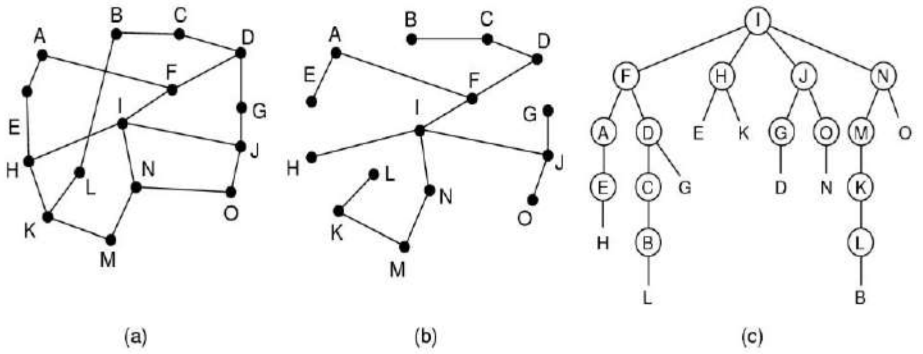


Figure 9.3.Reverse Path Forwarding. (a) A Subnet. (b) A Sink Tree. (c) The Tree Built by Reverse Path Forwarding.

9.4. MULTI CAST ROUTING

Some applications require that widely-separated processes work together in groups, for example, a group of processes implementing a distributed database system. In these situations, it is frequently necessary for one process to send a message to all the other members of the group. If the group is small, it can just send each other member a point-to-point message. If the group is large, this strategy is expensive. Sometimes broadcasting can be used, but using broadcasting to inform 1000 machines on a million-node network is inefficient because most receivers are not interested in the message (or worse yet, they are definitely interested but are not supposed to see it). Thus, we need a way to send messages to well-defined groups that are numerically large in size but small compared to the network as a whole.

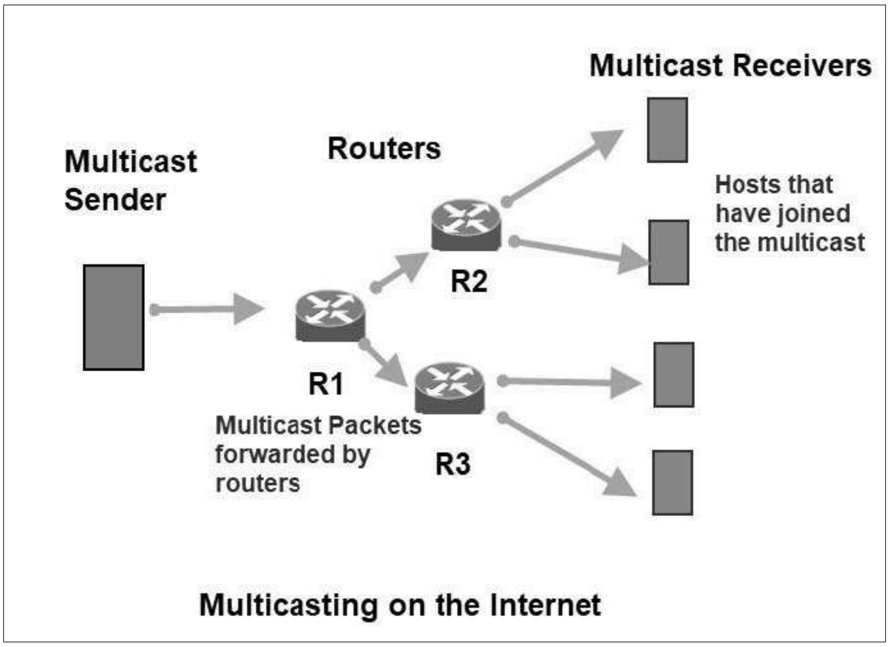

Multicast Receivers
Multicast Sender
Routers
Hosts that have joined the multicast
R2
R1
R3
Multicast Packets forwarded by routers
Multicasting on the Internet

Figure 9.4. Multicast Routing

Sending a message to such a group is called multicasting, and its routing algorithm is called **Multicast Routing**. In this section we will describe one way of doing multicast routing. Multicasting requires group management. Some way is needed to create and destroy groups, and to allow processes to join and leave groups. How these tasks are accomplished is not of concern to the routing algorithm. What is of concern is that when a process joins a group, it informs its host of this fact. It is important that routers know which of their hosts belong to which groups. Either hosts must inform their routers about changes in group membership, or routers must query their hosts periodically. Either way, routers learn about which of their hosts are in which groups. Routers tell their neighbors, so the information propagates through the subnet. To do multicast routing, each router computes a spanning tree covering all other routers. For example, in Fig. 9.5(a) we have two groups, 1 and 2. Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure. A spanning tree for the leftmost router is shown in Fig. 9.5(b).

When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group. In our example, Fig. 9.5(c) shows the pruned spanning tree for group 1. Similarly, Fig. 9.5(d) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

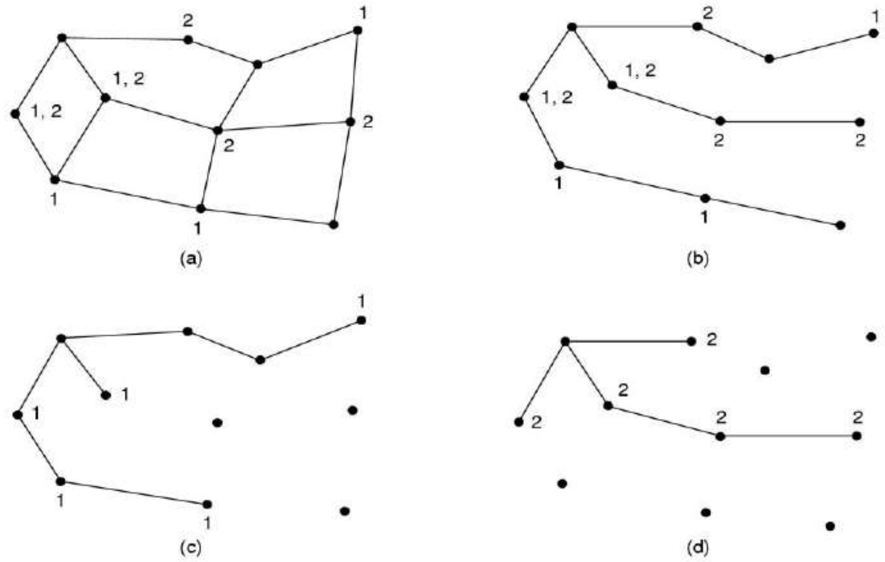


Figure 9.5.(a) A network. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1 (d) A multicast tree for group 2.

Table 9.1.Comparison of Broadcast Routing and Multicast

BASIS FOR COMPARISON	BROADCAST ROUTING	MULTICAST ROUTING
Basic	The packet is transmitted to all the hosts connected to the network.	The packet is transmitted only to intended recipients in the network.
Transmission	One-to-all.	One-to-many.
Management	Broadcasting does not require any group management.	Multicasting requires group management to define the group of hosts/stations which will receive packets.
Bandwidth	Bandwidth is wasted.	Bandwidth is utilized efficiently.
Traffic	Unnecessarily huge amount traffic is generated in the network.	Traffic is under control.
Process	Slow.	Fast.

NOTES

9.5. CONGESTION CONTROL ALGORITHMS

When too many packets are present in the subnet, performance degrades. This situation is called **congestion** which is shown in figure 9.6. When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except for a few that are afflicted with transmission errors) and the number delivered is proportional to the number sent. However, as traffic increases too far, the routers are no longer able to cope and they begin losing packets. This tends to make matters worse. At very high traffic, performance collapses completely and almost no packets are delivered.

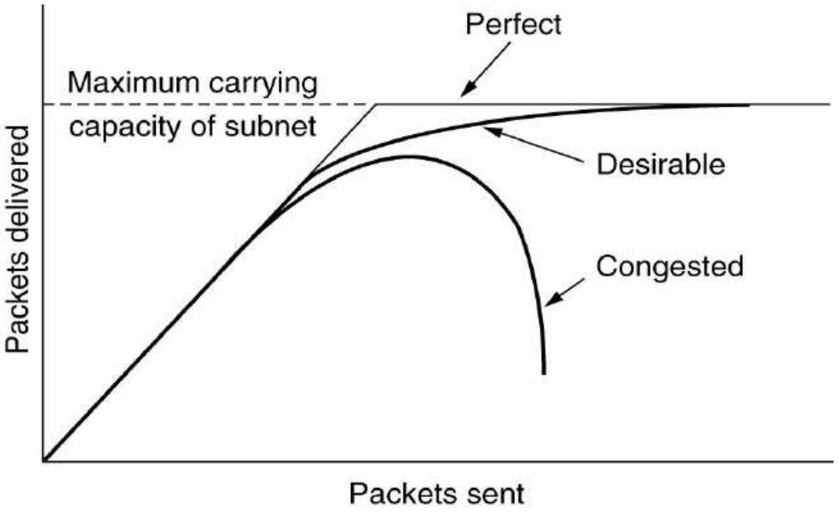


Figure 9.6.Congestion

Congestion can be brought on by several factors. If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packets will be lost. Adding more memory may help up to a point, but Nagle (1987) discovered that if routers have an infinite amount of memory, congestion gets worse, not better, because by the time packets get to the front of the queue, they have already timed out (repeatedly) and duplicates have been sent. All these packets will be dutifully forwarded to the next router, increasing the load all the way to the destination.

The Slow processors can also cause congestion. If the routers' CPUs are slow at performing the bookkeeping tasks required of them (queuing buffers, updating tables, etc.), queues can build up, even though there is excess line capacity. Similarly, low-bandwidth lines can also cause congestion. Upgrading the lines but not changing the processors, or vice versa, often helps a little, but frequently just shifts the bottleneck. Also, upgrading part, but not all, of the system, often just moves the bottleneck somewhere else. The real problem is frequently a mismatch between parts of the system. This problem will persist until all the components are in balance. It is worth explicitly pointing out the difference between congestion control and flow control, as the relationship is subtle. Congestion control has to do with making sure the subnet is able to carry the offered traffic. It is a global issue, involving the behaviour of all the hosts, all the routers, the store-and-forwarding processing within the routers, and all the other factors that tend to diminish the carrying capacity of the subnet.

Flow control, in contrast, relates to the point-to-point traffic between a given sender and a given receiver. Its job is to make sure that a fast sender cannot continually transmit data faster than the receiver is able to absorb it. Flow control frequently involves some direct feedback from the receiver to the sender to tell the sender how things are doing at the other end. To see the difference between these two concepts, consider a fiber optic network with a capacity of 1000 gigabits/sec on which a supercomputer is trying to transfer a file to a personal computer at 1 Gbps. Although there is no congestion (the network itself is not in trouble), flow control is needed to force the supercomputer to stop frequently to give the personal computer a chance to breathe.

At the other extreme, consider a store-and-forward network with 1-Mbps lines and 1000 large computers, half of which are trying to transfer files at 100 kbps to the other half. Here the problem is not that of fast senders overpowering slow receivers, but that the total offered traffic exceeds what the network can handle. The reason congestion control and flow control are often confused is that some congestion control algorithms operate by sending messages back to the various sources telling them to slow down when the network gets into trouble. Thus, a host can get a "slow down" message either because the receiver cannot handle the load or because the network cannot handle it.

Table 9.2. Comparison of Flow Control and **Congestion Control**

BASIS FOR COMPARISON	FLOW CONTROL	CONGESTION CONTROL
Basic	It controls the traffic from a particular sender to a receiver.	It controls the traffic entering the network.
Purpose	It prevents the receiver from being overwhelmed by the data.	It prevents the network from getting congested.
Responsibility	Flow control is the responsibility handled by data link layer and the transport layer.	Congestion Control is the responsibility handled by network layer and transport layer.
Responsible	The sender is responsible for transmitting extra traffic at receivers side.	The transport layer is responsible transmitting extra traffic into the network.
Preventive measures	The sender transmits the data slowly to the receiver.	Transport layer transmits the data into the network slowly.
Methods	Feedback-based flow control and Rate-based flow control	Provisioning, traffic-aware routing and admission control

NOTES

9.6 CHECK YOUR PROGRESS QUESTIONS

1. What is Optimality Principle?
2. Define Broadcasting.
3. State the importance of Multicasting.

9.7 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. **Optimality Principle:** It is very usefulto collect the information regarding optimal routes without regard to network topology or traffic. This statement is known as the optimality principle**.**
2. **Broadcasting** means sending a packet to all destinations simultaneously and various methods available for doing it.
3. **Multicasting** means sending a message to such a group and its routing algorithm is called **Multicast Routing**. Multicasting requires group management. Some way is needed to create and destroy groups, and to allow processes to join and leave groups.

9.8 SUMMARY

The network layer provides services to the transport layer. It can be based on either virtual circuit or datagram. Many routing algorithms are used in computer networks. Static algorithms include shortest path routing and flooding. Dynamic algorithms include distance vector routing and link state routing. Most actual networks use one of these. Other important routing topics are hierarchical routing, routing for mobile hosts, broadcast routing, multicast routing, and routing in peer-to-peer networks. Subnets can easily become congested, increasing the delay and lowering the throughput for packets. Network designers attempt to avoid congestion by proper design. Techniques include retransmission policy, caching, flow control, and more. If congestion does occur, it must be dealt with. Choke packets can be sent back, load can be shed, and other methods applied. The next step beyond just dealing with congestion is to actually try to achieve a promised quality of service.

9.9 KEY WORDS

- **Sink Tree:** 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

- **Spanning tree:** A **spanning tree** is a subset of the subnet that includes all the routers but contains no loops. If each router knows which of its lines belong to the spanning tree, it can copy an incoming broadcast packet on to all the spanning tree lines except the one it arrived on. This method utilizes the bandwidth, generating the absolute minimum number of packets excellently.

- **Congestion**: When too many packets are present in the subnet, the performance degrades. This situation is called congestion.

9.10 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. What is meant by Reverse Path Forwarding?
2. Mention about Congestion Control Algorithms.

Long Answer Questions

1. Explain about Hierarchical Routing.
2. Differentiate Broadcast Routing and Multicast Routing.

9.11 FURTHER READINGS

1. Computer Networks, 3rd Edition, Andrew S Tanenbaum, Pearson Education, 2010.
2. Data and Computer Communications, 8th Edition, William Stallings, Prentice Hall.
3. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.
4. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.

BLOCK 4: TRANSPORT LAYER

UNIT-10 TRANSPORT LAYER

Structure

10.0 INTRODUCTION

In this unit, we can have the idea about the **heart of whole protocol hierarchy** is known as **Transport Layer**. The transport layer offers efficient, reliable and economic data transport from source to destination computer. The transport layer makes use of the services provided by the network layer. The hardware and/or software within the transport layer that does the work are called the **transport entity**. The transport entity can be located in the operating system kernel, in a separate user process, in a library package bound into network applications, or conceivably on the network interface card.

In essence, the existence of the transport layer makes it possible for the transport service to be more reliable than the underlying network service. Lost packets and mangled data can be detected and compensated for by the transport layer. The bottom four layers can be seen as the **Transport Service Provider**, whereas the upper layer(s) are the **Transport Service User**. This distinction of provider versus user has a considerable impact on the design of the layers and puts the transport layer in a key position, since it forms the major boundary between the provider and user of the reliable data transmission service.

10.1 OBJECTIVES

After going through this unit, you will explain about

- Functions of Transport Layer
- Design Issues with Transport Layer
- Process to process delivery
- User Datagram Protocol (UDP)
- Transmission Control Protocol (TCP)
- TCP Service Model
- TCP Segment Header
- Connection oriented service
- Connectionless services.

10.2 INTRODUCTION TO TRANSPORT LAYER

The basic function of the Transport layer is to accept data from the layer above, split it up into smaller units, pass these data units to the Network layer, and ensure that all the pieces arrive correctly at the other end. Furthermore, all this must be done efficiently and in a way that isolates the upper layers from the inevitable changes in the hardware technology.

The Transport layer also determines what type of service to provide to the Session layer, and, ultimately, to the users of the network. The most popular type of transport connection is an **error-free point-to-point channel** that delivers messages or bytes in the order in which they were sent. The Transport layer is a true end-to-end layer, all the way from the source to the destination. In other words, a program on the source machine carries on a conversation with a similar program on the destination machine, using the message headers and control messages.

10.2.1 Functions of Transport Layer

a). **Service Point Addressing:** Transport Layer header includes service point address which is port address. This layer gets the message to the correct process on the computer unlike Network Layer, which gets each packet to the correct computer.

b). **Segmentation and Reassembling:** A message is divided into segments; each segment contains sequence number, which enables this layer in reassembling the message. Message is reassembled correctly upon arrival at the destination and replaces packets which were lost in transmission.

c). **Connection Control:** It includes 2 types:

- Connectionless Transport Layer: Each segment is considered as an independent packet and delivered to the transport layer at the destination machine.

- Connection Oriented Transport Layer: Before delivering packets, connection is made with transport layer at the destination machine.

d). **Flow Control:** In this layer, flow control is performed end to end.

e). **Error Control:** Error Control is performed end to end in this layer to ensure that the complete message arrives at the receiving transport layer without any error. Error Correction is done through retransmission.

10.2.2 Design Issues with Transport Layer

- Accepting data from Session layer, split it into segments and send to the network layer.
- Ensure correct delivery of data with efficiency.
- Isolate upper layers from the technological changes.
- Error control and flow control.

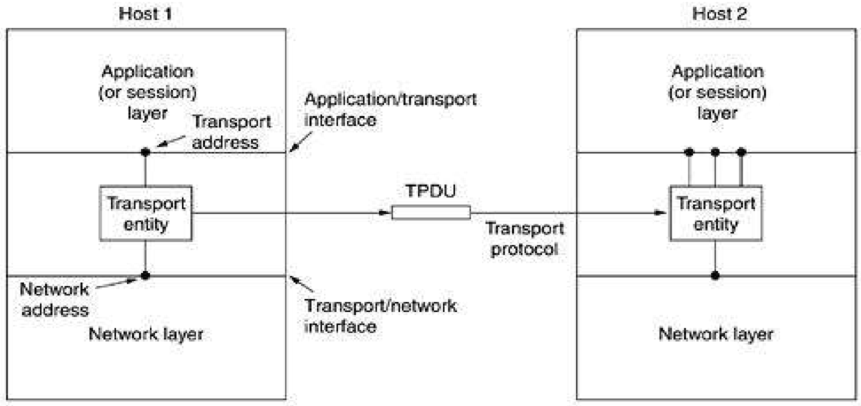


Figure 10.1. The Network, Transport and Application Layers.

In order to acquire an idea of what a transport service might be like, consider the five primitives listed in Table 10.1. This transport interface is truly bare bones, but it gives the essential flavour of what a connection-oriented transport interface has to do. It allows application programs to establish, use, and then release connections, which is sufficient for many applications.

Table. 10.1. The Primitives for a Simple Transport Service.

Primitive	Meaning
SOCKET	Create a new communication and port
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections, give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEOVE	Receive some data from the connection
CLOSE	Release the connection

10.3. PROCESS TO PROCESS DELIVERY

Transport layer is responsible for process-to-process delivery, the delivery of a packet, part of a message, from one process to another. Two processes communicate in a client-server relationship. The most common process-to-process communication is though the Client/Server Paradigm.

Client : A process on the local host

Server : A process on the remote host to provide services.

The Internet model has three protocols at the transport layer: UDP, TCP, and SCTP. The data link layer is responsible for delivery of frames between two neighboring nodes over a link. This is called node-to-node delivery. The network layer is responsible for delivery of datagrams between two hosts. This is called host-to-host delivery. Communication on the Internet is not defined as the exchange of data between two nodes or between two hosts. Real communication takes place between two processes. So that we need process-to-process delivery.

However, at any moment, several processes may be running on the source host and several on the destination host. To complete the delivery, we need a mechanism to deliver data from one of these processes running on the source host to the corresponding process running on the destination host.The transport layer is responsible for process-to-process delivery-the delivery of a packet, part of a message, from one process to another. The following figure shows these three types of deliveries and their domains.

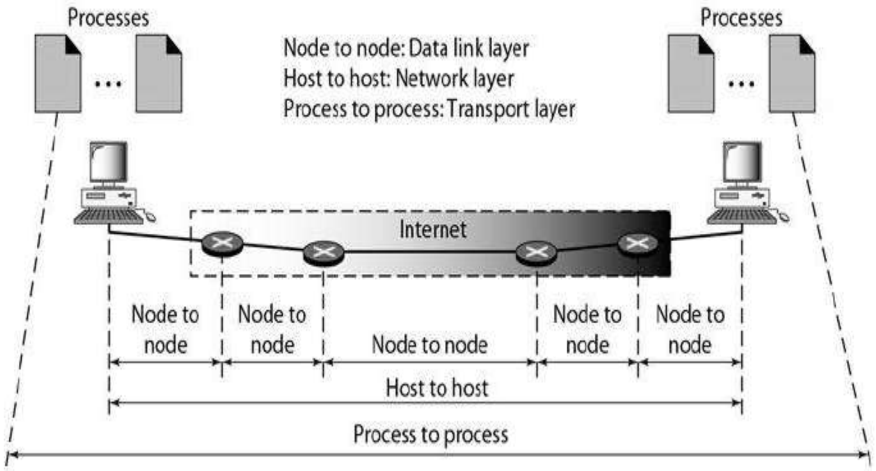


Figure 10.2. The Process-to-Process Delivery

- **Client Server Paradigm:**

There are several ways to achieve process-to-process communication; the most common one is through the client/server paradigm. A process on the local host, called a client, needs services from a process usually on the remote host, called a server.

i). User Datagram Protocol (UDP)
ii). TCP services
iii). TCP Segment

Both processes (client and server) have the same name. For example, to get the day and time from a remote machine, we need a Daytime client process running on the local host and a Daytime server process running on a remote machine.

A remote computer can run several server programs at the same time, just as local computers can run one or more client programs at the same time. For communication, we must define the following:

i). Local host
ii). Local process
iii). Remote host
iv). Remote process

a). Addressing

Whenever we need to deliver something to one specific destination among many, we need an address. At the data link layer, we need a MAC address to choose one node among several nodes if the connection is not point-to-point. A frame in the data link layer needs a destination MAC address for delivery and a source address for the next node's reply.At the network layer, we need an IP address to choose one host among millions. A datagram in the network layer needs a destination IP address for delivery and a source IP address for the destination's reply.

At the transport layer, we need a transport layer address, called a port number, to choose among multiple processes running on the destination host. The destination port number is needed for delivery; the source port number is needed for the reply. In the Internet model, the port numbers are 16-bit integers between 0 and 65,535. The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host. This is the ephemeral port number.

The server process must also define itself with a port number. This port number, however, cannot be chosen randomly. If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and use its services will not know the port number.

Every client process knows the well-known port number of the corresponding server process. For example, while the Daytime client process, discussed above, can use an ephemeral (temporary) port number 52,000 to identify itself, the Daytime server process must use the well-known (permanent) port number 13.

It should be clear by now that the IP addresses and port numbers play different roles in selecting the final destination of data. The destination IP address defines the host among the different hosts in the world. After the host has been selected, the port number defines one of the processes on this particular host.

b). IANA Ranges:

The lANA (Internet Assigned Number Authority) has divided the port numbers into three ranges: well known, registered, and dynamic (or private).

i). Well-known ports: The ports ranging from 0 to 1023 are assigned and controlled by lANA. These are the well-known ports.

ii). Registered ports: The ports ranging from 1024 to 49,151 are not assigned or controlled by lANA. They can only be registered with lANA to prevent duplication.

iii). Dynamic ports: The ports ranging from 49,152 to 65,535 are neither controlled nor registered. They can be used by any process. These are the ephemeral ports.

c). Socket Addresses:

The Process-to-process delivery needs two identifiers, IP address and the port number, at each end to make a connection. The combination of an IP address and a port number is called a socket address. The client socket address defines the client process uniquely just as the server socket address defines the server process uniquely which is represented in the figure 10.3.

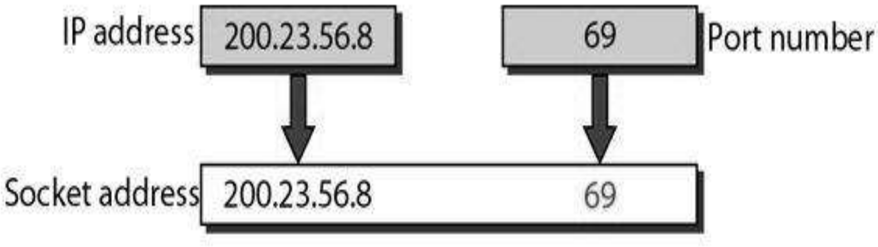


Figure 10.3. The Socket Address

d). Multiplexing and Demultiplexing:

The addressing mechanism allows multiplexing and demultiplexing by the transport layer.

- **Multiplexing:**

At the sender site, there may be several processes that need to send packets. However, there is only one transport layer protocol at any time. This is a many-to-one relationship and requires multiplexing. The protocol accepts messages from different processes, differentiated by their assigned port numbers. After adding the header, the transport layer passes the packet to the network layer.

- **Demultiplexing:**

At the receiver site, the relationship is one-to-many and requires demultiplexing. The transport layer receives datagrams from the network layer. After error checking and dropping of the header, the transport layer delivers each message to the appropriate process based on the port number.

e). Connectionless Versus Connection-Oriented Service

A transport layer protocol can either be connectionless or connection-oriented.

- **Connectionless Service:**

In a connectionless service, the packets are sent from one party to another with no need for connection establishment or connection release. The packets are not numbered; they may be delayed or lost or may arrive out of sequence. There is no acknowledgment either. UDP, is connectionless.

- **Connection-Oriented Service:**

In a connection-oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released. The TCP and SCTP are connection-oriented protocols.

f). Reliable Versus Unreliable:

The transport layer service can be reliable or unreliable. If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer. If the application program does not need reliability because it uses its own flow and error control mechanism or it needs fast service or the nature of the service does not demand flow and error control (real-time applications), then an unreliable protocol can be used. In the Internet, UDP is connectionless and unreliable; TCP and SCTP are connection oriented and reliable.

10.4. USER DATAGRAM PROTOCOL (UDP)

The User Datagram Protocol (UDP) is a connectionless oriented transport protocol in Internet protocol suite. UDP provides a way for applications to send encapsulated IP datagrams and send them without having to establish a connection. UDP is described in RFC 768. UDP (User

Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss-tolerating connections between applications on the internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. But there are important differences between the two. Where UDP enables process-to-process communication, TCP supports host-to-host communication. TCP sends individual packets and is considered a reliable transport medium; UDP sends messages, called Datagram, and is considered a best-effort mode of communications.

Additionally, TCP offers error and flow control mechanism; no such mechanisms are available in UDP. UDP is considered a connectionless protocol because it doesn't require a virtual circuit to be established before any data transfer occurs. UDP provides two services not provided by the IP layer. It provides port numbers to help distinguish different user requests and, optionally, a checksum capability to verify that the data arrived intact. TCP has emerged as the dominant protocol used for the bulk of internet connectivity due to its ability to break large data sets into individual packets, check for and resend lost packets, and reassemble packets in the correct sequence. But these additional services come at a cost in terms of additional data overhead and delays called latency. In contrast, UDP just sends the packets, which means that it has much lower bandwidth overhead and latency. With UDP, packets may take different paths between sender and receiver and, as a result, some packets may be lost or received out of order. UDP transmits segments consisting of an 8-byte header followed by the payload. The header is shown in Fig. 10.4. The two ports serve to identify the end points within the source and destination machines.

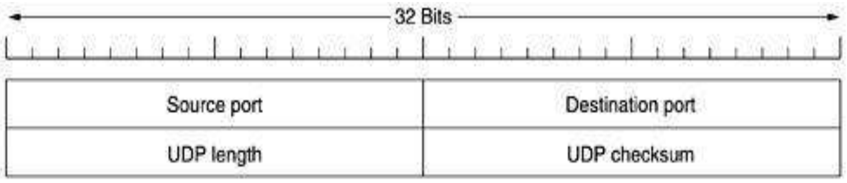


Figure 10.4. The UDP Header.

a) User datagram protocol features

The user datagram protocol has attributes that make it advantageous for use with applications that can tolerate lost data.

i). It allows packets to be dropped and received in a different order than they were transmitted, making it suitable for real-time applications where latency might be a concern.

ii). It can be used for transaction-based protocols, such as DNS or Network Time Protocol.

iii). It can be used where a large number of clients are connected and where real-time error correction isn't necessary, such as gaming, voice or video conferencing, and streaming media.

b) UDP header composition

The User Datagram Protocol header has four fields, each of which is 2 bytes. They are:

i). source port number, which is the number of the sender;

ii). destination port number, the port the datagram is addressed to;

iii). length, the length in bytes of the UDP header and any encapsulated data; and

iv). Checksum, which is used in error checking. Its use is required in IPv6 and optional in IPv4.

c) Applications of UDP

UDP is an ideal protocol for network applications in which perceived latency is critical, such as in gaming and voice and video communications, which can suffer some data loss without adversely affecting perceived quality. In some cases, forward error correction techniques are used to improve audio and video quality in spite of some loss. UDP can also be used in applications that require lossless data transmission when the application is configured to manage the process of retransmitting lost packets and correctly arranging received packets. This approach can help to improve the data transfer rate of large files compared to TCP. In the Open Systems Interconnection (OSI) communication model, UDP, like TCP, is in Layer 4, the transport layer. UDP works in conjunction with higher level protocols to help manage data transmission services including Trivial File Transfer Protocol (TFTP), Real Time Streaming Protocol (RTSP), Simple Network Protocol (SNP) and domain name system (DNS) lookups.

10.5 TRANSMISSION CONTROL PROTOCOL (TCP)

The **Transmission Control Protocol (TCP)** is a connection-oriented reliable protocol. It provides a reliable transport service between pairs of processes executing on End Systems (ES) using the network layer service provided by the IP protocol. The TCP was specifically designed to provide a reliable end-to-end byte stream over an unreliable internetwork. An internetwork differs from a single network because different parts may have wildly different topologies, bandwidths, delays, packet sizes, and other parameters. TCP was designed to dynamically adapt to properties of the internetwork and to be robust in the face of many kinds of failures. TCP was formally defined in RFC 793. As time went on, various errors and inconsistencies were detected, and the requirements were changed in some areas. These clarifications and some bug fixes are detailed in RFC 1122. Extensions are given in RFC 1323.

Each machine supporting TCP has a TCP transport entity, either a library procedure, a user process, or part of the kernel. In all cases, it manages TCP streams and interfaces to the IP layer. A TCP entity accepts user data streams from local processes, breaks them up into pieces not exceeding 64 KB (in practice, often 1460 data bytes in order to fit in a single Ethernet frame with the IP and TCP headers), and sends each piece as a separate IP datagram. When datagrams containing TCP data arrive at a machine, they are given to the TCP entity, which reconstructs the original byte streams. For simplicity, we will sometimes use just "TCP" to mean the TCP transport entity (a piece of software) or the TCP protocol (a set of rules). From the context it will be clear which is meant. For example, in "The user gives TCP the data," the TCP transport entity is clearly intended.

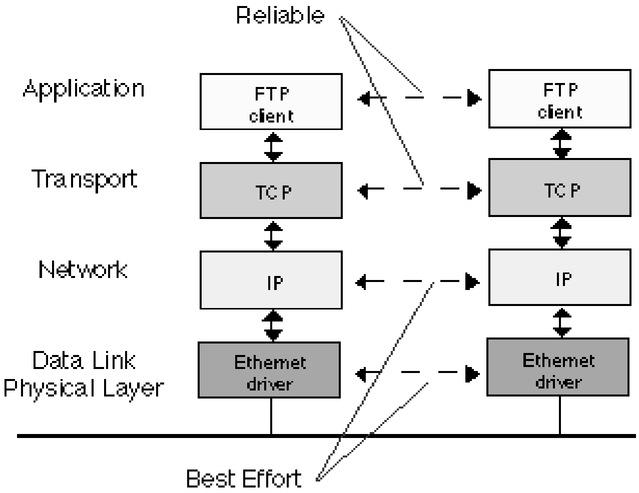


Figure 10.5. Transmission Control Protocol.

10.5.1 The TCP Service Model

TCP service is obtained by both the sender and receiver creating end points, called **sockets**. Each socket has a socket number (address) consisting of the IP address of the host and a 16-bit number local to that host, called a **port**. A port is the TCP name for a TSAP. For TCP service to be obtained, a connection must be explicitly established between a socket on the sending machine and a socket on the receiving machine. A socket may be used for multiple connections at the same time. In other words, two or more connections may terminate at the same socket. Connections are identified by the socket identifiers at ends **(*socket1, socket2*).**

No virtual circuit numbers or other identifiers are used. Port numbers below 1024 are called **well-known ports** and are reserved for standard services. For example, any process wishing to establish a connection to a host to transfer a file using FTP can connect to the destination host's port 21 to contact its FTP daemon. A few of the better known ones are listed in table 10.2.

Table. 10.2. Some assigned ports.

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial file transfer protocol
79	Finger	Lookup information about a user
80	HTTP	World Wide web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

It would certainly be possible to have the FTP daemon attach itself to port 21 at boot time, the telnet daemon to attach itself to port 23 at boot time, and so on. However, doing so would clutter up memory with daemons that were idle most of the time. Instead, what is generally done is to have a single daemon, called **inetd** (**Internet daemon**) in UNIX, attach itself to multiple ports and wait for the first incoming connection. When that occurs, inetd forks off a new process and executes the appropriate daemon in it, letting that daemon handle the request. In this way, the daemons other than inetd are only active when there is work for them to do. Inetd learns which ports it is to use from a configuration file. Consequently, the system administrator can set up the system to have permanent daemons on the busiest ports (e.g., port 80) and inetd on the rest.

Some early applications used the PUSH flag as a kind of marker to delineate messages boundaries. While this trick sometimes works, it sometimes fails since not all implementations of TCP pass the PUSH flag to the application on the receiving side. Furthermore, if additional PUSHes come in before the first one has been transmitted (e.g., because the output line is busy), TCP is free to collect all the PUSHed data into a single IP datagram, with no separation between the various pieces. One last feature of the TCP service that is worth mentioning here is **urgent data**. When an interactive user hits the DEL or CTRL-C key to break off a remote computation that has already begun, the sending application puts some control information in the data stream and gives it to TCP along with the URGENT flag. This event causes TCP to stop accumulating data and transmit everything it has for that connection immediately.

When the urgent data are received at the destination, the receiving application is interrupted (e.g., given a signal in UNIX terms) so it can stop whatever it was doing and read the data stream to find the urgent data. The end of the urgent data is marked so the application knows when it is over. The start of the urgent data is not marked. It is up to the application to figure that out. This scheme basically provides a crude signalling mechanism and leaves everything else up to the application.

10.5.2 TCP Segment Header

The TCP provides a great deal of functionality, as reflected in the complexity of its header. The Figure 10.6 presents the layout of a TCP segment. Every segment begins with a fixed-format, 20-byte header. The fixed header may be followed by header options. After the options, if any, up to 65,535 - 20 - 20 = 65,495 data bytes may follow, where the first 20 refer to the IP header and the second to the TCP header. Segments without any data are legal and are commonly used for acknowledgements and control messages. As shown in Figure 10.6, the following are the TCP header fields:

i). **Source Port** The port is used to transmit the packet.

ii). **Destination Port** The port to which the packet will be transmitted.

iii). **Sequence Number** The number used to identify a TCP segment. This field is used to ensure that parts of a data stream are not missing.

iv). **Acknowledgment Number** The sequence number that is to be expected in the next packet from the other device taking part in the communication.

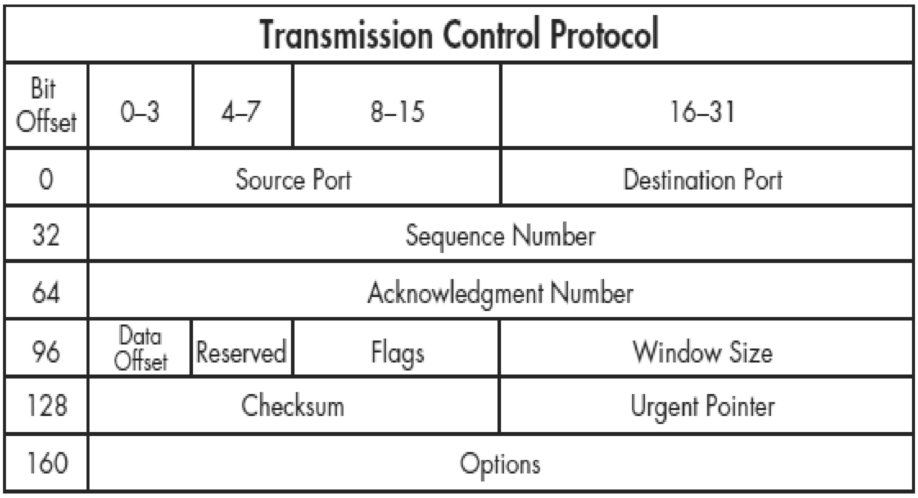
v). **Flags** The URG, ACK, PSH, RST, SYN, and FIN flags for identifying the type of TCP packet being transmitted.

vi). **Window Size** The size of the TCP receiver buffer in bytes.

vii). **Checksum** Used to ensure the contents of the TCP header and data are intact upon arrival.

viii). **Urgent Pointer** If the URG flag is set, this field is examined for additional instructions for where the CPU should begin reading the data within the packet.

ix). **Options** Various optional fields that can be specified in a TCP packet.



Transmission Control Protocol				
Bit Offset	0–3	4–7	8–15	16–31
0	Source Port			Destination Port
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Reserved	Flags	Window Size
128	Checksum			Urgent Pointer
160	Options			

Figure 10.6. TCP Header

10.6. CONNECTION ORIENTED Vs CONNECTIONLESS SERVICES.

These are the two services given by the layers to layers above them. These services are:

a). Connection Oriented Service

b). Connectionless Services

a) Connection Oriented Services

There is a sequence of operation to be followed by the users of connection oriented service. These are:

- ➢ Connection is established.
    - o Information is sent.
    - o Connection is released.

In connection oriented service we have to establish a connection before starting the communication. When connection is established, we send the message or the information and then we release the connection. Connection oriented service is more reliable than connectionless service. Example of connection oriented is TCP (Transmission Control Protocol) protocol.

b) Connectionless Services

It is similar to the postal services, as it carries the full address where the message (letter) is to be carried. Each message is routed independently from source to destination. The order of message sent can be different from the order received. In connectionless the data is transferred in one direction from source to destination without checking that destination is still there or not or if it prepared to accept the message. Authentication is not needed in this. Example of Connectionless service is UDP (User Datagram Protocol) protocol.

c) Difference between Connection oriented and Connectionless service

i). In connection oriented service authentication is needed, while connectionless service does not need any authentication.

ii). Connection oriented protocol makes a connection and checks whether message is received or not and sends again if an error occurs, while connectionless service protocol does not guarantees a message delivery.

iii). Connection oriented service is more reliable than connectionless service.

iv). Connection oriented service interface is stream based and connectionless is message based.

10.7 CHECK YOUR PROGRESS QUESTIONS

1. What is Transport Entity?
2. Define Service Point Addressing.
3. Mention about Segmentation and Reassembling.

10.8 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. **Transport Entity:** The hardware and/or software within the transport layer that does the work are called the **transport entity**. The transport entity can be located in the operating system kernel.

2. **Service Point Addressing:** Transport Layer header includes service point address which is port address. This layer gets the message to the correct process on the computer unlike Network Layer, which gets each packet to the correct computer.

3. **Segmentation and Reassembling:** A message is divided into segments; each segment contains sequence number, which enables this layer in reassembling the message. Message is reassembled correctly upon arrival at the destination and replaces packets which were lost in transmission.

10.9 SUMMARY

The transport layer offers dissimilar services such as end-to-end, reliable, connection oriented byte stream from sender to receiver. Transport protocols are capable of doing connection management more than unreliable networks. The internet hastwo significanttransport protocols namely TCP and UDP.

10.10 KEY WORDS

- The bottom four layers of OSI model are known as **Transport Service Provider.**

- The upper three layers of OSI model are known as **Transport Service User**.

- TCP service is obtained by both the sender and receiver creating end points, called **sockets**. Each socket has a socket number or address.

- **Source Port:** The port is used to transmit the packet.

- **Destination Port** : The port to which the packet will be transmitted.

- **TCP\IP:** Transmission Control protocol.
- **UDP:** User Datagram Protocol.

10.11 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. State the Applications of UDP
2. What is meant by Internet daemon?
3. Differentiate Connection oriented and Connectionless Service.

Long Answer Questions

3. Explain about TCP.
4. Discuss about the role of UDP.

10.12 FURTHER READINGS

1. IEEE Internet Computing, Jan.-Feb. 2000
2. Comer, The Internet Book
3. Garber, "Will 3G Really Be the Next Big Wireless Technology?"
4. Computer Networks, 3rd Edition, Andrew.S.Tanenbaum, Pearson Education, 2010.
5. Data and Computer Communications, 8th Edition, William Stallings, Prentice Hall.
6. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.
7. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.

UNIT-11: APPLICATIONS AND SERVICES

Structure

11.0 INTRODUCTION

In this unit, you will study about transport layer applications and services. To allow users to access the transport service, the transport layer must provide some operations to application programs, that is, a transport service interface. Each transport service has its own interface. The transport service is similar to the network service, but there are also some important differences. The main difference is that the network service is intended to model the service offered by real networks, growths and all. The connection-oriented transport service, in contrast, is reliable. Of course, real networks are not error-free, but that is precisely the purpose of the transport layer to provide a reliable service on top of an unreliable network.

11.1 OBJECTIVES

After going through this unit, you will elucidate about

- Uniform Resource Locator (URL)
- Domain Name System Architecture
- Types of Name Servers
- DNS Working Procedure

- Remote Login
- Mail Exchange
- Mail Architecture and Services
- Message Formats
- File Transfer

11.2. DOMAIN NAME SYSTEM

The **Domain Name System** facilitates to determine the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names.

When **DNS** was not into existence, one had to download a **Host file** containing host names and their corresponding IP address. But with increase in number of hosts of internet, the size of host file also increased. This resulted in increased traffic on downloading this file. To solve this problem the DNS system was introduced.

IP Address

IP address is a unique logical address assigned to a machine over the network. An IP address exhibits the following properties:

i). IP address is the unique address assigned to each host present on Internet.
ii). IP address is 32 bits (4 bytes) long.
iii). IP address consists of two components: network component and host component.
iv). Each of the 4 bytes is represented by a number from 0 to 255, separated with dots. For example 137.170.4.124

IP address is 32-bit number while on the other hand domain names are easy to remember names. For example, when we enter an email address we always enter a symbolic string such as webmaster@auk.com.

11.2.1 Uniform Resource Locator (URL)

The Uniform Resource Locator (URL) refers to a web address which uniquely identifies a document over the internet. This document can be a web page, image, audio, video or anything else present on the web.

For example, **www.auk.com/internet_technology/index.html** is an URL to the index.html which is stored on auk web server under internet technology directory.

URL Types

There are two forms of URL as listed below:

a). Absolute URL
b). Relative URL

a). Absolute URL

Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

For example http:// www.auk.com / internet_technology /index.htm.where:

- **http** is the protocol.
- **auk.com** is the server name.
- **index.htm** is the file name.

The protocol part tells the web browser how to handle the file. Similarly we have some other protocols also that can be used to create URL are:

- FTP
- https
- Gopher
- mailto
- news

b). Relative URL

Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL. Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the link.

For example, to link an image on

auk.com/internet_technology/internet_referemce_models,

we can use the relative URL which can take the form

like **/internet_technologies/internet-osi_model.jpg.**

The difference between Absolute and Relative URL are offered in table 11.1.

Table. 11.1. Difference between Absolute and Relative URL

Absolute URL	Relative URL
Used to link web pages on different websites	Used to link web pages within the same website.
Difficult to manage.	Easy to Manage
Changes when the server name or directory name changes	Remains same even of we change the server name or directory name.
Take time to access	Comparatively faster to access.

11.2.2 Domain Name System Architecture

The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

i). Domain Names

Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as **com, edu, gov, net** etc, while some country level domain names such as **au, in, za, us** etc.The following table shows the common Top-Level Domain names:

Table. 11.2. Common Top-Level Domain Names

Domain Name	Description
com	Commercial organizations such as business
edu	Educational organizations such as universities
gov	US government organizations etc
Int	International organizations
mil	US military organization
net	A network that doesn't fit into one of the other organizational domain categories
org	An organization that doesn't fit into one of the other organizational domain categories

The following table shows the **Country top-level** domain names:

Table. 11.3. Country top-level Domain Names

Domain Name	**Meaning**
au	Australia
in	India
cl	Chile
fr	France
us	United States
za	South Africa
uk	United Kingdom
jp	Japan
es	Spain
de	Germany
ca	Canada
ee	Estonia
hk	Hong Kong

ii). Domain Name Space

The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. The figure 11.1.shows the domain name space hierarchy.In the figure 11.1, each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

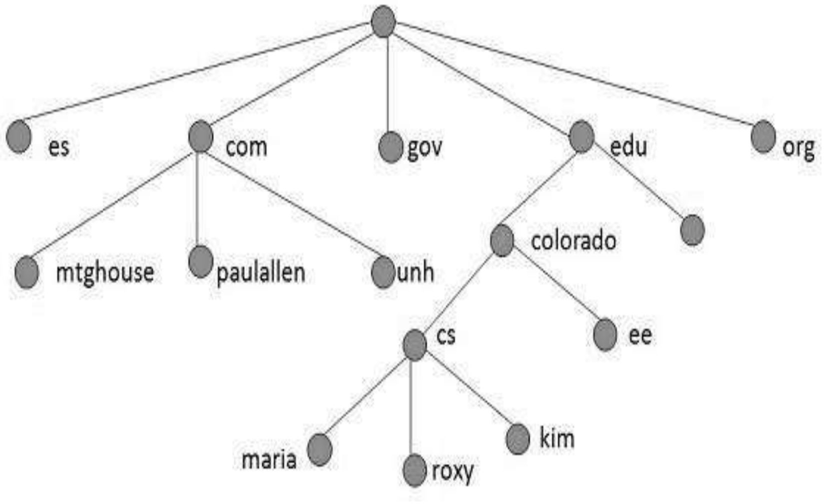


Figure 11.1.Domain Name Space Hierarchy.

iii). Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is notpossible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

- Hierarchy of server is same as hierarchy of names.
- The entire name space is divided into the zones

➢ **Zones**

Zone is collection of nodes (sub domains) under the main domain. The server maintains a database called zone file for every zone. If the domain is not further divided into sub domains then domain and zone refers to the same thing. The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers. The figure 11.2 shows the Zones of Domain Name Space.

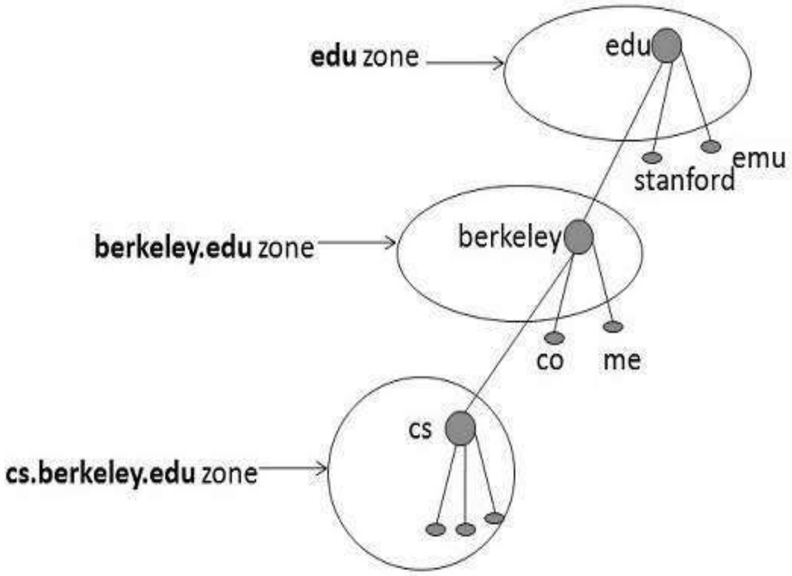


Figure 11.2.The Zones of Domain Name Space

11.2.3 Types of Name Servers

Following are the three categories of Name Servers that manages the entire Domain Name System:

a). Root Server
b). Primary Server
c). Secondary Server

a). Root Server

Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server.

b). Primary Server

Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

c). Secondary Server

Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

11.2.4 DNS Working Procedure

DNS translates the domain name into IP address automatically. Following steps will take you through the steps included in domain resolution process:

- When we type **www.auk.com** into the browser, it asks the local DNS Server for its IP address.
    - Here the local DNS is at ISP end.
- When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.
- The root DNS server replies with delegation that **I do not know the IP address of www.auk.com but know the IP address of DNS Server.**
- The local DNS server then asks the com DNS Server the same question.
- The **com** DNS Server replies the same that it does not know the IP address of www.auk.com but knows the address of auk.com.
- Then the local DNS asks the auk.com DNS server the same question.
- Then auk.com DNS server replies with IP address of www.auk.com.
- Now, the local DNS sends the IP address of www.auk.com to the computer that sends the request.

11.3. Remote Login

The virtual private networks (VPN) were the only way to remotely access work files from home. But VPN access isn't the same as accessing the hard drive of your work computer. VPN gives you access to the local area network (LAN) at your office. With VPN, you're only able to access your PowerPoint presentation files if you've saved them on the network, not just on your computer's hard drive. Remote login, however, uses simple desktop sharing software to give you a "remote control" for accessing your computer and all of its software and hard drive filesfrom any Internet-connected device anywhere in the world. Remote login works exactly the same way as desktop sharing. In desktop sharing, there are two separate parties: the host computer and the remote user. To share a

NOTES

desktop, the host computer allows a remote user to view the contents of the host computer's desktop over the Internet. The host computer can also hand over keyboard and mouse controls to the remote user. With remote log-in, your home or work computer is the host and you (in this case) are the remote user.

Remote login requires three basic components:

i). Software download

ii). Internet connection

iii). Secure desktop sharing network

For remote login to work, both the host computer and all remote users have to download and install the same desktop sharing software. Desktop sharing software typically includes two distinct programs:

i). The desktop sharing client that runs on the host computer.

ii). A viewer program that allows the remote user to view the contents of the host computer's desktop in a resizable window.

Remote login will only work if the host computer is powered on, connected to the Internet and running the desktop sharing software. Each time you open and run the desktop sharing software on the host computer, the software starts a new session. Each session has a particular ID and/or password that are required to remotely log in to the host computer. Once the session has been established, most desktop sharing software quietly runs in the background of the host computer until a remote login request is made. To log in to the host computer from home (or while traveling), you'll need to run your version of the same desktop sharing software and enter in the correct session ID or password. Or some services allow you to log in through a Web site. Once you're logged in, both computers will communicate with each other over a secure desktop sharing network. Access to this network can be free or subscription-based, depending on the service. While connected, you'll have access to keyboard controls, mouse controls, all software and all files on the host machine.

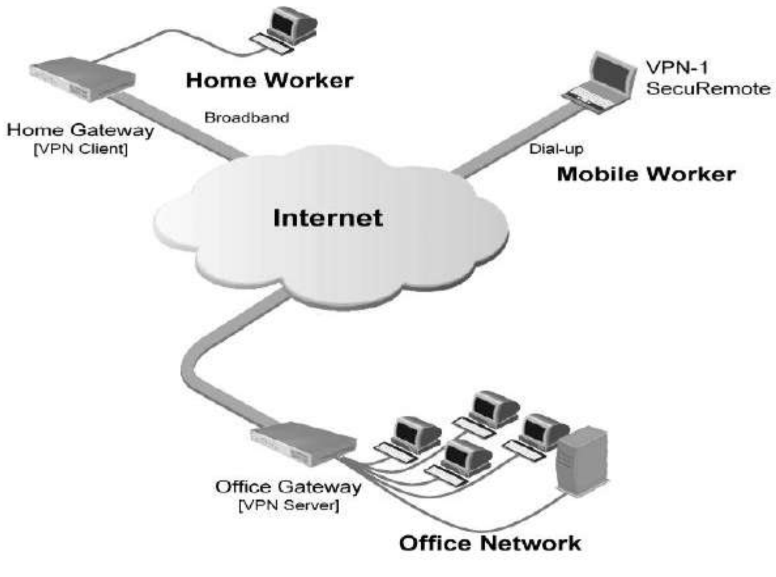


Figure 11.3.Virtual Private Networks for Remote Login

Desktop sharing has many applications.

i). Remote login allows workers to access their work computers from any Internet-enabled device, including home computers, laptops and even PDAs.

ii). Desktop sharing allows for interactive, real-time collaboration between global coworkers.

iii). Presentation sharing turns dry teleconferences into engaging online presentations.

iv). Application sharing lets you test drive software without buying, downloading or installing anything.

Desktop sharing software works by sending packets of information from a host computer to a remote computer describing what's on the host computer's screen at any given time. The encrypted data travels over the Internet. Some data arrives as image files (JPEGs and GIFs), while others arrive as individual pixels assigned to a particular X and Y coordinate. Desktop sharing software is smart enough to only send information updates on the sections of the screen that have changed and to compress the data significantly, minimizing the amount of necessary bandwidth.

For security purposes, all packets of information that are sent over the network are typically encrypted on each end with secure shell (SSH) or 128-bit advanced encryption standard (AES) encoding. For added security, no session IDs or passwords are stored on desktop sharing servers; they're automatically generated by the host machine.

11.4. MAIL EXCHANGE

The electronic mail (email) is a digital mechanism for exchanging messages through Internet or intranet communication platforms. Email messages are relayed through email servers, which are provided by all Internet service providers (ISP). Emails are transmitted between two dedicated server folders: sender and recipient. A sender saves, sends or forwards email messages, whereas a recipient reads or downloads emails by accessing an email server. The email is emerging as one of the most valuable services on the internet today. An e-mail is the ability to compose, send, and receive electronic mail has been around since the early days of the ARPANET and is enormously popular. Many people get dozens of messages a day and consider it their primary way of interacting with the outside world, far outdistancing the telephone and **snail mail** (i.e., paper) letters. e-mail programs are available on virtually every kind of computer these days.

A line of messaging and collaborative software from Microsoft that is comprised of **an email** server, email client, and several groupware applications. The Microsoft **Exchange** line is prevalent in large corporations and is frequently used in conjunction with Microsoft Outlook. A mail exchange record (MX record) is a resource record or settings within the Domain Name System (DNS) that redirects email to a specified mail server that accepts email on behalf of a domain or users. Most of the internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver's side.

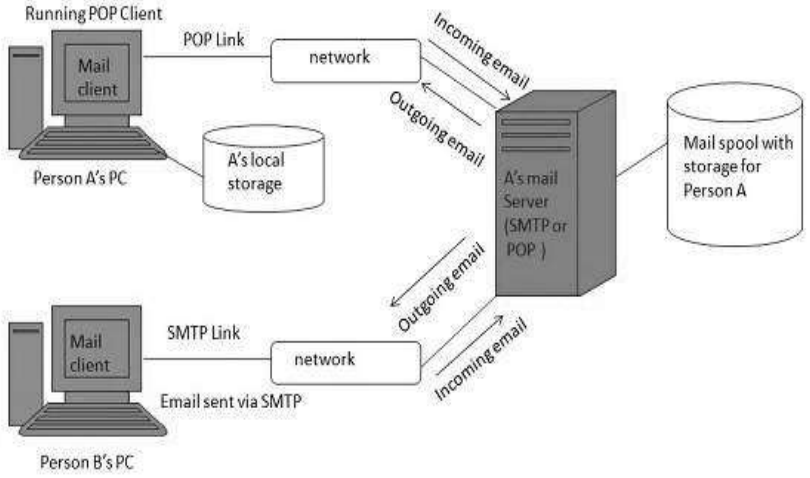


Figure 11.4.Working of e-mail

The E-mail is full of jargon such as BTW (By The Way), ROTFL (Rolling On The Floor Laughing), and IMHO (In My Humble Opinion). Many people also use little ASCII symbols called smileys or emoticons in their e-mail. A few of the more interesting ones are reproduced in table 11.4. For most, rotating the book 90 degrees clockwise will make them clearer. For a minibook giving over 650 smileys, see (Sanderson and Dougherty, 1993).

Table. 11.4. Some smileys.

Smiley	Meaning	Smiley	Meaning
[:)]	Big Smile	[:(]	Shy
[:D]	Cool	[8)]	Shocked
[8D]	Blush	[:O]	Angry
[:I]	Tongue	[:(!]	Dead
[:P]	Evil	[xx(]	Sleepy
[}:)]	Wink	[|)]	Kisses
[;)]	Clown	[:X]	Approve
[:o)]	Black Eye	[^]	Disapprove
[B)]	Eightball	[V]	Question
[8]	Big Smile	[?]	Shy

The first e-mail systems simply consisted of file transfer protocols, with the convention that the first line of each message (i.e., file) contained the recipient's address. As time went on, the limitations of this approach became more obvious.

Some of the complaints were as follows:

i). Sending a message to a group of people was inconvenient. Managers often need this facility to send memos to all their subordinates.

ii). Messages had no internal structure, making computer processing difficult. For example, if a forwarded message was included in the body of another message, extracting the forwarded part from the received message was difficult.

iii). The originator (sender) never knew if a message arrived or not.

iv). If someone was planning to be away on business for several weeks and wanted all incoming e-mail to be handled by his secretary, this was not easy to arrange.

v). The user interface was poorly integrated with the transmission system requiring users first to edit a file, then leave the editor and invoke the file transfer program.

vi). It was not possible to create and send messages containing a mixture of text, drawings, facsimile, and voice.

In 1982, the ARPANET e-mail proposals were published as RFC 821 (transmission protocol) and RFC 822 (message format). Minor revisions, RFC 2821 and RFC 2822, have become Internet standards, but everyone still refers to Internet e-mail as RFC 822. In 1984, CCITT drafted its X.400 recommendation. After two decades of competition, e-mail systems based on RFC 822 are widely used, whereas those based on X.400 have disappeared.

11.4.1 Mail Architecture and Services

The explanation regarding what e-mail systems can do and how they are organized is given in this section. They normally consist of two subsystems: the **user agents**, which allow people to read and send e-mail, and the **message transfer agents**, which move the messages from the source to the destination. The user agents are local programs that provide a command-based, menu-based, or graphical method for interacting with the e-mail system. The message transfer agents are typically system **daemons**, that is, processes that run in the background. Their job is to move e-mail through the system.

Typically, e-mail systems support five basic functions. Let us take a look at them.

a). Composition refers to the process of creating messages and answers. Although any text editor can be used for the body of the message, the system itself can provide assistance with addressing and the numerous header fields attached to each message. For example, when answering a message, the e-mail system can extract the originator's address from the incoming e-mail and automatically insert it into the proper place in the reply.

b). Transfer refers to moving messages from the originator to the recipient. In large part, this requires establishing a connection to the destination or some intermediate machine, outputting the message, and releasing the connection. The e-mail system should do this automatically, without bothering the user.

c). Reporting has to do with telling the originator what happened to the message. Was it delivered? Was it rejected? Was it lost? Numerous applications exist in which confirmation of delivery is important and may even have legal significance.

d). Displaying incoming messages is needed so people can read their e-mail. Sometimes conversion is required or a special viewer must be invoked, for example, if the message is a PostScript file or digitized voice.

e). Disposition is the final step and concerns what the recipient does with the message after receiving it. Possibilities include throwing it away before reading, throwing it away after reading, saving it, and so on. It should also be possible to retrieve and reread saved messages, forward them, or process them in other ways.

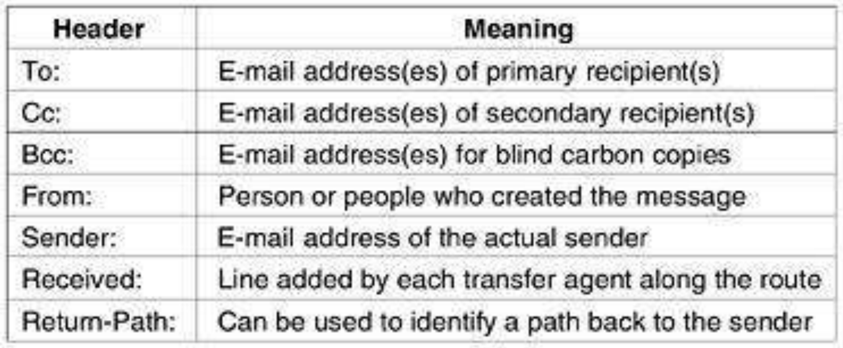
The majority of the systems allow users to create **mailboxes** to store incoming e-mail. The Commands are required to create and destroy mailboxes, inspect the contents of mailboxes, insert and delete messages from mailboxes, and so on. Corporate managers often need to send a message to each of their subordinates, customers, or suppliers. This gives rise to the idea of a **mailing list**, which is a list of e-mail addresses. The primary consideration in e-mail systems is the distinction between the **envelope** and its contents.

The envelope encapsulates the message. It contains all the information needed for transporting the message, such as the destination address, priority, and security level, all of which are distinct from the message itself. The message transport agents use the envelope for routing, just as the post office does. The message inside the envelope consists of two parts: the **header** and the **body**. The header contains control information for the user agents.

11.4.2 Message Formats

The e-mail messages format using RFC 822 standard. The RFC 822 messages consist of a primitive envelope, some number of header fields, a blank line, and then the message body. Each header field (logically) consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value. RFC 822 was designed decades ago and does not clearly distinguish the envelope fields from the header fields. Even though it was revised in RFC 2822, completely redoing it was not possible due to its widespread usage. In normal usage, the user agent builds a message and passes it to the message transfer agent, which then uses some of the header fields to construct the actual envelope, a somewhat old-fashioned mixing of message and envelope. The principal header fields related to message transport are listed in table 11.5.

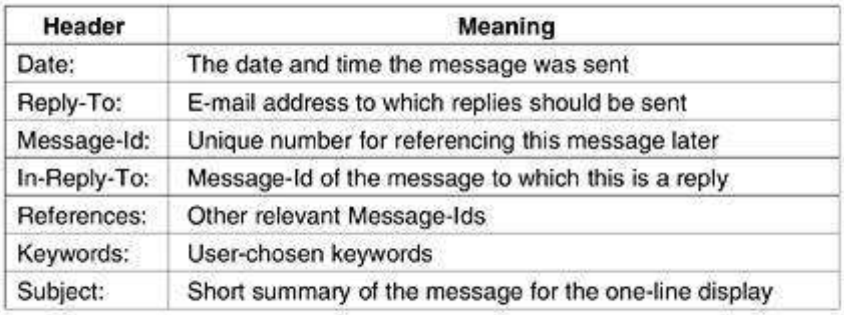
Table. 11.5. RFC 822 Header Fields Related to Message Transport



Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

In addition to the fields of table 11.5, RFC 822 messages may also contain a variety of header fields used by the user agents or human recipients. The most common ones are listed in table 11.6. Most of these are self-explanatory, so we will not go into all of them in detail.

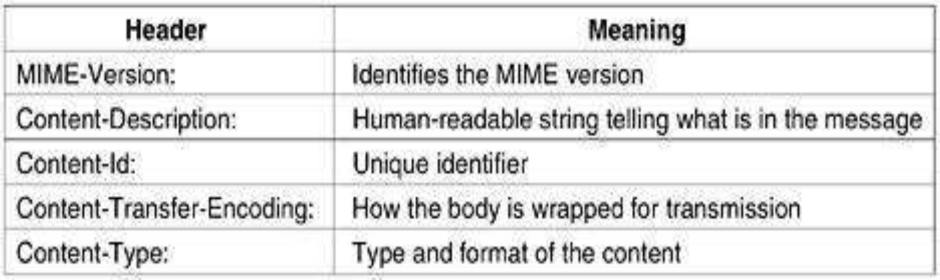
Table. 11.6. Some fields used in the RFC 822 Message Header



Header	Meaning
Date:	The date and time the message was sent
Reply-To:	E-mail address to which replies should be sent
Message-Id:	Unique number for referencing this message later
In-Reply-To:	Message-Id of the message to which this is a reply
References:	Other relevant Message-Ids
Keywords:	User-chosen keywords
Subject:	Short summary of the message for the one-line display

A solution was proposed in RFC 1341 and updated in RFCs 2045–2049. This solution, called **MIME** (**Multipurpose Internet Mail Extensions**) is now widely used. MIME defines five new message headers, as shown in table11.7. The first of these simply tells the user agent receiving the message that it is dealing with a MIME message, and which version of MIME it uses.

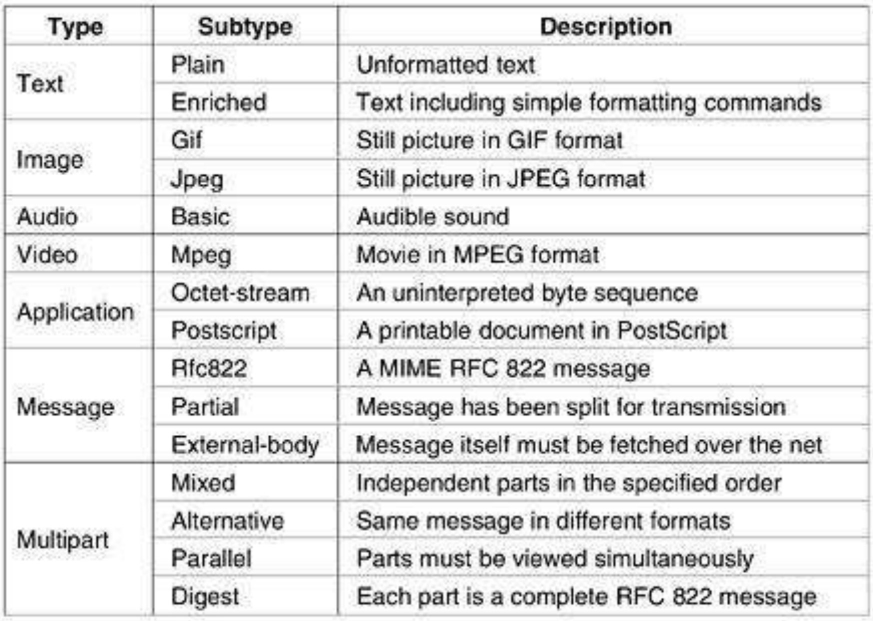
Table 11.7. MIME-The Multipurpose Internet Mail Extensions



Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

The subtype must be given explicitly in the header; no defaults are provided. The initial list of types and subtypes specified in RFC 2045 is given in table11.8.

Figure 11.8.RFC 822 Headers Added by MIME.



Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

11.5. FILE TRANSFER

The **File transfer** is the process of copying or moving a **file** from one computer to another over a **network** or Internet connection. It enables sharing, **transferring** or transmitting a **file** or a logical data object between different users and/or computers both locally and remotely. A file transfer can be an upload or download.If a node is receiving the file, it is called downloading, whereas if it is sending the file to another node or server, then it is uploading. File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), BitTorrent and Simple File Transfer Protocol are the most common file transfer protocols used in computer networks and online.There are two main types of file transfer:

i). Pull-Based: The file transfer request is initiated by the receiver.
ii). Push Based: The file transfer request is initiated by the sender.

Moreover, other than network or Internet, file transfer can be performed manually by copying a file to a new folder or drive in the same computer or by copying it on a USB pen drive, CD or other portable storage device to be transferred to another computer.

A **file transfer protocol** is a convention that describes how to transfer files between two computing endpoints. As well as the stream of bits from a file stored as a single unit in a file system, some may also send relevant metadata such as the filename, file size and timestamp - and even file system permissions and file attributes.

Some examples:

i). FTP is an older cross-platform file transfer protocol.
ii). SSH File Transfer Protocol a file transfer protocol secured by the Secure Shell (SSH) protocol.
iii). Secure copy (SCP) is based on the Secure Shell (SSH) protocol.
iv). HTTP can support file transfer.
v). Bittorent, Gnutella and other distributed file transfers systems use peer-to-peer.
vi). In IBM Systems Network Architecture, LU 6.2 Connect:Direct and XCOM Data Transport are traditional.
vii). Many instant messaging or LAN messenger systems support the ability to transfer files.
viii). Computers may transfer files to peripheral devices such as USB flash drives.
ix). Dial-up modems null modem links used XMODEM, YMODEM, ZMODEM and similar.

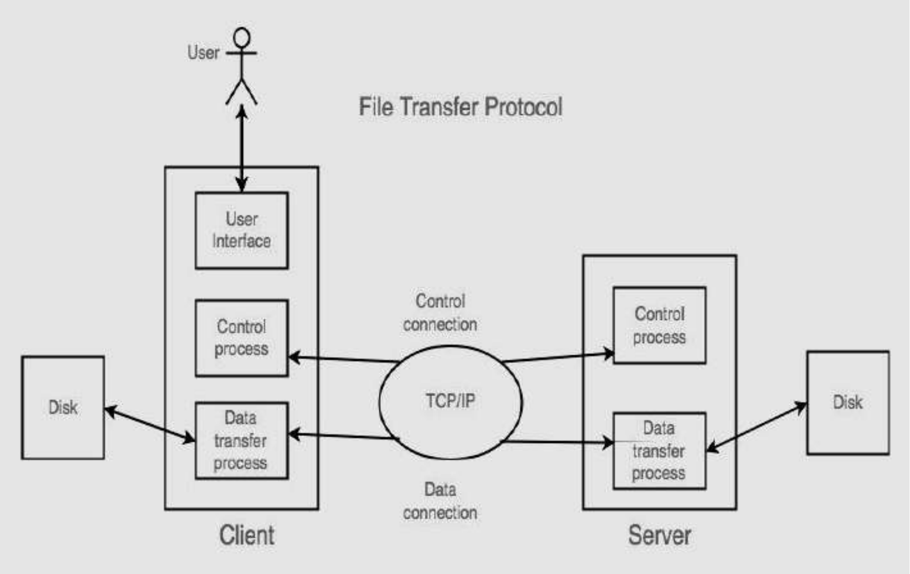


Figure 11.5.File Transfer Protocol

11.6 CHECK YOUR PROGRESS QUESTIONS

1. State about Domain Name System.
2. Define Uniform Resource Locator .
3. What is Zone?
4. Define file transfer protocol.

11.7 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. **Domain Name System** helps to resolve the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names.

2. Uniform Resource Locator (URL) refers to a web address which uniquely identifies a document over the internet. This document can be a web page, image, audio, video or anything else present on the web.

3. **Zone** is collection of nodes (sub domains) under the main domain. The server maintains a database called zone file for every zone. If the domain is not further divided into sub domains then domain and zone refers to the same thing. The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

4. A **file transfer protocol** is a convention that describes how to transfer files between two computing endpoints. In addition to that the stream of bits from a file stored as a single unit in a file system, some may also send relevant metadata such as the filename, file size and timestamp and even file system permissions and file attributes.

11.8 SUMMARY

Naming in the Internet uses a hierarchical scheme called the Domain Name System (DNS). At the top level are the well-known generic domains, including com and edu as well as about 200 country domains. NS is implemented as a distributed database system with servers all over the world. DNS holds records with IP addresses, mail exchanges, and other information. By querying a DNS server, a process can map an Internet domain name onto the IP address used to communicate with that domain. E-mail is one of the two killer apps for the Internet. Everyone from small children to grandparents now use it. Most e-mail systems in the world use the mail system now defined in RFCs 2821 and 2822. Messages sent in this system use system ASCII headers to define message properties. Many kinds of content can be sent using MIME. Messages are sent using SMTP, which works by making a TCP connection from the source host to the destination host and directly delivering the e-mail over the TCP connection.

11.9 KEY WORDS

- **IP Address:** is a unique logical address assigned to a machine over the network.
- **Email:** Electronic mail is a digital mechanism for exchanging messages through Internet or intranet communication platforms.Email messages are relayed through email servers, which are provided by all Internet service providers (ISP). Emails are transmitted between two dedicated server folders: sender and recipient.
- **File transfer** is the process of copying or moving a file from one computer to another over a network or Internet connection. It enables sharing, transferring or transmitting a file or a logical data object between different users and/or computers both locally and remotely.

11.10 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. Define IP address.
2. What is meant by Message Format?
3. State about Name Server.
4. Differentiate Absolute URL and Relative URL.

Long Answer Questions

1. Illustrate about DNS Architecture
2. Discuss about Mail Exchange.

11.11 FURTHER READINGS

1. IEEE Internet Computing, Jan.-Feb. 2000.
2. Garber, "Will 3G Really Be the Next Big Wireless Technology?"
3. Computer Networks, Andrew.S.Tanenbaum, Pearson Education,3rd Edn 2010.
4. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.
5. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.

NOTES

UNIT-12: REMOTE PROCEDURE CALL

Structure

12.0 INTRODUCTION

In this unit, you will gain knowledge about remote procedure call of transport layer. Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details. A procedure call is also sometimes known as a function call or a subroutine call. In RPC, the sending computer makes a request in the form of a procedure, function, or method call. RPC translates these calls into requests and sends them over the network to the intended destination. The RPC recipient then processes the request based on the procedure name and argument list, and sends a response to the sender when complete. Also gain knowledge about WWW, HTTP, SNMP.

12.1 OBJECTIVES

After going through this unit, you will describe about

- Remote Procedure Call
- Remote File Access
- WWW and HTTP
- Simple Network Management Protocol.

12.2 REMOTE PROCEDURE CALL

In 1984, Birrell and Nelson have suggested an idea to allow programs to call procedures located on remote hosts. When a process on machine 1 calls a procedure on machine 2, the calling process on 1 is suspended and execution of the called procedure takes place on 2. Information can be transported from the caller to the callee in the parameters and can come back in the procedure result. No message passing is visible to the programmer. This technique is known as **Remote Procedure Call (RPC)** and has become the basis for many networking applications. Traditionally, the calling procedure is known as the client and the called procedure is known as the server.In the simplest form, to call a remote procedure, the client program must be bound with a small library procedure, called the **client stub**, that represents the server procedure in the client's address space. Similarly, the server is bound with a procedure called the **server stub**.

Remote Procedure Call (**RPC**) is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details. A **procedure call** is also sometimes known as a function **call** or a subroutine **call**. RPC uses the client-server model. The requesting program is a client and the service providing program is the server. Like a regular or local procedure call, an RPC is a synchronous operation requiring the requesting program to be suspended until the results of the remote procedure are returned. However, the use of lightweight processes or threads that share the same address space allows multiple RPCs to be performed concurrently.

In a certain sense, sending a message to a remote host and getting a reply back is a lot like making a function call in a programming language. In both cases you start with one or more parameters and you get back a result. This observation has led people to try to arrange request-reply interactions on networks to be cast in the form of procedure calls. Such an arrangement makes network applications much easier to program and more familiar to deal with. For example, just imagine a procedure named *get_IP_address* (*host_name*) that works by sending a UDP packet to a DNS server and waiting for the reply, timing out and trying again if one is not forthcoming quickly enough. In this way, all the details of networking can be hidden from the programmer.

The idea behind RPC is to make a remote procedure call look as much as possible like a local one. In the simplest form, to call a remote procedure, the client program must be bound with a small library procedure, called the **client stub**, that represents the server procedure in the client's address space. Similarly, the server is bound with a procedure called the **server stub**. These procedures hide the fact that the procedure call from the client to the server is not local.

The actual steps in making an RPC are shown in fig 12.1 Step 1 is the

NOTES

client calling the client stub. This call is a local procedure call, with the parameters pushed onto the stack in the normal way. Step 2 is the client stub packing the parameters into a message and making a system call to send the message. Packing the parameters is called **marshalling**. Step 3 is the kernel sending the message from the client machine to the server machine. Step 4 is the kernel passing the incoming packet to the server stub. Finally, step 5 is the server stub calling the server procedure with the unmarshaled parameters. The reply traces the same path in the other direction.

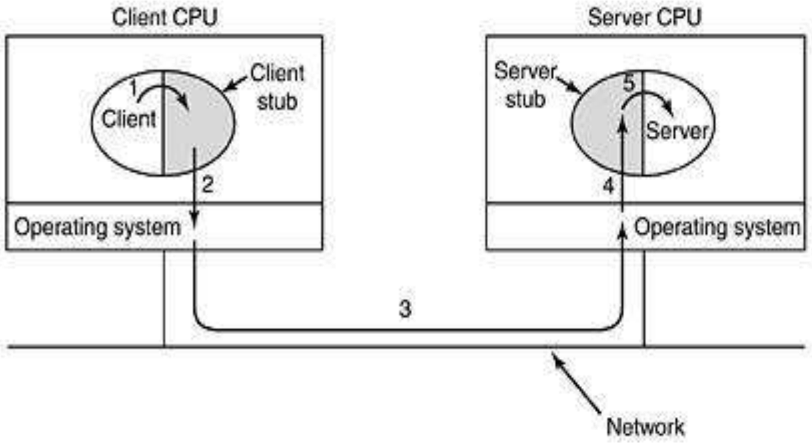


Figure 12.1.Steps in making a remote procedure call. The stubs are shaded.

The key item to note here is that the client procedure, written by the user, just makes a normal (i.e., local) procedure call to the client stub, which has the same name as the server procedure. Since the client procedure and client stub are in the same address space, the parameters are passed in the usual way. Similarly, the server procedure is called by a procedure in its address space with the parameters it expects. To the server procedure, nothing is unusual. In this way, instead of I/O being done on sockets, network communication is done by faking a normal procedure call.

12.2.1 RPC Message Procedure

When program statements that use RPC framework are compiled into an executable program, a stub is included in the compiled code that acts as the representative of the remote procedure code. When the program is run and the procedure call is issued, the stub receives the request and forwards it to a client runtime program in the local computer. The client runtime program has the knowledge of how to address the remote computer and server application and sends the message across the network that requests the remote procedure. Similarly, the server includes a runtime program and stub that interface with the remote procedure itself. Response-request protocols are returned the same way.

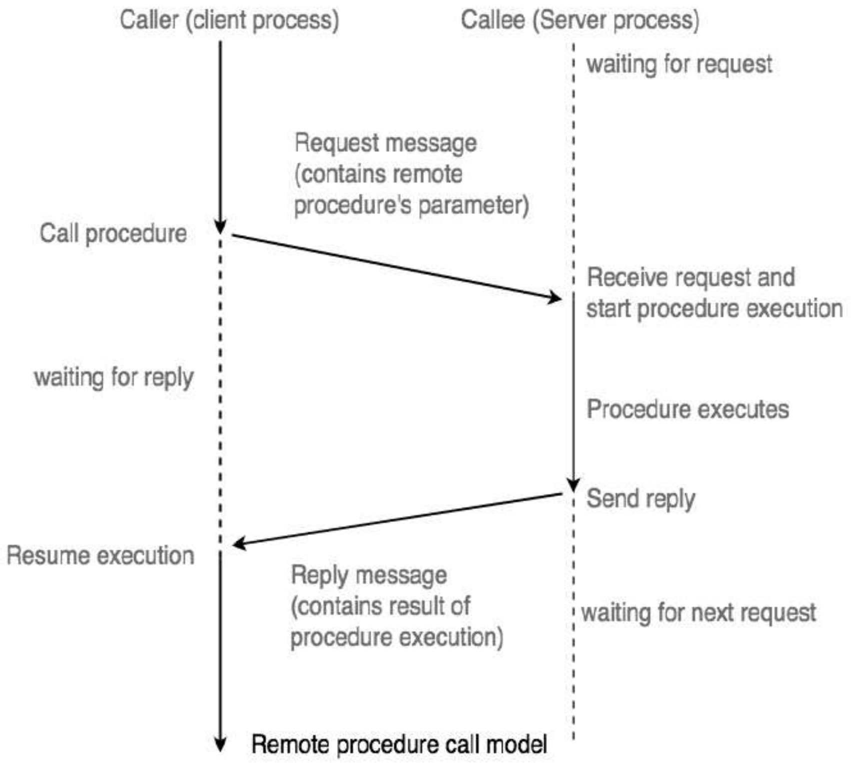

Caller (client process)
Callee (Server process)
waiting for request
Request message (contains remote procedure's parameter)
Call procedure
Receive request and start procedure execution
waiting for reply
Procedure executes
Send reply
Resume execution
Reply message (contains result of procedure execution)
waiting for next request
Remote procedure call model

Figure 12.2.Remote Procedure Call

12.3. REMOTE FILE ACCESS

Remote access is the ability to **access** a computer or a network **remotely** through a network connection. **Remote access** enables users to **access** the systems they need when they are not physically able to connect directly; in other words, users **access** systems **remotely** by using a telecommunications or internet connection. In computer networking, remote access technology allows someone to log in to a system as an authorized user without being physically present at its keyboard. Remote access is commonly used on corporate computer networks but can also be used on home networks.

Remote access can be set up using a local area network (LAN), wide area network (WAN) or even a virtual private network (VPN) so that resources and systems can be accessed remotely. Another method for performing remote access is by establishing a virtual private network (VPN), a network that usually uses the Internet to connect remote sites and users together. This type of network uses encryption and tunneling to access a company's network. This can be a great choice for a relatively small organization. Other means of establishing remote access include the use of an integrated services digital network, wireless network, cable modem or digital subscriber line.

NOTES

Remote access means allowing people to access your business computer systems even when not directly connected to your company network. For instance:

i). Allowing staff to log in to your customer database from home.
ii). Setting up a project workspace where clients can share and view files.
iii). Allowing employees to send and receive email from any computer.

12.4 WWW and HTTP

12.4.1 World Wide Web (WWW)

The World Wide Web (WWW) is a network of online content that is formatted in HTML and accessed via HTTP. The term refers to all the interlinked HTML pages that can be accessed over the Internet. Researcher Tim Berners-Lee led the development of the World Wide Web in the late 1980s and early 1990s.The World Wide Web was originally designed in 1991at CERN. The World Wide Web is most often referred to simply as "the Web." The World Wide Web (WWW) is combination of all resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP). A broader definition comes from the World Wide Web Consortium (W3C): "The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge”. The figure 12.3 briefly describes the evolution of World Wide Web.

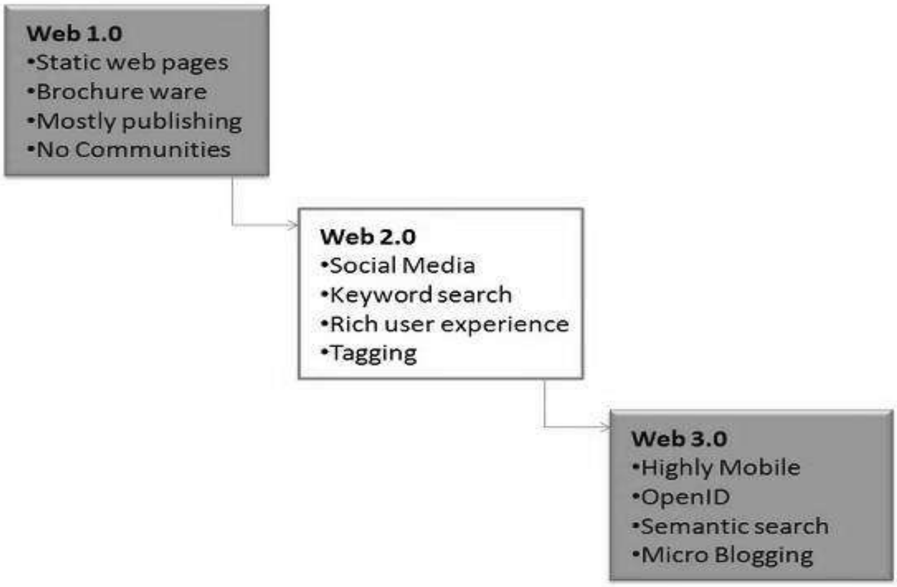


Figure 12.3 Evolution of World Wide Web

The World Wide Web is what most people think of as the Internet. It is all the Web pages, pictures, videos and other online content that can be accessed via a Web browser. The Internet, in contrast, is the underlying network connection that allows us to send email and access the World Wide Web. The early Web was a collection of text-based sites hosted by organizations that were technically gifted enough to set up a Web server and learn HTML. It has continued to evolve since the original design, and it now includes interactive (social) media and user-generated content that requires little to no technical skills. All major websites have adjusted their content design and development approach to accommodate the rapidly increasing fraction of the population accessing the web from small-screen phones instead of large-screen desktop and laptop computers.

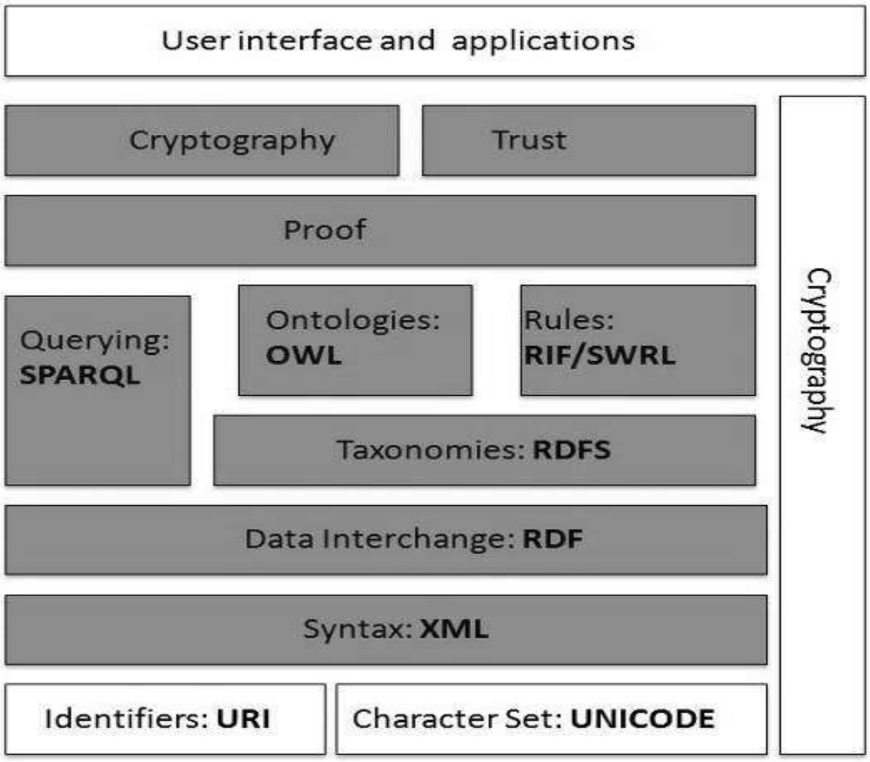


Figure 12.4. Architecture of WWW

A uniform resource locator (URL) is the address of a resource on the Internet. A URL indicates the location of a resource as well as the protocol used to access it. A URL contains the following information:

i). The protocol used to a access the resource
ii). The location of the server (whether by IP address or domain name)
iii). The port number on the server (optional)
iv). The location of the resource in the directory structure of the server
v). A fragment identifier (optional)

Further more known as a Universal Resource Locator (URL) or Web address. A URL is a type of uniform resource identifier (URI). In common practice, the term URI isn't used, or is used synonymously with

NOTES

URL, even though this is technically incorrect.Tim Berners-Lee and the Internet Engineering Task Force working group is credited with developing the URL in 1994. It is formally specified in RFC 1738.

All URLs are presented in the following order:

i). Scheme name
ii). Colon and two slashes
iii). Location of the server
iv). The port (optional) and the location of the resource on the server
v). Fragment identifier (optional)

So, the format will look like this:

scheme://location:port/file-on-server.htm?querystring=1

This looks more complex that it is. The most common schemes (protocols) are HTTP and HTTPS, which any WWW user will recognize. The location of the server is generally a domain name. Given this, the following URLs are much more simple to understand:

http://www.google.com/default.htm
https://www.google.com/default.htm

Both these URLs indicate that there is a file named default.htm on a server with the address of "google.com". One uses regular HTTP, while the other uses a secure version of this scheme.

Operation of WWW

The operation of World Wide Web (WWW) is offered in figure 12.5.

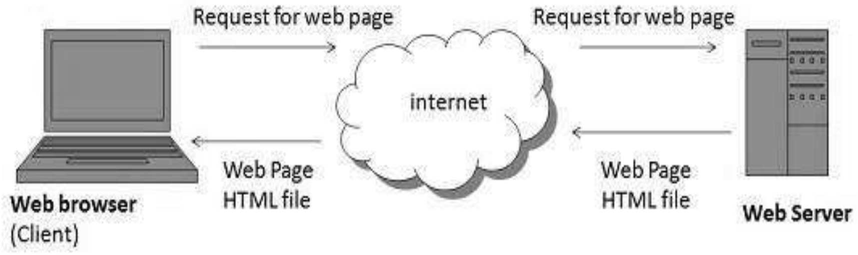


Figure 12.5. Operation of WWW

The WWW works on client- server approach. Following steps clarify how the web works:

1. User enters the URL (say, http://www.auk.com) of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to www.tutorialspoint.com.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.

4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, It interprets it and display the contents of web page in web browser's window.

Two common components of confusion about URLs:

i). The "www" is not actually part of the technical protocol. Websites just started using this to indicate the user is using the World Wide Web. This is why if you go to http://google.com, it redirects to http://www.google.com.

ii). Most users access the Internet via a Web browser, which inserts port 80 on HTTP connections behind the scenes. This is why if you go to http://www.google.com:80, you will see the same website as if there were no port number.

Finally, the following URL demonstrates a fragment identifier, more commonly known as a query string:

http://www.google.com/some-page?search=hello

This is saying that to use the HTTP protocol to send a request to the website at google.com (over port 80) and to ask for "some-page" and send in the search variable "hello". This is why you'll sometimes see an extremely long URL as many variables are being sent to the Web server in more interactive Web applications.

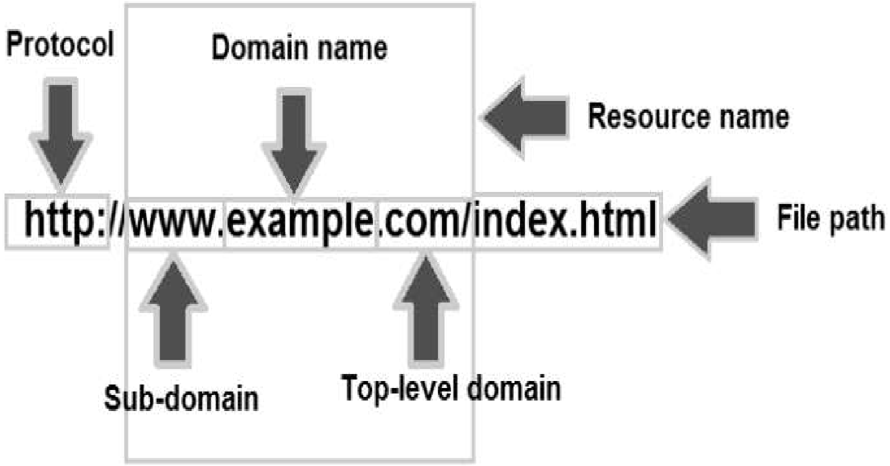


Figure 12.6.Components of URL

NOTES

The difference between the **Internet** and **WWW** are presented in the table 12.1.

Table 12.1 Difference between Internet and WWW

S.NO.	INTERNET	WWW
1.	Internet originated sometimes in late 1960s.	English scientist Tim Berners-Lee invented the World Wide Web in 1989
2.	Nature of Internet is hardware.	Nature of www is software.
3.	Internet consists of computers, routers, cables, bridges, servers, cellular towers, satellites etc.	www consists of information like text, images, audio, video
4.	The first version of the Internet was known as ARPANET	In the beginning WWW was known as NSFNET
5.	Internet works on the basis of Internet Protocol (IP)	WWW works on the basis of Hyper Text Transfer Protocol (HTTP)
6.	Internet is independent of WWW	WWW requires the Internet to exist
7.	Internet is superset of WWW	WWW is a subset of the Internet. Apart from supporting www, the Internet's hardware infrastructure is used for other things as well (e.g. FTP, SMTP)
8.	Computing devices are identified by IP Addresses	Information pieces are identified by Uniform Resource Locator (URL)

12.4.2 Hyper Text Transfer Protocol (HTTP)

Hyper Text Transfer Protocol (HTTP) is an application-layer protocol used primarily on the World Wide Web. HTTP uses a client-server model where the web browser is the client and communicates with the web server that hosts the website. The browser uses HTTP, which is carried over TCP/IP to communicate to the server and retrieve Web content for the user.HTTP is a widely used protocol and has been rapidly adopted over the Internet because of its simplicity. It is a stateless and connectionless protocol.HTTP means Hyper Text Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.

Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other. HTTP specification specifies how clients' request data will be constructed and sent to the server, and how the servers respond to these requests.

a). Basic Features

There are three basic features that make HTTP a simple but powerful protocol:

i). **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.

ii). **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.

iii). **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request.

Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

HTTP/1.0 uses a new connection for each request/response exchange, where as HTTP/1.1 connection may be used for one or more request/response exchanges.

b). Basic Architecture

The following figure 12.7 shows a very basic architecture of a web application and depicts where HTTP sits:

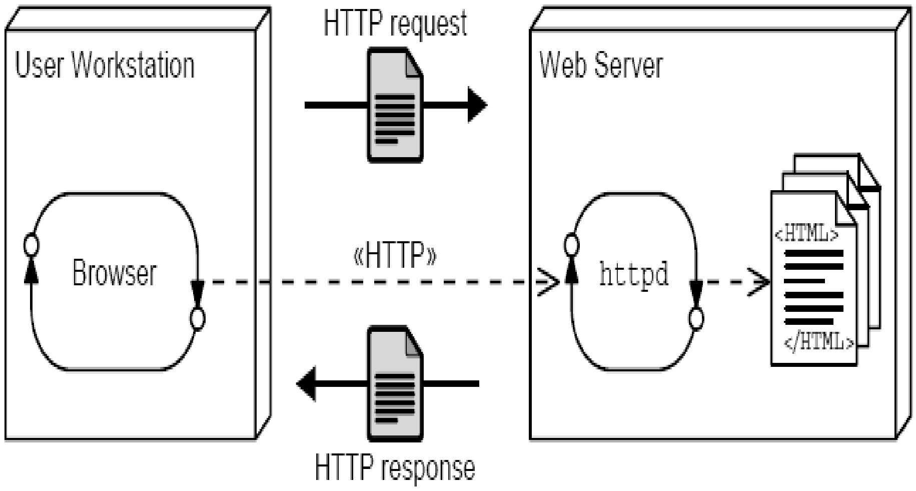


Figure12.7 Architecture of a Web Application and HTTP

The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

- **Client**

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

- **Server**

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

12.5. SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

The Simple Network Management Protocol (SNMP) is an application–layer protocol defined by the Internet Architecture Board (IAB) in RFC1157 for exchanging management information between network devices. It is a part of Transmission Control Protocol or Internet Protocol (TCP⁄IP) protocol suite.

SNMP is one of the widely accepted protocols to manage and monitor network elements. Most of the professional–grade network elements come with bundled SNMP agent. These agents have to be enabled and configured to communicate with the network management system (NMS). SNMP basic components and their functionalities are mentioned below. SNMP consists of

i). SNMP Manager
ii). Managed devices
iii). SNMP agent
iv). Management Information Database Otherwise called as Management Information Base (MIB)

The **S**imple Network Management Protocol (**SNMP) i**s used to manage network devices by setting value for certain attribute and monitor network devices by polling necessary metrics from the device. SNMP comprises simple Client-Server Architecture. The SNMP client running on your Network management solution will be responsible for polling data or setting data. And the SNMP server running on your actual device will respond to SNMP client’s call. The SNMP Agent will not be turned on in network devices by default. The network admin has to enable SNMP if needed.

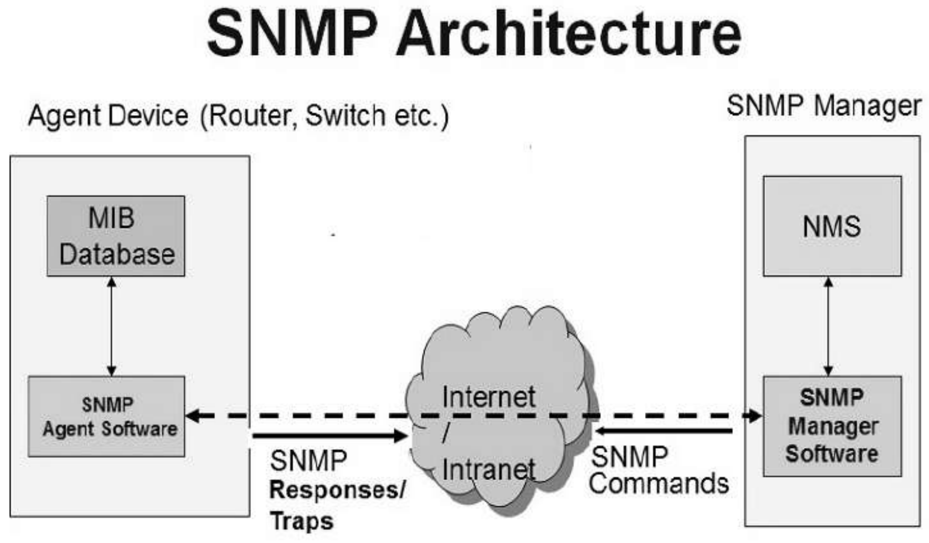


Figure 12.8Architecture of SNMP

- **SNMP Manager:**

A manager or management system is a separate entity that is responsible to communicate with the SNMP agent implemented network devices. This is typically a computer that is used to run one or more network management systems.

➢ **SNMP Manager's key functions**

i). Queries agents
ii). Gets responses from agents
iii). Sets variables in agents
iv). Acknowledges asynchronous events from agents

➢ **Managed Devices:**

A managed device or the network element is a part of the network that requires some form of monitoring and management e.g. routers, switches, servers, workstations, printers, UPSs, etc...

➢ **SNMP Agent:**

The agent is a program that is packaged within the network element. Enabling the agent allows it to collect the management information database from the device locally and makes it available to the SNMP manager, when it is queried for. These agents could be standard (e.g. Net-SNMP) or specific to a vendor (e.g. HP insight agent)

➢ **SNMP agent's key functions:**

i). Collects management information about its local environment
ii). Stores and retrieves management information as defined in the MIB.
iii). Signals an event to the manager.
iv). Acts as a proxy for some non–SNMP manageable network node.

It is necessary to understand SNMP MIB and SNMP OIDs to use SNMP and poll the metrics that we need.

➢ **Understanding SNMP MIB and OIDs:**

MIB stands for **Management Information Base** and is a collection of definitions that define the properties of the managed object within the device to be managed. MIB files are written in an independent format and the object information they contain is organized hierarchically. The various pieces of information can be accessed by SNMP.

OIDs or **Object Identifiers** uniquely identify managed objects in the MIB. Generally, an OID is a long sequence of numbers, coding the nodes, separated by dots. Here is a sample structure of an OID:

For example: To get system up time of a managed device , you can poll this OID -1.3.6.1.6.3.10.2.1.3 and it will return a the number of seconds since the SNMP engine last. So OID is to uniquely identify a certain metric and MIB contains tree of OIDs based on the feature and organization of the manufacturer.

➢ Typical SNMP communication

Being the part of TCP⁄ IP protocol suite, the SNMP messages are wrapped as User Datagram Protocol (UDP) and intern wrapped and transmitted in the Internet Protocol. The following diagram will illustrate the four–layer model developed by Department of Defense (DoD).

i). GET⁄GET NEXT⁄GET BULK⁄SET

ii). TRAP

iii). INFORM

By default the SNMP port is 161 and TRAP⁄ INFORM uses SNMP port 162 for communication.

➢ SNMP versions

Since the inception SNMP, has gone through significant upgrades. However SNMP Protocol v1 and v2c are the most implemented versions of SNMP. Support to SNMP Protocol v3 has recently started catching up as it is more secured when compare to its older versions, but still it has not reached considerable market share.

i). **SNMPv**1:
This is the first version of SNMP protocol, which is defined in RFCs 1155 and 1157.

ii). SNMPv2c:
This is the revised protocol, which includes enhancements of SNMPv1 in the areas of protocol packet types, transport mappings, MIB structure elements but using the existing SNMPv1 administration structure ("community based" and hence SNMPv2c). It is defined in RFC 1901, RFC 1905, RFC 1906, RFC 2578.

iii). SNMPv3:
SNMPv3 defines the secure version of the SNMP. SNMPv3 protocol also facilitates remote configuration of the SNMP entities. It is defined by RFC 1905, RFC 1906, RFC 3411, RFC 3412, RFC 3414, RFC 3415.

Though each version had matured towards rich functionalities, additional emphasis was given to the security aspect on each upgrade. Here is a small clip on each editions security aspect.

Table. 12.2. SNMP Versions

SNMP v1	Community–based security
SNMP v2c	Community–based security
SNMP v2u	User–based security
SNMP v2	Party–based security
SNMP v3	User–based security

- **SNMP Operations:**

Pulling the data from Managed devices:

i). **GetRequest**- To get values for particular OID
ii). **SetRequest**- To set values on particular OID
iii). **GetNextRequest**- To get values from next OID
iv). **GetBulkRequest**- To get values of the MIB tree in bulk

Pushing data from Managed devices to SNMP server:

Traps- Event traps Will be sent from a network device to trap server in case any event occurred in Network device ie: Interface down, VPN down and etc. The trap server location and credentials have to be configured in each network devices supposed to be monitored.

- **Prometheus and What it does:**

Prometheus is a Time series Database, where the information changing as time moves on can be stored efficiently, queried in a tailored manner, and retrieved quickly than ever.

- **Prime features of Prometheus :**

i). A multi-dimensional data model with time series data identified by metric name and key/value pairs
ii). A flexible query language to leverage this dimensionality
iii). Time series collection happens via a pull model over HTTP
iv). Multiple modes of graphing and exposed API to get Time series data.

Let us start with Prometheus installing and we will cover few of advantage using Prometheus TSDB(Time Series Database)

- **Exporter:** An exporter is a library, that collects data from a source and transforms it into a format that will be accepted by my Prometheus server.

- **SNMP Exporter:** An SNMP Exporter is a tool which collects data from the managed device and exposes it in a format that will be accepted by Prometheus server. SNMP Exporter is Opensource and you can get it from here and run it by./snmp_exporter

The SNMP Exporter reads a config file "snmp.yml" by default and configuration contains the OIDs to walk/get from device and credentials to use in case if it is SNMP v2 or SNMP v3.

The snmp.yml configuration file is not intended to be handwritten, as there will be a large number of OIDs be specified in the configuration and it is complex to name and labeling the metrics. So we can use a generator to generate the snmp..ymlconfiguration.Thisconfig generator uses NetSNMP to parse MIBs, and generates configs for the snmp_exporter using them.

12.6 CHECK YOUR PROGRESS QUESTIONS

1. Mention the role of Remote Procedure Call.
2. Define Hypertext Transfer Protocol.

12.7 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Remote Procedure Call (RPC) is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details. A procedure call is also sometimes known as a function call or a subroutine call.

2. The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990.

12.8 SUMMARY

Remote Procedure Call (RPC) is an example of a higher-level mechanism built on top of the TCP. RPC involves making a call to a procedure that is in a different process space to that of the calling procedure.A Remote Procedure Call (RPC) provides a standard way to invoke services on remote servers that are linked by a network. The

Remote Procedure Call (RPC) mechanism allows two networked computers to communicate with each other using a common protocol. RPC is the basis for some widely used networking protocols, including the Network File System (NFS) and the Network Information Service (NIS). In addition, the daemons used in HP's TruCluster product are heavy users of the RPC mechanism. The RPC file contains program definitions. It is an EDIT file. The definitions include the program name, program number, and possible aliases for the program name. The combination of host address, program number, and procedure number specifies one remote service procedure.

12.9 KEY WORDS

- **Remote File Access:** Remote access is the ability to access a computer or a network remotely through a network connection. Remote access enables users to access the systems they need when they are not physically able to connect directly; in other words, user's access systems remotely by using a telecommunications or internet connection.

- **WWW :** The World Wide Web is a network of online content that is formatted in HTML and accessed via HTTP. The term refers to all the interlinked HTML pages that can be accessed over the Internet.

- **HTTP:** Hyper Text Transfer Protocol (HTTP) is an application-layer protocol used primarily on the World Wide Web. HTTP uses a client-server model where the web browser is the client and communicates with the web server that hosts the website.

- **VPN:** A virtual private network is utilizing the Internet to connect remote sites and users together. This type of network uses encryption and tunneling to access a company's network.

- **SNMP:**TheSimple Network Management Protocol **i**s used to manage network devices by setting value for certain attribute and monitor network devices by polling necessary metrics from the device.SNMP comprises simple Client-Server Architecture

12.10 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. What is mail exchange?
2. State the significance of SNMP exporter.

Long Answer Questions

1. Explain about the architecture of HTTP.
2. Discuss about the architecture and functions of SNMP .

12.11 FURTHER READINGS

1. IEEE Internet Computing, Jan.-Feb. 2000
2. Comer, The Internet Book
3. Garber, "Will 3G Really Be the Next Big Wireless Technology?"
4. Computer Networks, 3rd Edition, Andrew.S.Tanenbaum, Pearson Education, 2010.
5. Data and Computer Communications, 8th Edition, William Stallings, Prentice Hall.
6. An Engineering Approach to Computer Networks, 2nd Edition, S.Keshav, Pearson Education, 2008.
7. Behrouz A. Forouzan, Data Communications and Networking, Third Edition, TataMcGraw Hill, 2003.

BLOCK 5: NETWORK SECURITY

UNIT 13: INTRODUCTION TONETWORK SECURITY

Structure

13.0 INTRODUCTION

Within this unit, we will find out about cryptography, plaintext, cipher text, secrecy, authentication, non-repudiation, and integrity, cryptanalysis, cryptology. Also this unit describes about Encryption Model, Transposition and Substitution Ciphers and Cryptographic principles. Cryptography means science of secret writing. Encryption Algorithm is a mathematical process that produces a cipher text for any given plaintext and encryption key.

13.1 OBJECTIVES

After referring this unit, you will illustrate about

- Network security principles and model.
- Understand the network security mechanisms such as Transposition Ciphers, Substitution Ciphers.
- Cryptographic principles.

13.2 CRYPTOGRAPHY

Cryptography is defined as science of "secret writing." A **cipher** is a character-for-character or bit-for-bit transformation, without regard to the linguistic structure of the message. In contrast, a **code** replaces one word with another word or symbol. Codes are not used any more, although they have a glorious history. The most successful code ever devised was used by the U.S. armed forces during World War II in the Pacific. Network security problems can be divided roughly into four closely intertwined areas: **secrecy, authentication, non-repudiation, and integrity control**. Secrecy, also called confidentiality, has to do with keeping information out of the hands of unauthorized users.

A security service is a processing or communicating service that can prevent or detect the above- mentioned attacks. Various security servicesare:

i). Authentication: the recipient should be able to identify the sender, and verify that the sender, who claims to be the sender, actually did send the message.
ii). Data Confidentiality: An attacker should not be able to read the transmitted data or extract data in case of encrypted data. In short, confidentiality is the protection of transmitted data from passive attacks.
iii). Data Integrity: Make sure that the message received was exactly the message the sender sent.
iv). Non-repudiation: The sender should not be able to deny sending and should not be able to deny receiving the message.

Evolution of Cryptography

It is during and after the European Renaissance, various Italian and Papal states led the rapid proliferation of cryptographic techniques. Various analysis and attack techniques were researched in this era to break the secret codes.

- Improved coding techniques such as **Vigenere Coding** came into existence in the 15th century, which offered moving letters in the message with a number of variable places instead of moving them the same number of places.
- Only after the 19th century, cryptography evolved from the ad hoc approaches to encryption to the more sophisticated art and science of information security.
- In the early 20th century, the invention of mechanical and electromechanical machines, such as the **Enigma rotor machine,** provided more advanced and efficient means of coding the information.
- During the period of World War II, both cryptography and cryptanalysis became excessively mathematical.

With the advances taking place in this field, government organizations, military units, and some corporate houses started adopting the applications of cryptography. They used cryptography to guard their secrets from others. Now, the arrival of computers and the Internet has brought effective cryptography within the reach of common people.

13.3 ENCRYPTION MODEL

Encryption is the process of using an algorithm to transform information to make it unreadable for unauthorized users. This cryptographic method protects sensitive data such as credit card numbers by encoding and transforming information into unreadable cipher text. This encoded data may only be decrypted or made readable with a key. Symmetric-key and asymmetric-key are the two primary types of encryption. Encryption is essential for ensured and trusted delivery of sensitive information.

The messages to be encrypted, known as the **plaintext**, are transformed by a function that is parameterized by a **key**. The output of the encryption process, known as the **cipher text**, is then transmitted, often by messenger or radio. We assume that the enemy, or **intruder**, hears and accurately copies down the complete cipher text. However, unlike the intended recipient, he does not know what the decryption key is and so cannot decrypt the cipher text easily. Sometimes the intruder can not only listen to the communication channel (passive intruder) but can also record messages and play them back later, inject his own messages, or modify legitimate messages before they get to the receiver (active intruder). The art of breaking ciphers, called **cryptanalysis**, and the art of devising cryptography is collectively identified as **cryptology**.

It will often be useful to have a notation for relating plaintext, cipher text, and keys. We will use $C = E_K(P)$ to mean that the encryption of the plaintext P using key K gives the cipher text$C$. Similarly, $P = D_K(C)$ represents the decryption of C to get the plaintext again. It then follows that $D_K(E_K(P))=P$.

This notation suggests that *E* and *D* are just mathematical functions, which they are. The only tricky part is that both are functions of two parameters, and we have written one of the parameters (the key) as a subscript, rather than as an argument, to distinguish it from the message. A fundamental rule of cryptography is that one must assume that the cryptanalyst knows the methods used for encryption and decryption. In other words, the cryptanalyst knows how the encryption method, *E,* and decryption, *D*, of Fig. 13.1 work in detail. The amount of effort necessary to invent, test, and install a new algorithm every time the old method is compromised has always made it impractical to keep the encryption algorithm secret.

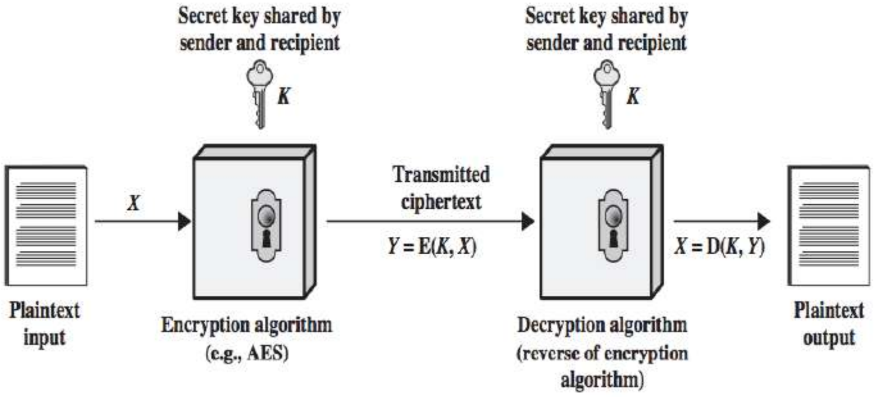

Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E(K, X)
X = D(K, Y)
Plaintext input
Encryption algorithm (e.g., AES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 13.1. Symmetric-Key Encryption Model

13.3.1 Components of a Cryptosystem

The various components of a basic cryptosystem are as follows:

a). **Plaintext.** It is the data to be protected during transmission.

b). **Encryption Algorithm.** It is a mathematical process that produces a ciphertext for any given plaintext and encryption key. It is a cryptographic algorithm that takes plaintext and an encryption key as input and produces a ciphertext.

c). **Ciphertext.** It is the scrambled version of the plaintext produced by the encryption algorithm using a specific the encryption key. The ciphertext is not guarded. It flows on public channel. It can be intercepted or compromised by anyone who has access to the communication channel.

d). **Decryption Algorithm,** It is a mathematical process, that produces a unique plaintext for any given ciphertext and decryption key. It is a cryptographic algorithm that takes a ciphertext and a decryption key as input, and outputs a plaintext. The decryption algorithm essentially reverses the encryption algorithm and is thus closely related to it.

e). **Encryption Key.** It is a value that is known to the sender. The sender inputs the encryption key into the encryption algorithm along with the plaintext in order to compute the ciphertext.

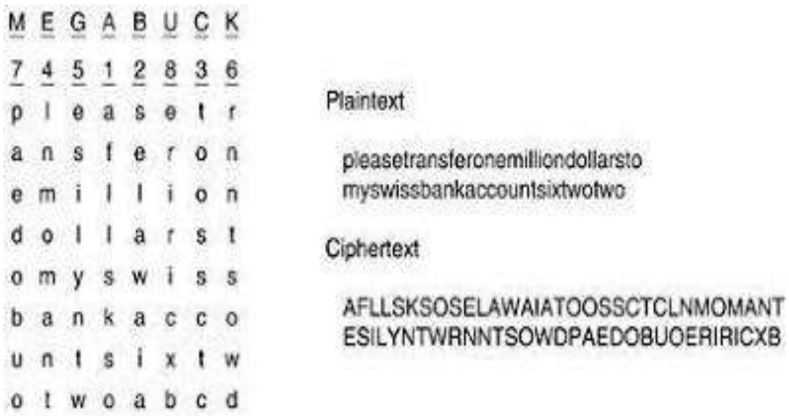
f). **Decryption Key.** It is a value that is known to the receiver. The decryption key is related to the encryption key, but is not always identical to it. The receiver inputs the decryption key into the decryption algorithm along with the ciphertext in order to compute the plaintext.

NOTES

13.4. TRANSPOSITION AND SUBSTITUTION CIPHERS

13.4.1. Transposition Ciphers:

The Substitution ciphers safeguard the order of the plaintext symbols but masquerade them. **Transposition ciphers**, in contrast, reorder the letters but do not masquerade them. The figure13.2 illustrates a common transposition cipher, the columnar transposition. The cipher is keyed by a word or phrase not containing any repeated letters. In this example, MEGABUCK is the key. The purpose of the key is to number the columns, column 1 being under the key letter closest to the start of the alphabet, and so on. The plaintext is written horizontally, in rows, padded to fill the matrix if need be. The cipher text is read out by columns, starting with the column whose key letter is the lowest.



M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Figure 13.2.Transposition Cipher.

In order to crash a transposition cipher, the cryptanalyst must first be aware that he is dealing with a transposition cipher. By looking at the frequency of *E, T, A, O, I, N*, etc., it is easy to see if they fit the normal pattern for plaintext. If so, the cipher is clearly a transposition cipher, because in such a cipher every letter represents itself, keeping the frequency distribution intact.

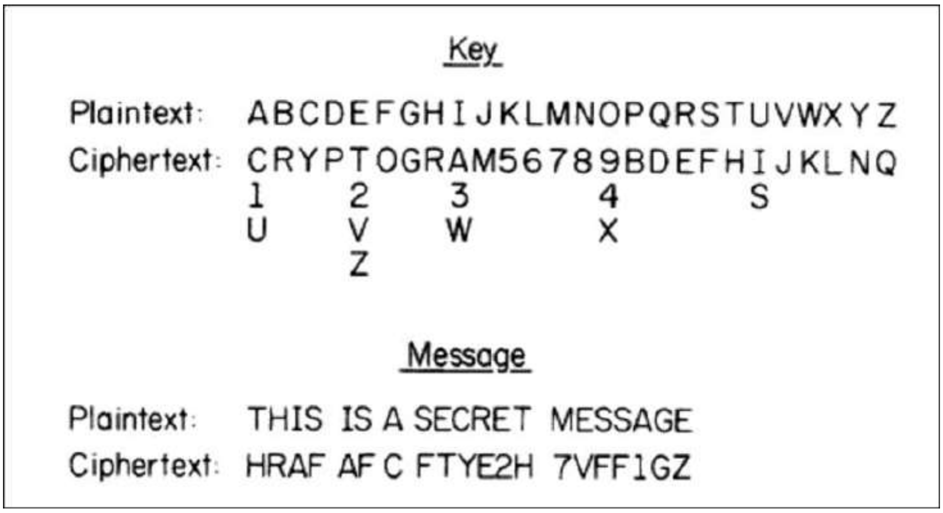
The next step is to make a guess at the number of columns. In many cases a probable word or phrase may be guessed at from the context. For example, suppose that our cryptanalyst suspects that the plaintext phrase *million dollars* occurs somewhere in the message. Observe that digrams *MO, IL, LL, LA, IR* and *OS* occur in the cipher text as a result of this phrase wrapping around. The cipher text letter *O* follows the cipher text letter *M* (i.e., they are vertically adjacent in column 4) because they are separated in the probable phrase by a distance equal to the key length. If a key of length seven had been used, the diagrams *MD, IO, LL, LL, IA, OR,* and *NS* would have occurred instead. In fact, for each key length, a different set of diagrams is produced in the cipher text. By hunting for the various possibilities, the cryptanalyst can often easily determine the key length.

The remaining step is to order the columns. When the number of columns, *k*, is small, each of the *k*(*k* - 1) column pairs can be examined to see if its diagram frequencies match those for English plaintext. The pair with the best match is assumed to be correctly positioned. Now each remaining column is tentatively tried as the successor to this pair. The column whose diagram and trigram frequencies give the best match is tentatively assumed to be correct. The predecessor column is found in the same way. The entire process is continued until a potential ordering is found. Chances are that the plaintext will be recognizable at this point (e.g., if *million* occurs, it is clear what the error is).

Some transposition ciphers accept a fixed-length block of input and produce a fixed-length block of output. These ciphers can be completely described by giving a list telling the order in which the characters are to be output. For example, the cipher of figure 13.2.can be seen as a 64 character block cipher. Its output is 4, 12, 20, 28, 36, 44, 52, 60, 5, 13 , ... , 62. In other words, the fourth input character, *a*, is the first to be output, followed by the twelfth, *f*, and so on.

13.4.2. Substitution Ciphers

In a **substitution cipher** each letter or group of letters is replaced by another letter or group of letters to disguise it. One of the oldest known ciphers is the **Caesar cipher**, attributed to Julius Caesar. In this method, *a* becomes *D*, *b* becomes *E*, *c* becomes *F*, ... , and *z* becomes *C*. For example, *attack* becomes *DWWDFN*. In examples, plaintext will be given in lower case letters, and cipher text in upper case letters. A slight generalization of the Caesar cipher allows the cipher text alphabet to be shifted by *k* letters, instead of always 3. In this case *k* becomes a key to the general method of circularly shifted alphabets. The Caesar cipher may have fooled Pompey, but it has not fooled anyone since.



```
                     Key

Plaintext:  ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext: CRYPTOGRAM56789BDEFHIJKLNQ
            1    2  3     4     S
            U    V  W     X
                 Z

                   Message

Plaintext:  THIS IS A SECRET MESSAGE
Ciphertext: HRAF AF C FTYE2H 7VFF1GZ
```

Figure 13.3.Substitution Ciphers

The next improvement is to have each of the symbols in the plaintext, say, the 26 letters for simplicity, map onto some other letter. For example,

Plaintext: a b c d e f g h i j k l m n o p q r s t u v w x y z

Ciphertext: Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

The common scheme of symbol-for-symbol substitution is called a **mono alphabetic substitution**, with the key being the 26-letter string corresponding to the full alphabet. For the key above, the plaintext *attack* would be transformed into the cipher text *QZZQEA*. At first glance this might appear to be a safe system because although the cryptanalyst knows the general system (letter-for-letter substitution), he does not know which of the 26! $\approx$4 x 10^{26} possible keys is in use.

In contrast with the Caesar cipher, trying all of them is not a promising approach. Even at 1nsec per solution, a computer would take 10^{10} years to try all the keys. The cipher can be broken easily. The basic attack takes advantage of the statistical properties of natural languages. In English, for example, *e* is the most common letter, followed by *t*, *o*, *a*, *n*, *i*, etc. The most common two-letter combinations, or **diagrams**, are *th*, *in*, *er*, *re*, and *an*. The most common three-letter combinations, or **trigrams**, are *the*, *ing*, *and*, and *ion*.

A cryptanalyst trying to break a mono alphabetic cipher would start out by counting the relative frequencies of all letters in the cipher text. Then he might tentatively assign the most common one to *e* and the next most common one to *t*. He would then look at trigrams to find a common one of the form *tXe*, which strongly suggests that *X* is *h*. Similarly, if the pattern *thYt* occurs frequently, the *Y* probably stands for *a*.

With this information, he can look for a frequently occurring trigram of the form *aZW*, which is most likely *and*. By making guesses at common letters, digrams, and trigrams and knowing about likely patterns of vowels and consonants, the cryptanalyst builds up a tentative plaintext, letter by letter.

The difference between substitution cipher technique and transposition cipher technique are listed in table 13.1.

Table 13.1 **Difference between Substitution Cipher Technique and Transposition Cipher Technique**

S.No	Substitution Cipher Technique	Transposition Cipher Technique
1.	In substitution Cipher Technique, plain text characters are replaced with other characters, numbers and symbols.	In transposition Cipher Technique, plain text characters are rearranged with respect to the position.
2.	Substitution Cipher's forms are: Mono alphabetic substitution cipher and poly alphabetic substitution cipher.	Transposition Cipher's forms are: Key-less transposition cipher and keyed transposition cipher.
3.	In substitution Cipher Technique, character's identity is changed while its position remains unchanged.	While in transposition Cipher Technique, The position of the character is changed but character's identity is not changed.
4.	In substitution Cipher Technique, The letter with low frequency can detect plaint ext.	While in transposition Cipher Technique, The Keys which are nearer to correct key can disclose plain text.
5.	The example of substitution Cipher is Caesar Cipher.	The example of transposition Cipher is Reil Fence Cipher.

13.5. CRYPTOGRAPHIC PRINCIPLES

Fundamental Cryptographic Principles are classified in to two:

(1) Redundancy:

i). The first principle is that all encrypted messages must contain some redundancy, that is, information not needed to understand the message.

ii). Cryptographic principle 1: "***Messages must contain some redundancy.***"

iii). In other words, upon decrypting a message, the recipient must be able to tell whether it is valid by simply inspecting it and perhaps performing a simple computation.

iv). This redundancy is needed to prevent active intruders from sending garbage and tricking the receiver into decrypting the garbage and acting on the "plaintext."

v). However, this same redundancy makes it much easier for passive intruders to break the system.

(2) Freshness:

i). The second cryptographic principle is that some measures must be taken to ensure that each message received can be verified as being fresh, that is, sent very recently.

ii). This measure is needed to prevent active intruders from playing back old messages.

iii). Cryptographic principle 2: “***Some method is needed to foil replay attacks.***”

iv). One such measure is including in every message a timestamp valid only for, say, 10 seconds.

v). The receiver can then just keep messages around for 10 seconds, to compare newly arrived messages to previous ones to filter out duplicates. Messages older than 10 seconds can be thrown out, since any replays sent more than 10 seconds later will be rejected as too old.

13.6 CHECK YOUR PROGRESS QUESTIONS

1. What is Cryptography?

2. Mention the role of the Intruder.

3. Define Cryptanalysis.

4. What is meant by Cryptology?

13.7 ANSWERS TO CHECK YOUR PROGRESS QUESTIONS

1. Cryptography means science of "secret writing."

2. The Intruder means hacker who hears and accurately copies down the complete cipher text.

3. The process of breaking ciphers is known as Cryptanalysis.

4. The art of devising cryptography is collectively known ascryptology.

13.8 SUMMARY

Most security problems are intentionally caused by malicious people trying to gain some benefit, get attention, or to harm someone. Network security problems can be divided roughly into four closely intertwined areas: secrecy, authentication, non-repudiation, and integrity control. Secrecy, also called confidentiality, has to do with keeping information out of the hands of unauthorized users. The network security mechanisms are able to provide security for data transmission while sending and receiving through the mass media like internet. Transposition ciphers, in contrast, reorder the letters but do not disguise them. The cipher is keyed by a word or phrase not containing any repeated letters. Some transposition ciphers accept a fixed-length block of input and produce a fixed-length block of output. In a substitution cipher each letter or group of letters is replaced by another letter or group of letters to disguise it. One of the oldest known ciphers is the Caesar cipher, attributed to Julius Caesar.

13.9 KEY WORDS

- **Plain Text :**The messages to be encrypted, known as the plaintext, are transformed by a function that is parameterized by a key.

- **Cipher Text :**The output of the encryption process, known as the cipher text, is then transmitted, often by messenger or radio.

- **Encryption Algorithm:** The algorithm is used to encrypts the plain text with help of key.

- **Decryption Algorithm:**The algorithm is used to decrypts the cipher text with help of key.

- **Encryption Key:** The key is used for Encryption.

- **Decryption Key :** The key is used for Decryption.

- A cryptographic **principles** are :

    - **Redundancy:** Messages must contain some redundancy.
    - **Freshness:** Some method is needed to foil replay attacks.

13.10 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1) What is meant by encryption model?

2) Differentiate Substitution and Transposition ciphers.

Long Answer Questions

1. Explain about Substitution and Transposition ciphers with suitable examples.

13.11 FURTHER READINGS

1. Andrew S. Tanenbaum, Computer Networks, PHI, Fourth Edition, 2003.
2. William Stallings, Data and Computer Communication, PHI, Eighth Edition, 2009
3. BehrouzA.Forouzan, "Cryptography and Network Security", TMH Edition, 2010.
4. Bruce Schneider, "Applied Cryptography", John Wiley and Sons, 1996.

UNIT 14. SYMMETRIC KEY CRYPTOGRAPHY

Structure

14.0 INTRODUCTION

In this unit, we will discover about Symmetric key cryptography such as Block algorithms and Stream algorithms, Asymmetric key cryptography and Security services.

14.1 OBJECTIVES

After going through this unit, you will describe about DES algorithm, AES algorithm, RSA algorithm, Message confidentiality. Message Integrity, Message Authentication, Message non-reproduction, Entity Authentication.

14.2. SYMMETRIC KEY CRYPTOGRAPHY

Symmetric encryption is a type of encryption where only one key (ie. secret key) is used to both encrypt and decrypt electronic information. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. This encryption method differs from asymmetric encryption where a pair of keys, one public and one private, is used to encrypt and decrypt messages.

Through using symmetric encryption algorithms, data is converted to a form that cannot be understood by anyone who does not possess the secret key to decrypt it. Once the intended recipient who possesses the key has the message, the algorithm reverses its action so that the message is returned to its original and understandable form. The secret key that the sender and recipient both use could be a specific password/code and it can be random string of letters or numbers that have been generated by a secure random number generator (RNG). For banking-grade encryption, the symmetric keys must be created using an RNG that is certified according to industry standards, such as FIPS 140-2.

There are two types of symmetric encryption algorithms:

a) **Block algorithms.** Set lengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks.

b) **Stream algorithms.** Data is encrypted as it streams instead of being retained in the system's memory. Some examples of symmetric encryption algorithms include:

i). AES (Advanced Encryption Standard)
ii). DES (Data Encryption Standard)
iii). IDEA (International Data Encryption Algorithm)
iv). Blowfish (Drop-in replacement for DES or IDEA)
v). RC4 (Rivest Cipher 4)
vi). RC5 (Rivest Cipher 5)
vii). RC6 (Rivest Cipher 6)

The AES, DES, IDEA, Blowfish, RC5 and RC6 are block ciphers. RC4 is stream cipher. Since DES is based on the Feistel Cipher, all that is required to specify DES is

a). Round function
b). Key schedule
c). Any additional processing − Initial and final permutation.

14.2.1. DATA ENCRYPTION STANDARD (DES)

In modern computing, DES was the first standardized cipher for securing electronic communications, and is used in variations (e.g. 2-key or 3-key 3DES). The Data Encryption Standard (**DES**) is a symmetric-key block cipher which is adopted by U.S. Government and developed by IBM as its official standard for unclassified information in 1977. **DES** is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. DES is an implementation of a Feistel Cipher. An outline of DES is shown in Fig. 14.1(a) Plaintext is encrypted in blocks of 64 bits, yielding 64 bits of cipher text.

The algorithm, which is parameterized by a 56-bit key, has 19 distinct stages. The first stage is a key-independent transposition on the 64-bit plaintext. The last stage is the exact inverse of this transposition. The stage prior to the last one exchanges the leftmost 32 bits with the rightmost 32 bits. The remaining 16 stages are functionally identical but are parameterized by different functions of the key. The algorithm has been designed to allow decryption to be done with the same key as encryption, a property needed in any symmetric-key algorithm. The steps are just run in the reverse order.

The operation of one of these intermediate stages is illustrated in Fig. 14.1(b). Each stage takes two 32-bit inputs and produces two 32-bit outputs. The left output is simply a copy of the right input. The right output is the bitwise XOR of the left input and a function of the right input and the key for this stage, Ki . All the complexity lies in this function.

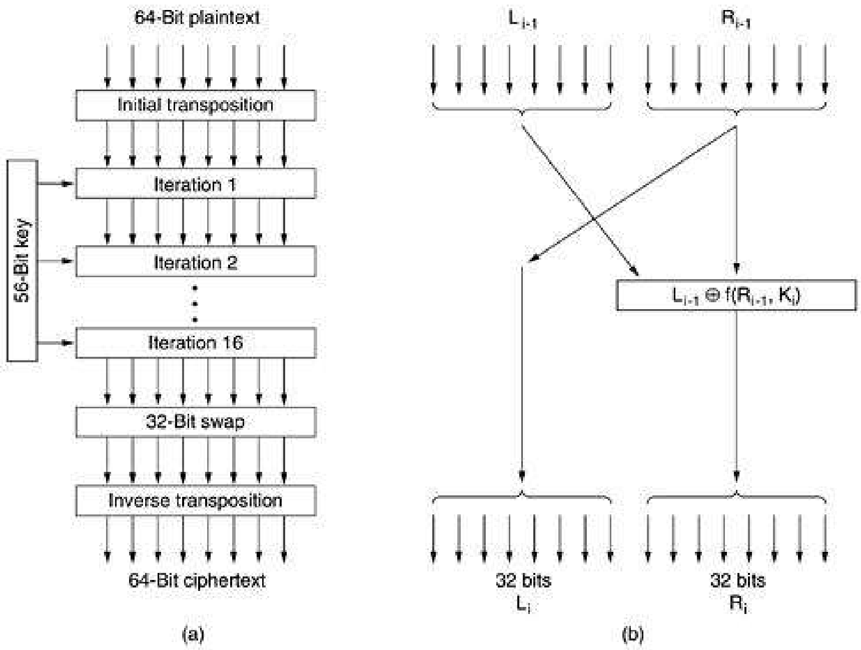


Figure 14.1.The Data Encryption Standard. (a) General outline. (b) Detail of one iteration. The circled + means exclusive OR.

The function is processed in four steps as sequence.

i). First, a 48-bit number, *E*, is constructed by expanding the 32-bit R_{i-1} according to a fixed transposition and duplication rule.

ii). Second, *E* and K_i are XORed together. This output is then partitioned into eight groups of 6 bits each, each of which is fed into a different S-box.

iii). Each of the 64 possible inputs to an S-box is mapped onto a 4-bit output.

iv). Finally, these 8 x 4 bits are passed through a P-box.

In each of the 16 iterations, a different key is used. Before the algorithm starts, a 56-bit transposition is applied to the key. Just before each iteration, the key is partitioned into two 28-bit units, each of which is rotated left by a number of bits dependent on the iteration number. K_i is derived from this rotated key by applying yet another 56-bit transposition to it. A different 48-bit subset of the 56 bits is extracted and permuted on each round. A technique that is sometimes used to make DES stronger is called **whitening**. It consists of XORing a random 64-bit key with each plaintext block before feeding it into DES and then XORing a second 64-bit key with the resulting cipher text before transmitting it. Whitening can easily be removed by running the reverse operations (if the receiver has the two whitening keys). Since this technique effectively adds more bits to the key length, it makes exhaustive search of the key space much more time consuming. Note that the same whitening key is used for each block (i.e., there is only one whitening key).

14.2.2. TRIPLE DES

As early as 1979, IBM realized that the DES key length was too short and devised a way to effectively increase it, using triple encryption (Tuchman, 1979). The Triple DES is an enhancement of DES which has 64 bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. It is a known fact that Triple DES is slower than other block cipher methods. The method chosen, which has since been incorporated in International Standard 8732, is illustrated in fig. 14.2. Here two keys and three stages are used. In the first stage, the plaintext is encrypted using DES in the usual way with K_1. In the second stage, DES is run in decryption mode, using K_2 as the key. Finally, another DES encryption is done with K_1.

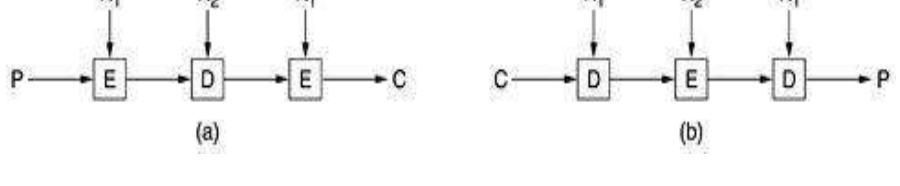


Figure 14.2.(a) Triple Encryption using DES.
(b) Decryption.

This design immediately gives rise to two questions. First, why are only two keys used, instead of three? Second, why is **EDE** (**Encrypt Decrypt Encrypt**) used, instead of **EEE** (**Encrypt Encrypt Encrypt**)? The reason that two keys are used is that even the most paranoid cryptographers believe that 112 bits is adequate for routine commercial applications for the time being. Going to 168 bits would just add the unnecessary overhead of managing and transporting another key for little real gain.

The reason for encrypting, decrypting, and then encrypting again is backward compatibility with existing single-key DES systems. Both the encryption and decryption functions are mappings between sets of 64-bit numbers. From a cryptographic point of view, the two mappings are equally strong. By using EDE, however, instead of EEE, a computer using triple encryption can speak to one using single encryption by just setting K1 = K2. This property allows triple encryption to be phased in gradually, something of no concern to academic cryptographers, but of considerable importance to IBM and its customers.

14.2.3. AES

The most commonly used symmetric algorithm is the Advanced Encryption Standard (AES), which was originally known as Rijndael. This is the standard set by the U.S. National Institute of Standards and Technology in 2001 for the encryption of electronic data announced in U.S. FIPS PUB 197. This standard supersedes DES, which had been in use since 1977. Under NIST, the AES cipher has a block size of 128 bits, but can have three different key lengths as shown with AES-128, AES-192 and AES-256. As DES began approaching the end of its useful life, even with triple DES, **NIST** (**National Institute of Standards and Technology**), the agency of the U.S. Dept. of Commerce charged with approving standards for the U.S. Federal Government, decided that the government needed a new cryptographic standard for unclassified use. NIST was keenly aware of all the controversy surrounding DES and well knew that if it just announced a new standard, everyone knowing anything about cryptography would automatically assume that NSA had built a back door into it so NSA could read everything encrypted with it. Under these conditions, probably no one would use the standard and it would most likely die a quiet death. The architecture of Advanced Encryption Standard algorithm is presented in fig. 14.3.

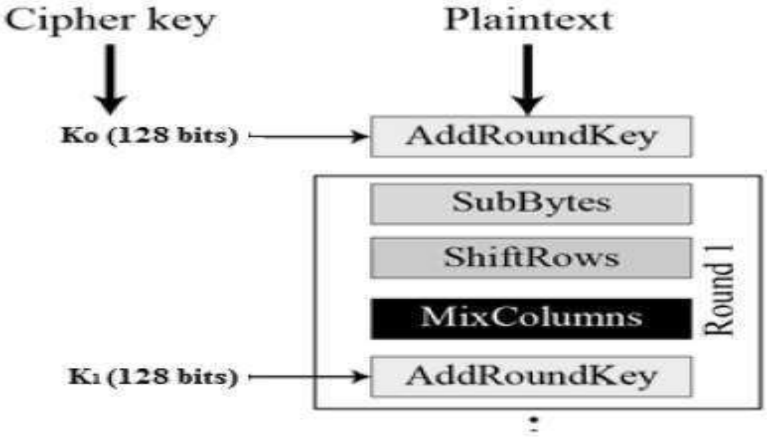


Figure 14.3.Advanced Encryption Standard

The NIST have developed different approach for a government bureaucracy: it sponsored a cryptographic bake-off (contest). In January

1997, researchers from all over the world were invited to submit proposals for a new standard, to be called **AES** (**Advanced Encryption Standard**). The bake-off rules were:

i). The algorithm must be a symmetric block cipher.
ii). The full design must be public.
iii). Key lengths of 128, 192, and 256 bits must be supported.
iv). Both software and hardware implementations must be possible.
v). The algorithm must be public or licensed on non-discriminatory terms.

Owing to the extraordinary openness of the competition, the technical properties of Rijndael, and the fact that the winning team consisted of two young Belgian cryptographers, it is expected that Rijndael will become the world's dominant cryptographic standard for at least a decade. Rijndael supports key lengths and block sizes from 128 bits to 256 bits in steps of 32 bits. The key length and block length may be chosen independently. However, AES specifies that the block size must be 128 bits and the key length must be 128, 192, or 256 bits. It is doubtful that anyone will ever use 192-bit keys, so de facto, AES has two variants: a 128-bit block with 128-bit key and a 128-bit block with a 256-bit key. A 128-bit key gives a key space of $2^{128} \approx 3$ x 103^{8} keys. Even if NSA manages to build a machine with 1 billion parallel processors, each being able to evaluate one key per picoseconds, it would take such a machine about 10^{10} years to search the key space. The fig.14.4.shows the architecture of AES-128 bit key generation.

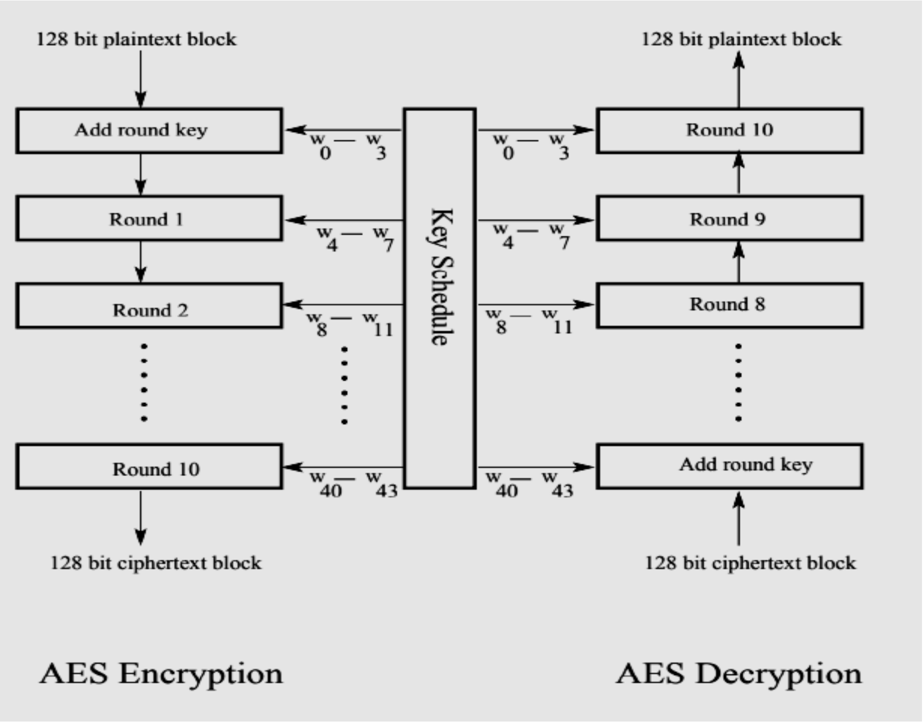


Figure 14.4.Architecture of AES-128 Bit Key Generation

Rijndael is based on Galois field theory, which gives it some provable security properties. However, it can also be viewed as C code, without getting into the mathematics. Like DES, Rijndael uses substitution and permutations, and it also uses multiple rounds. The number of rounds depends on the key size and block size, being 10 for 128-bit keys with 128-bit blocks and moving up to 14 for the largest key or the largest block. However, unlike DES, all operations involve entire bytes, to allow for efficient implementations in both hardware and software. An outline of Rijndael Code for Advanced Encryption Standard is given in Fig.14.5.

The Pseudo C code for Rijndael is:

```
Rijndael(State,CipherKey) {
KeyExpansion(CipherKey,ExpandedKey);
AddRoundKey(State,ExpandedKey);
For( i=1 ; i FinalRound(State,ExpandedKey + Nb*Nr);
}
```

And the round function is defined as:

```
Round(State,RoundKey) {
ByteSub(State);
ShiftRow(State);
MixColumn(State);
AddRoundKey(State,RoundKey);
}
```

Figure 14.5.Rijndael Code for Advanced Encryption Standard

The function Rijndael has three parameters. They are: plaintext, an array of 16 bytes containing the input data, ciphertext, an array of 16 bytes where the enciphered output will be returned, and key, the 16-byte key. During the calculation, the current state of the data is maintained in a byte array, state, whose size is NROWS x NCOLS. For 128-bit blocks, this array is 4 x 4 bytes. With 16 bytes, the full 128-bit data block can be stored.

14.2.4. Applications of Symmetric Encryption

While symmetric encryption is an older method of encryption, it is faster and more efficient than asymmetric encryption, which takes a toll on networks due to performance issues with data size and heavy CPU use. Due to the better performance and faster speed of symmetric encryption (compared to asymmetric), symmetric cryptography is typically used for encrypting large amounts of data, e.g. for database encryption. In the case of a database, the secret key might only be available to the database itself to encrypt or decrypt.

Some examples of where symmetric cryptography is used are:

i). Payment applications, such as card transactions where PII needs to be protected to prevent identity theft or fraudulent charges.
ii). Validations to confirm that the sender of a message is who he claims to be
iii). Random number generation or hashing

- **Drawbacks of Symmetric Encryption**

Unfortunately, symmetric encryption does come with its own drawbacks. Its weakest point is its aspects of key management, including:

i). Key Exhaustion

Symmetric Encryption suffers from behavior where every use of a key ‘leaks’ some information that can potentially be used by an attacker to reconstruct the key. The defenses against this behavior include using a key hierarchy to ensure that master or key-encryption keys are not over-used and the appropriate rotation of keys that do encrypt volumes of data. To be tractable, both these solutions require competent key-management strategies as if (for example) a retired encryption key cannot be recovered the data is potentially lost.

ii). Attribution data

Unlike asymmetric (public-key) *Certificates*, symmetric keys do not have embedded metadata to record information such as expiry date or an Access Control List to indicate the use the key may be put to - to Encrypt but not Decrypt for example. The latter issue is somewhat addressed by standards such as ANSI X9-31 where a key can be bound to information prescribing its usage. But for full control over *what* a key can be used for and *when* it can be used, a key-management system is required.

14.3. Asymmetric Key Cryptography

The **Asymmetric cryptography**, also known as **public key cryptography**, uses **public** and private **keys** to encrypt and decrypt data. The **keys** are simply large numbers that have been paired together but are not identical (**asymmetric**). One **key** in the pair can be shared with everyone; it is called the **public key**. The figure14.6 offers the architecture of asymmetric key cryptography.

The most important properties of public key encryption scheme are

i). Different keys are used for encryption and decryption. This is a property which set this scheme different than symmetric encryption scheme.
ii). Each receiver possesses a unique decryption key, generally referred to as his private key.
iii). Receiver needs to publish an encryption key, referred to as his public key.
iv). Some assurance of the authenticity of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.
v). Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from the cipher text and the encryption (public) key.
vi). Though private and public keys are related mathematically, it is not be feasible to calculate the private key from the public key. In fact, intelligent part of any public-key cryptosystem is in designing a relationship between two keys.

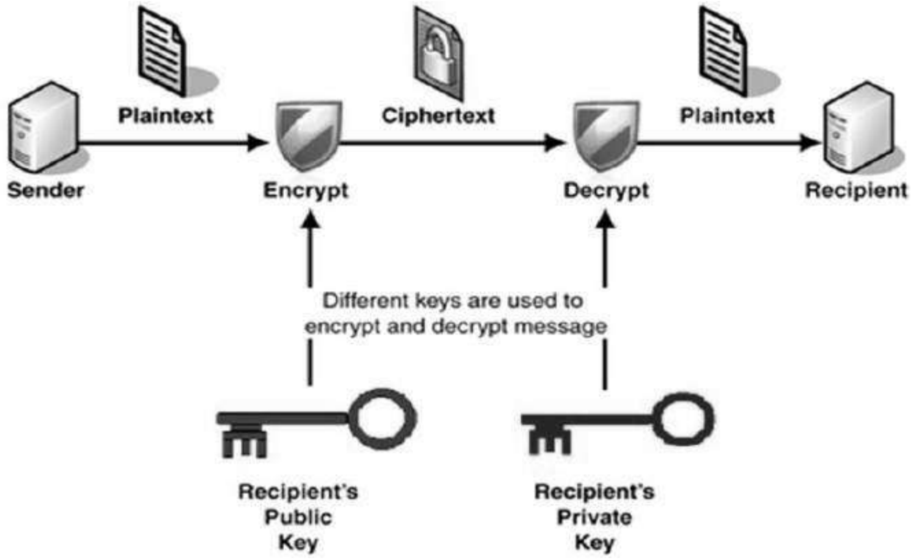


Figure 14.6. Architecture of Asymmetric Key Cryptography

14.3.1. RSA Cryptosystem

In 1977, the three scholars **Ron Rivest, Adi Shamir,** and **Len Adleman** was invented first and publicly illustrated the **RSA** public-key cryptosystems for secure data transmission. In RSA, the encryption key is public and it is different from the decryption key which is kept secret or private. In RSA, this asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem". The RSA cryptosystem has two aspects, the first aspect is generation of key pair and the second aspect is encryption-decryption algorithms.

A. Generation of RSA Key Pair

Each person or a party who desires to participate in communication using encryption needs to generate a pair of keys, namely public key and private key. The process followed in the generation of keys is described below:

➢ **Generate the RSA modulus (n)**

i). Select two large primes, p and q.

ii). Calculate n=p*q. For strong unbreakable encryption, let n be a large number, typically a minimum of 512 bits.

➢ **Find Derived Number (e)**

i). Number **e** must be greater than 1 and less than (p − 1) (q − 1).

ii). There must be no common factor for e and (p − 1) (q − 1) except for 1. In other words two numbers e and (p – 1) (q – 1) are co-prime.

➢ **Form the public key**

i). The pair of numbers (n, e) form the RSA public key and is made public.

ii). Interestingly, though n is part of the public key, difficulty in factorizing a large prime number ensures that attacker cannot find in finite time the two primes (p & q) used to obtain n. This is strength of RSA.

➢ **Generate the private key**

i). Private Key d is calculated from p, q, and e. For given n and e, there is unique number d.

ii). Number d is the inverse of e modulo (p - 1)(q – 1). This means that d is the number less than (p - 1)(q - 1) such that when multiplied by e, it is equal to 1 modulo (p - 1)(q - 1).

iii). This relationship is written mathematically as follows –

ed = 1 mod (p − 1)(q − 1)

The Extended Euclidean Algorithm takes p, q, and e as input and gives d as output.

➢ **Algorithm for Generation of RSA Key Pair.**

INPUT: Required modulus bit length, kk.

OUTPUT: An RSA key pair ((N,e),d) ((N,e),d) where N is the modulus, the product of two primes (N=pq , N=pq) not exceeding kk bits in length; ee is the public exponent, a number less than and coprime to $(p-1)(q-1)(p-1)(q-1)$; and dd is the private exponent such that $ed \equiv 1 \bmod (p-1)(q-1) ed \equiv 1 \bmod (p-1)(q-1)$.

```
Select a value of ee from 3, 5, 17, 257, 655373, 5, 17, 257, 65537
repeat
   p ← genprime(k/2)
until (pmode)≠1(pmode)≠1
repeat
   q ← genprime(k - k/2)
until (qmode)≠1(qmode)≠1
N ← pq
L ← (p-1)(q-1)
d ← modinv(e, L)
return (N,e,d)
```

- **Example for Generation of RSA Key Pair**

An example of generating RSA Key pair is given below. (For ease of understanding, the primes p & q taken here are small values. Practically, these values are very high).

i). Let two primes be p = 7 and q = 13. Thus, modulus n = pq = 7 x 13 = 91.

ii). Select e = 5, which is a valid choice since there is no number that is common factor of 5 and (p − 1)(q − 1) = 6 × 12 = 72, except for 1.

iii). The pair of numbers (n, e) = (91, 5) forms the public key and can be made available to anyone whom we wish to be able to send us encrypted messages.

iv). Input p = 7, q = 13, and e = 5 to the Extended Euclidean Algorithm. The output will be d = 29.

v). Check that the d calculated is correct by computing −

de = 29 × 5 = 145 = 1 mod 72

- Hence, public key is (91, 5) and private keys is (91, 29).

B. RSA Encryption and Decryption

Once the key pair has been generated, the process of encryption and decryption are relatively straightforward and computationally easy. Interestingly, RSA does not directly operate on strings of bits as in case of symmetric key encryption. It operates on numbers modulo n. Hence, it is necessary to represent the plaintext as a series of numbers less than n. The figure 14.7 presents the architecture of RSA cryptosystem.

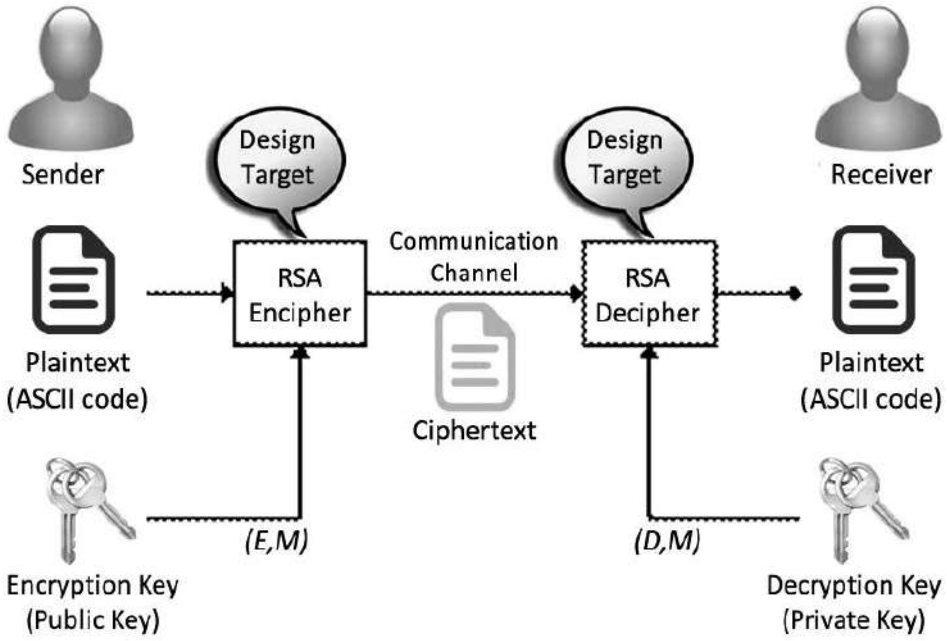


Figure 14.7. Architecture of RSA Cryptosystem

➢ **RSA Encryption**

i). Suppose the sender wish to send some text message to someone whose public key is (n, e).

ii). The sender then represents the plaintext as a series of numbers less than n.

iii). To encrypt the first plaintext P, which is a number modulo n. The encryption process is simple mathematical step as −

$$C = P^e \text{ mod } n$$

i). In other words, the ciphertext C is equal to the plaintext P multiplied by itself e times and then reduced modulo n. This means that C is also a number less than n.

ii). Returning to our Key Generation example with plaintext P = 10, we get ciphertext C −

$$C = 10^5 \text{ mod } 91$$

➢ **RSA Decryption**

- The decryption process for RSA is also very straightforward. Suppose that the receiver of public-key pair (n, e) has received a ciphertext C.
- Receiver raises C to the power of his private key d. The result modulo n will be the plaintext P.

$$\text{Plaintext} = C^d \text{ mod } n$$

- Returning again to our numerical example, the ciphertext C = 82 would get decrypted to number 10 using private key 29 −

$$\text{Plaintext} = 82^{29} \text{ mod } 91 = 10$$

- **RSA Analysis**

The security of RSA depends on the strengths of two separate functions. The RSA cryptosystem is most popular public-key cryptosystem strength of which is based on the practical difficulty of factoring the very large numbers.

1. **Encryption Function** − It is considered as a one-way function of converting plaintext into cipher text and it can be reversed only with the knowledge of private key d.

2. **Key Generation** − The difficulty of determining a private key from an RSA public key is equivalent to factoring the modulus n. An attacker thus cannot use knowledge of an RSA public key to determine an RSA private key unless he can factor n. It is also a one way function, going from p & q values to modulus n is easy but reverse is not possible.

If either of these two functions are proved non one-way, then RSA will be broken. In fact, if a technique for factoring efficiently is developed then RSA will no longer be safe. The strength of RSA encryption drastically goes down against attacks if the number p and q are not large primes and/ or chosen public key e is a small number.

14.3.2 Applications of Asymmetric Cryptography

a). Digital Signatures

Digital signatures allow developers to verify that a message was provided from a trusted identity.

For example: digital signatures would enable the development of **secure automatic updates into WordPress** (whereas, currently, if an attacker gained access to their update server, they'd be able to immediately install malware on over **30% of the websites on the Internet**).

Digital signatures are extremely common. They underpin every public key infrastructure (PKI), especially the Certificate Authority system upon which Transport-Layer Security (TLS) depends.

b). Transport-Layer Security

TLS is the most common use-case for asymmetric cryptography, and the ones that developers and end users should be least involved withTLS is an online protocol that authenticates the server (and optionally the client), negotiates a shared encryption key, then encrypts normal traffic.

When HTTP is communicated over TLS, it's called HTTPS.

TLS is widely regarded as the most important cryptography protocol on the Internet, and if your website doesn't support it: **You're insecure**! Get TLS today; it's free.

Recently the IETF finalized **RFC 8446 (TLS version 1.3)**, which is a *considerable* improvement over the previous versions of the protocol. It is my hope that every website on the Internet will one day speak a minimum of TLS 1.3. It really *is* that good.

However, and this may come as a surprise to some cryptography experts, TLS is ***NOT*** the end-all be-all of software developers' experience with asymmetric cryptography. Read on to learn more.

c). Sealing APIs: Offline Public-Key Encryption

A sealing API is one that satisfies this use case:

- Encrypt some data with a public key, in an online application; then
- Decrypt the data with the appropriate secret key, hopefully in an offline (airgapped) computer

The actual message encryption itself can (and usually does) employ symmetric cryptography, so long as the symmetric key can only be obtained by the recipient in possession of the correct secret key.

In libsodium, crypto_box_seal generates a random ECDH keypair, performs a handshake with the long-term public key, encrypts the message using the shared secret (using an AEAD construction), then prepends the ephemeral public key to the authenticated ciphertext.

➢ **Why Sealing APIs Matter**

The use-case of "encrypt online, decrypt offline" rears its head a lot situations. The most common you're likely to encounter are eCommerce-related.Let's say you're storing sensitive information (e.g. credit card numbers) in a database, but don't want this information easily stolen by an attacker capable of dumping database tables. A sealing API will allow you to encrypt this information with your public key and store the ciphertext in the database. When the time comes to decrypt this data, you can load the ciphertext onto an airgappedcomputer, decrypt it with the secret key, then manually key in the transactions.

Unfortunately, none of **NIST's Post-Quantum Cryptography Round 1 Candidates** appear to be designed with this use-case in mind. The main use-case of these new cryptography designs seem to be simply: **TLS**.

The narrowness of use-case is particularly egregious in Learning With Errors (LWE) protocols, which have a nontrivial chance of failure built in, which in most cases would require the handshake be restarted.You can't restart a handshake against a static public key when the corresponding secret key is offline. You won't even know if it failed.

d). Post-Quantum Readiness: Where to Go From Here?

There are a lot of other use cases (Authenticated Key Exchanges, cryptocurrency, the Double Ratchet from the Signal Protocol, etc.) that weren't covered here, but they're all in the long tail of uncommon requirements.

If a practical quantum computer were developed today, we have proposed designs for digital signature protocols that will likely allow software security to survive (most notably, the SPHINCS family of digital signature algorithms). We also have designs undergoing review right now that could lead to a post-quantum secure TLS in the near future.

But for software that relies on a sealing API to operate securely, there is no immediate post-quantum secure alternative to migrate to. Unless we can convince NIST and the cryptography community to consider sealing APIs a priority, those applications may be left holding the bag long after a post-quantum secure TLS is in the works.

What worries me most is, in the absence of guidance from experts, developers have a tendency to just **roll their own cryptography**.

14.3.3. Difference between Symmetric and Asymmetric Encryption

i). Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetrical encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating.

ii). Symmetric encryption is an old technique while asymmetric encryption is relatively new.

iii). Asymmetric encryption was introduced to complement the inherent problem of the need to share the key in symmetrical encryption model, eliminating the need to share the key by using a pair of public-private keys.

iv). Asymmetric encryption takes relatively more time than the symmetric encryption.

14.4. SECURITY SERVICES

The network security is providing the following services associated to a message and entity. The figure 14.8 shows the network security services.

a). Message confidentiality.

b). Message Integrity.

c). Message Authentication.

d). Message non-reproduction.

e). Entity Authentication.

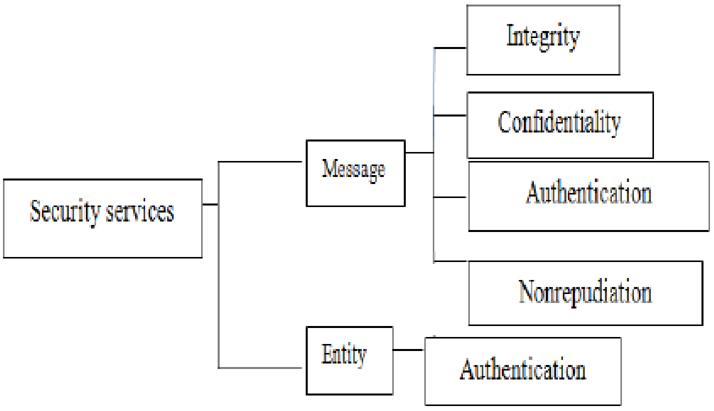


Figure 14.8. Network Security Services

a). Message confidentiality

i). It means that the content of a message when transmitted across a network must remain confidential, *i.e.* only the intended receiver and no one else should be able to read the message.

ii). The users; therefore, want to encrypt the message they send so that an eavesdropper on the network will not be able to read the contents of the message.

b). Message Integrity

i). It means the data must reach the destination without any adulteration *i.e.* exactly as it was sent.

ii). There must be no changes during transmission, neither accidentally nor maliciously.

iii). Integrity of a message is ensured by attaching a checksum to the message.

iv). The algorithm for generating the checksum ensures that an intruder cannot alter the checksum or the message.

c). Message Authentication

In message authentication the receiver needs to be .sure of the sender's identity *i.e.* the receiver has to make sure that the actual sender is the same as claimed to be.

There are different methods to check the genuineness of the sender:

The two parties share a common secret code word. A party is required to show the secret code word to the other for authentication.

i). Authentication can be done by sending digital signature.

ii). A trusted third party verifies the authenticity. One such way is to use digital certificates issued by a recognized certification authority.

d). Message non-reproduction

i). Non-repudiation means that a sender must not be able to deny sending a message that it actually sent.

ii). The burden of proof falls on the receiver.

iii). Non-reproduction is not only in respect of the ownership of the message; the receiver must prove that the contents of the message are also the same as the sender sent.

iv). Non-repudiation is achieved by authentication and integrity mechanisms.

e). Entity Authentication

In entity authentication (or user identification) the entity or user is verified prior to access to the system resources.

14.5 CHECK YOUR PROGRESS QUESTIONS

1. What is Encryption function?
2. Define Key generation.
3. Mention about Message confidentiality.

14.6 ANSWERS TO CHECK YOUR PROGRESS

1. **Encryption Function** is a one-way function of converting plaintext into cipher text and it can be reversed only through private key.

2. **Key Generation** is the process of generating keys for cryptography. The key is used to encrypt and decrypt data whatever the data is being encrypted or decrypted. Modern cryptographic systems include symmetric-key algorithms (such as DES and AES) and public-key algorithms (such as RSA).

3. **Message confidentiality**: means the content of a message must remain confidential while transmit over internet. The encryption of message prohibits an eavesdropper on the network will not be able to read the contents of the message.

14.7 SUMMARY

The DESwas the first encryption standard recommended by NIST, DES is having 64 bits key size with 64 bits block size . Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher. AES is a block cipher .It has variable key length of 128, 192, or 256 bits; default 256. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Asymmetric Key Encryption is utilizes public and private keys to encrypt and decrypt data. The keys are simply large numbers that have been paired together but are not identical (asymmetric). One key in the pair can be shared with everyone; it is called the public key. RSA scheme is a block cipher in which the plain text and cipher text are integers between 0 and n-1 for some n**.** Block algorithms meanslengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks. Stream algorithms means **d**ata is encrypted as it streams instead of being retained in the system's memory. Entity Authentication means an entity authentication is performed in order to verify the authenticity of users prior to access to the system resources.

14.8 KEY WORDS

- Symmetric key means only one key is used to encrypt and decrypt data.
- Asymmetric Key means two different keys are used; one for encryption and another for decryption.

14.9 SELF ASSESSMENT QUESTIONS AND EXERCISES

Short Answer Questions

1. What is meant by Triple DES algorithm?
2. Define security services.
3. Differentiate plain text and cipher text.

Long Answer Questions

1. Describe about DES Algorithm
2. Illustrate about AES Algorithm.
3. Elucidate about RSA Algorithm.

14.10 FURTHER READINGS

1. William Stallings, Cryptography and Network Security-Principles and Practices, Prentice-Hall, Third edition, 2003.

2. Johannes A. Buchaman, Introduction to cryptography, Springer-Verlag 2000.

3. AtulKahate, Cryptography and Network Security, TMH. 2007.

SUB.CODE:31333

MODEL QUESTION PAPER

ALAGAPPA UNIVERSITY, KARAIKUDI

DIRECTORATE OF DISTANCE EDUCATION

MSc. INFORMATION TECHNOLOGY DEGREE EXAMINATION
(CBCS - 2018 -2019 Academic Year Onwards)

COMPUTER NETWORKS

DURATION : 3 HRS MAXIMUM MARKS: 75

Part – A [10*2=20]
Answer All the Questions.

1. Define Computer Networks.
2. What is meant by Network Topology?
3. Mention about redundancy in error detection mechanism.
4. What is the role of Automatic Repeat reQuest (ARQ) Protocol?
5. Differentiate Circuit switching and packet switching.
6. State about Dynamic Routing.
7. Mention the significance of Transport Layer.
8. Differentiate UDP and TCP.
9. What is meant by Cryptography?
10. Define security services.

Part – B [5*5=25]
Answer all the Questions choosing either (a) or (b)

11. (a) Elucidate about categories of computer networks.
(or)
(b) Illustrate about Transmission Media.

12. (a) Write notes on Cyclic Redundancy Check (CRC).
(or)
(b) Explain about sliding window protocol.

13. (a) Discuss about Virtual Circuits.
(or)
(b) Explain about link state routing.

14. (a) Describe the organization and functioning on DNS.
(or)
(b) Write notes on Simple Network Management Protocol SNMP.

15. . (a) Discuss about Encryption Model.

(or)

(b) Illustrate about DES algorithm.

Part – C [3*10=30]
Answer any THREE Questions.

16. Describe about different types of Network Topology with suitable diagrams.
17. Illustrate the importance of ALOHA.
18. Write detail notes on shortest path routing.
19. Explain about Remote Procedure Call.
20. Describe about the working principle of RSA algorithm with examples.

****

Master of Computer Applications
31534
COMPUTER NETWORKS

III - Semester

ALAGAPPA UNIVERSITY

[Accredited with A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]

KARAIKUDI – 630 003

Vallal Dr.RM.Alagappa Chettiar

DIRECTORATE OF DISTANCE EDUCATION

COMPUTER NETWOKS

Master of Computer Applications