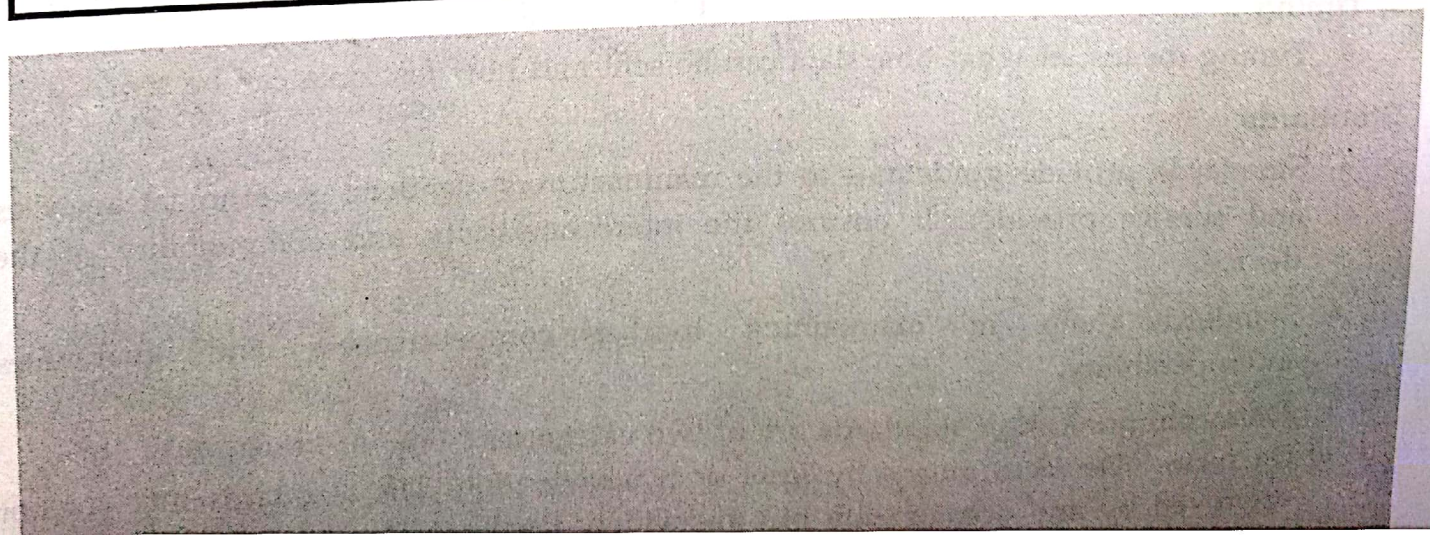
1 Introduction to Computer Networks and Internet

Syllabus

Understanding of network and Internet, The network edge, The network core, Understanding of delay, loss and throughput in the packet-switching network, Protocols layers and their service model, History of the computer network.

Contents

1.1 Internet
1.2 Network Topology ..... **Dec.-11,** ..... Marks 5
1.3 The Network Edge ..... **Winter-13,14,** ..... Marks 7
1.4 Network Core
1.5 Packet Switching
1.6 Delay and Loss in Packet-Switched Networks
1.7 Protocol Layers and Their Service Models
1.8 OSI Reference Model **Winter-13, 14, Dec.-10, 11**
..... **Summer-14,** ..... Marks 8
1.9 TCP/IP Protocol
1.10 Addressing in TCP/IP



Scanned with CamScanner

1.1 Internet

- Internet is known through its applications : The World Wide Web, email, streaming audio and video, chat rooms, and music (file) sharing.
- The Internet is a type of world-wide **computer network**. Internet is a network that interconnects millions of computing devices throughout the world.
- The Internet is a "network of networks" that consists of numerous academic, business, and government networks, which together carry various information and services, such as e-mail, web access, file transfer and many other.
- The Internet provides a much lower cost alternative to PSTN for support of multimedia applications.

1.1.1 Protocol and Standards

- A protocol is a set of rules that governs data communications. Protocol defines the method of communication, how to communicate, when to communicate etc. Important elements of protocol are

  1. Syntax 2. Semantics 3. Timing

1. Syntax

- Syntax means format of data or the structure how it is presented e.g. first eight bits are for sender address, next eight bits for receiver address and rest of the bits for message data.

2. Semantics

- Semantics is the meaning of each section of bits e.g. the address bit means the route of transmission or final destination of the message.

3. Timing

- Timing means, at what time data can be sent and how fast data can be sent.

Standards

- Standards provide guidelines to the manufacturers, venders, government agencies and service provider. It ensures the interconnectivity and compatibility of the device.
- Standards help in maintaining market competitiveness and guarantees interoperability.
- Data communication standards are of two categories

a) De facto : De facto means by facts or by convention. The standards that are not approved by any organization but are widely used are De facto standards. These are established by manufacturers.

Scanned with CamScanner

b) De jure : De jure means by law or by regulation. These are the standards that are recognized officially by an organization.

1.2 Types of Network

GTU : Dec.-11

1.2.1 Local Area Network (LAN)

- The IEEE 802 LAN is a popularly used shared medium peer-to-peer communications network that broadcasts information for all stations to receive.
- The LAN enables stations to communicate directly using a common physical medium on a point-to-point basis without any intermediate switching node being required.
- A LAN is a system composed of computer hardware and transmission media and software.
- LANs are privately owned networks within a single building or campus of upto few km in range. It generally use only one type of transmission media.
- Depends upon application and cost, various topology used in LAN. (e.g. star, bus, ring).
- The basic idea of a LAN is to provide easy access to Data Terminal Equipment (DTEs) within the office. These DTEs are not only computers but other devices, such as printer, plotters and electronic files and databases.
- Fig. 1.2.1 shows the local area networks.

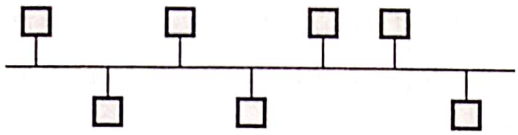


Fig. 1.2.1 LAN

- LAN can provide users
  1) Flexibility
  2) Speed
  3) Reliability
  4) Adaptability
  5) Security
  6) Transparent interface
  7) Access to the other LAN and WAN
  8) Hardware and software sharing
  9) Centralized management
  10) Private ownership of the LAN.

Scanned with CamScanner

- **Attributes of LAN**
  1) The LAN transmits data amongst user stations.
  2) The LAN transmission capacity is more than 1 Mbps.
  3) The LAN channel is typically privately owned by the organization using the facility.
  4) The geographical coverage of LANs is limited to areas less than 5 square kilometers.
- LANs are typically identified by the following properties -
  1. Multiple systems attached to shared medium.
  2. High total bandwidth (~10 Mbps).
  3. Low delay.
  4. Low error rate.
  5. Broadcast / Multicast capability.
  6. Limited geography (1-2 km).
  7. Limited number of stations.
  8. Peer relationship between stations.
  9. Confined to private property.
- The low level protocols used in such environments are different from those used in wide area networks.
- The common forms of LAN are those described by the IEEE standard 802. This standard describes operation upto and including OSI layer 2. Individuals may build what they like on top of these basic protocols.
- A common set of higher level protocols is called TCP/IP which provides OSI layer 3 and 4 functionality, on top of this may be found a set of protocols commonly called Telnet protocols.
- At the lowest level the IEEE 802 specifications split into 3 corresponding to three different but common LAN structures. These are -
  802.3, 802.4, 802.5 standards for topology.
- LANs are capable of transmitting data at very fast rates, much faster than data can be transmitted over a telephone line ; but the distances are limited, and there is also a limit on the number of computers that can be attached to a single LAN.
- The following characteristics differentiate one LAN from another :
  1. **Topology** : The geometric arrangement of devices on the network. For example, devices can be arranged in a ring or in a straight line.
  2. **Protocols** : The rules and encoding specifications for sending data. The protocols also determine whether the network uses a peer-to-peer or client /server architecture.

Scanned with CamScanner

3. **Media :** Devices can be connected by twisted-pair wire, co-axial cables, or fiber optic cables. Some networks do without connecting media altogether, communicating instead via radio waves.

1.2.2 Metropolitan Area Networks (MAN)

- A MAN, while larger than LAN is limited to city or group of nearby corporate offices. It uses similar technology of LAN.
- The Metropolitan Area Network standards are sponsored by the IEEE, ANSI and the Regional Bell operating companies. The MAN standard is organized around a topology and technique called Distributed Queue Dual Bus (DQDB).
- MAN provides the transfer rates from 34 to 150 Mbps.
- MAN is designed with two unidirectional buses. Each bus is independent of the other in the transfer of traffic. The topology can be designed as an open bus or a closed configuration.
- MANs are based on fiber optic transmission technology and provide high speed interconnection between sites. It can support both data and voice.
- MAN as a special category is that a standard has been adopted for them and this standard is now being implemented. It is called IEEE 802.6.

1.2.3 Wide Area Networks (WAN)

- A WAN provides long distance transmission of data and voice.
- A Network that covers a larger area such as a city, state, country or the world is called **wide area network.**
- The WAN contains host and collection of machines. User program is installed on the host and machines. All the host are connected by each other through communication subnet. Subnet carries messages from host to host.
- Fig. 1.2.2 shows the component of WAN.

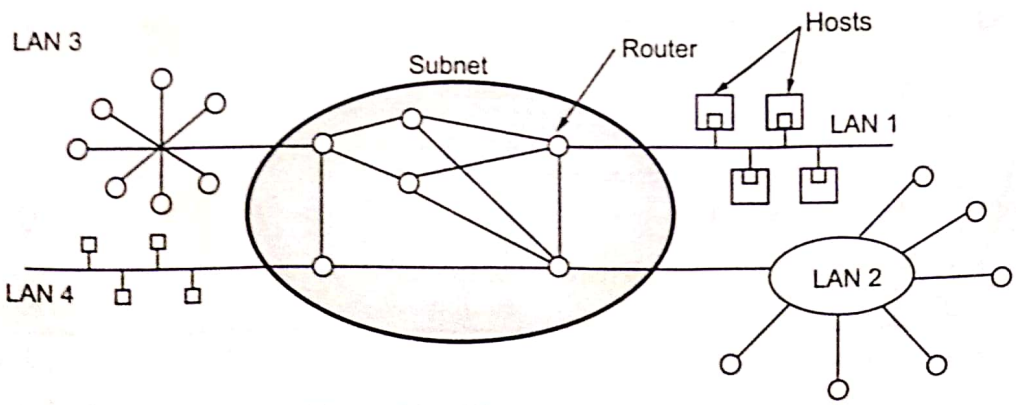


Fig. 1.2.2 Wide area network

Scanned with CamScanner

- Subnet consists of transmission lines and switching elements. The transmission line is used for data transfer between two machines. Switching elements are used for connecting two transmission lines. Switching elements are specialized computers. It selects the proper outgoing line for incoming data and forward the data on that line.
- The switching elements are basically computers and they are called packet switching nodes, intermediate systems and data switching exchanges. These switching elements are also called routers.
- Each host is connected to a LAN on which a router is present. Sometimes the host can be directly connected to the router. The interconnection of routers forms the subnet.
- In the WAN, when the packet is sent from one router to another via one or more intermediate routers, the packet is received at each intermediate router in its entirety. This packet is stored in that router until the required output line is free. The subnet which uses this principle is called point-to-point, store and forward, or packet switched subnet.
- Almost all the WANs use store and forward subnets.
- If the packets are small and of same size, they are also called cells.
- In the point-to-point subnet, the router interconnection topology becomes important. WANs can also use satellite or ground radio system. The routers have antenna, through which they can send or receive data, they can listen from satellite.
- WAN uses hierarchical addressing because they facilitate routing. Addressing is required to identify which network input is to be connected to which network output.

1.2.4 Comparison between LAN, WAN and MAN

Parameter	LAN	WAN	MAN
Area covered	Covers small area. i.e. within the building.	Covers large geographical area.	Covers larger than LAN & smaller than WAN.
Error rates	Lowest.	Highest.	Moderate.
Transmission speed	High speed.	Low speed.	Moderate speed.
Equipment cost	Uses inexpensive equipment.	Uses most expensive equipment.	Uses moderately expensive equipment.

Scanned with CamScanner

1.2.5 Comparison between LAN and WAN

Sr. No.	LAN	WAN
1.	It covers small area.	WAN covers large geographical area.
2.	LAN operates on the principal of broadcasting.	WAN operates on the principal of point to point.
3.	Used for time critical application.	Not used for time critical application.
4.	Transmission speed is high.	Transmission speed is low.
5.	Easy to design and maintain.	Design and maintenance is not easy.
6.	LAN is broadcasting in nature.	WAN is point-to-point in nature.
7.	Transmission medium is co-axial or UTP cable.	Transmission or communication medium is PSTN or satellite link.
8.	LAN does not suffer form propagation delay.	WAN suffer from propagation delay.

1.2.6 Wireless Networks

- A wireless LAN or WLAN is a wireless local area network that uses radio waves as its carrier. The last link with the users is wireless, to give a network connection to all users in a building or campus. The backbone network usually uses cables.
- Wireless LANs operate in almost the same way as wired LANs, using the same networking protocols and supporting the most of the same applications.

How are WLANs Different ?

1. They use specialized **physical and data link** protocols
2. They integrate into existing networks through **access points** which provide a bridging function
3. They let you stay connected as you **roam** from one coverage area to another
4. They have unique **security** considerations
5. They have specific **interoperability** requirements
6. They require **different hardware**
7. They offer **performance** that differs from wired LANs.

- **Physical Layer :** The wireless NIC takes **frames** of data from the link layer, scrambles the data in a predetermined way, then uses the modified data stream to modulate a **radio carrier signal.**
- **Data Link Layer :** Uses Carriers-Sense-Multiple-Access with Collision Avoidance (CSMA/CA).

Scanned with CamScanner

- Wireless Access Points (APs) is a small device that bridges wireless traffic to your network. Most access point's bridge wireless LANs into Ethernet networks, but Token-Ring options are available as well.
- A family of wireless LAN (WLAN) specifications developed by a working group at the Institute of Electrical and Electronic Engineers (IEEE). It defines standard for WLANs using the following four technologies
  1. Frequency Hopping Spread Spectrum (FHSS)
  2. Direct Sequence Spread Spectrum (DSSS)
  3. Infrared (IR)
  4. Orthogonal Frequency Division Multiplexing (OFDM)
- WLAN versions are : 802.11a, 802.11b, 802.11g, 802.11e, 802.11f, 802.11i

1.3 Network Topology

GTU : Winter-13,14

- The physical topology of LAN refers to the way in which the stations are physically interconnected.
- Topology is also defined as, the manner in which nodes are geometrically arranged and connected is known as the topology of the network.
- Physical topology of a local area network should have the following desirable features.
  1. The topology should be flexible to accommodate changes in physical locations of the stations, increase in the number of stations and increase in the LAN geographic coverage.
  2. The cost of physical media and installation should be minimum.
  3. The network should not have any single point of complete failures.
- Network topology refers to the physical layout of the network. Each topology has its own strengths and weaknesses.
- Four types of topologies are commonly used in the network. They are bus, star, ring and mesh topology.

1.3.1 Bus Topology

- Bus topology also called horizontal topology.
- In bus topology, multiple devices are connected one by one, by means of connectors or drop cables.
- When one computer sends a signal up (and down) the wire, all the computers on the network receive the information, but only one accepts the information (using address matching). The rest discard the message.

Scanned with CamScanner

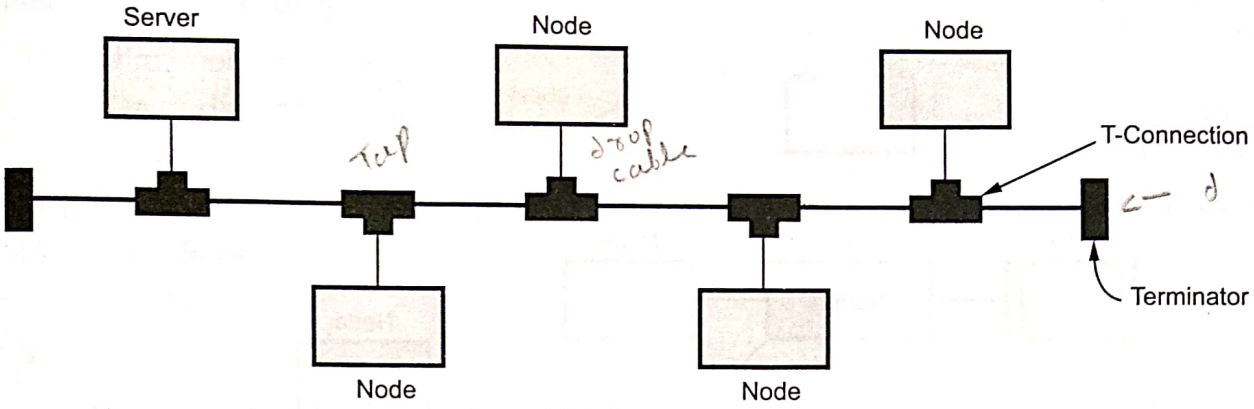


Fig. 1.3.1 Bus topology

- Bus is passive topology because it requires termination. Cable cannot be left unterminated in a bus network. Terminators were the 50 Ω resistors that were connected to each end of cable.

Advantages of Bus :

1) Easy to use and easy to install.
2) Needs fewer physical connectivity devices.
3) A repeater can also be used to extend a bus topology network.
4) Low cost.

Disadvantages of Bus :

1) Heavy network traffic can slow a bus considerably.
2) It is difficult to troubleshoot a bus.
3) Failure of cable affects all devices on the network.
4) Difficult to add new node.

1.3.2 Star Topology

- A star topology consists of a number of devices connected by point-to-point links to a central hub.
- Easy to control and traffic flow is simple.
- Data travels from the sender to central hub and then to the receiver.

Advantages of Star Topology :

1) It is easy to modify and add new nodes to a star network without disturbing the rest of the network.
2) Troubleshooting techniques are easy.
3) Failures of any node do not bring down the whole star network.

Scanned with CamScanner

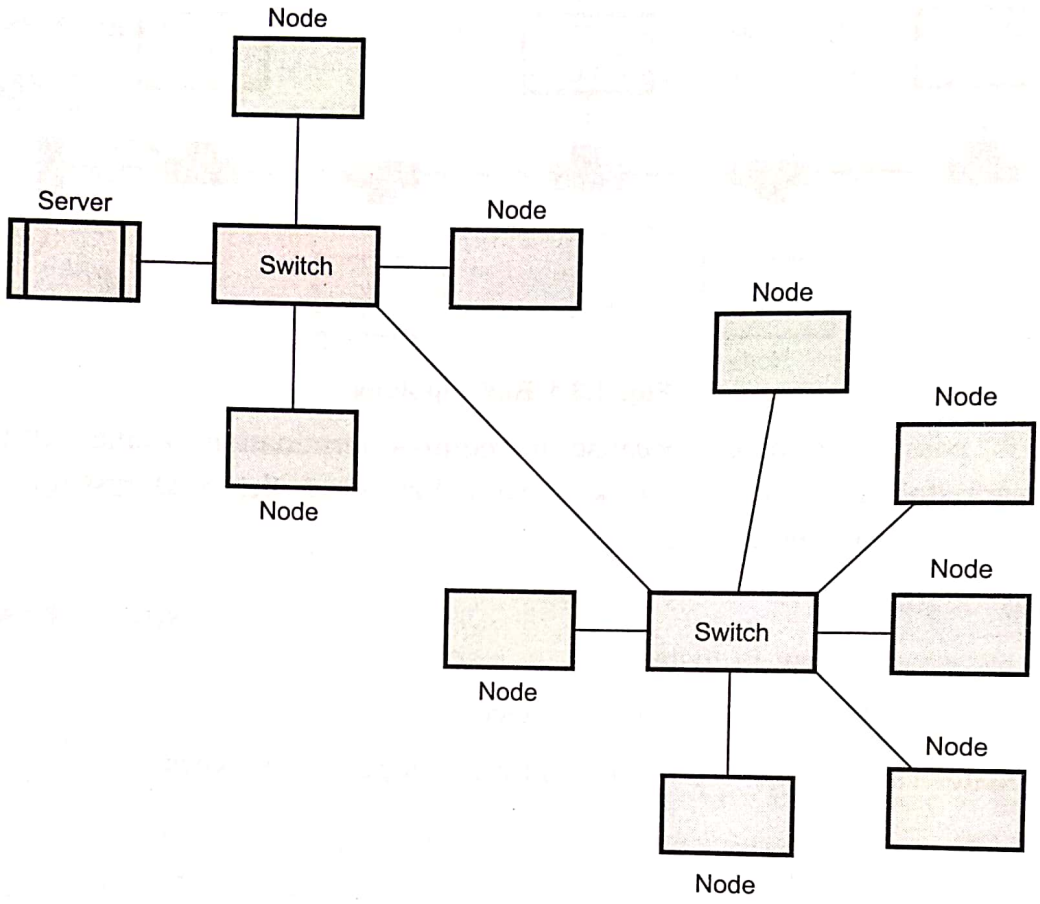


Fig. 1.3.2 Star topology

Disadvantages of Star Network :

1) If the central hub fails, the whole network fails to operate.
2) Each device requires its own cable segment.
3) Installation can be moderately difficult, especially in the hierarchical network.

1.3.3 Ring Topology

- In a ring topology, each computer is connected to the next computer, with the last one connected to the first. The signals travel on the cable in only one direction. Since each computer retransmits what it receives.
- Ring is an active network. Termination is not required.

Advantages of Ring :

1) Cable failures are easily found.
2) Because every node is given equal access to the token, no one node can monopolize the network.

Scanned with CamScanner

Disadvantages of Ring :

1) Adding or removing nodes disrupts the network.
2) It is difficult to troubleshoot a ring network.
3) Failure of one node on the ring can affect the whole network.
4) Cost of cable is more in ring network.

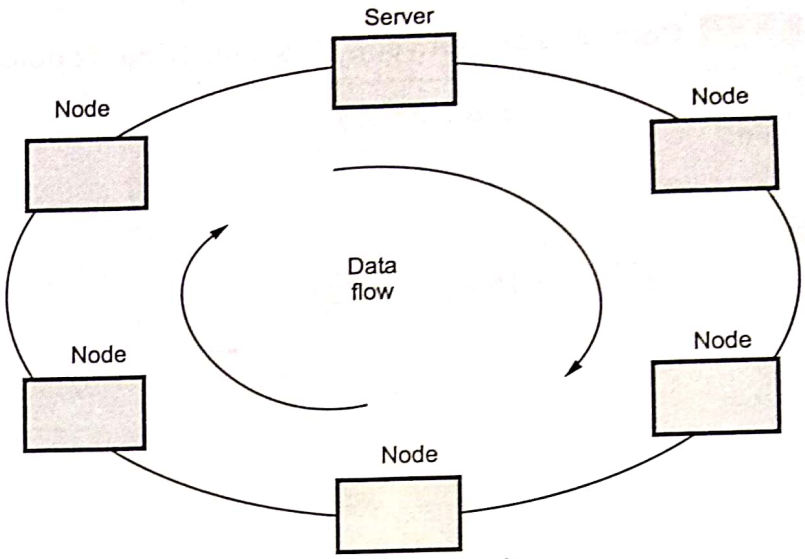


Fig. 1.3.3 Ring topology

1.3.4 Mesh Topology

- The mesh topology has a link between each device in the network. It is more difficult to install as the number of devices increases.
- Mesh networks are easy to troubleshoot.
- Much of the bandwidth available in mesh configuration is wasted.
- Most mesh topology networks are not true mesh networks. Rather, they are hybrid mesh networks, which contain some most important sites with multiple links.

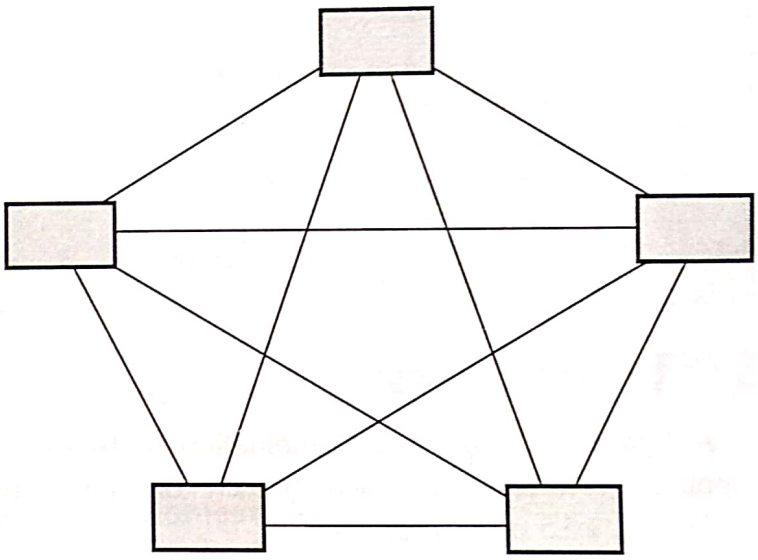


Fig. 1.3.4 True mesh topology

Advantages of Mesh :

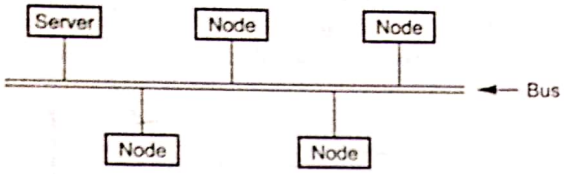
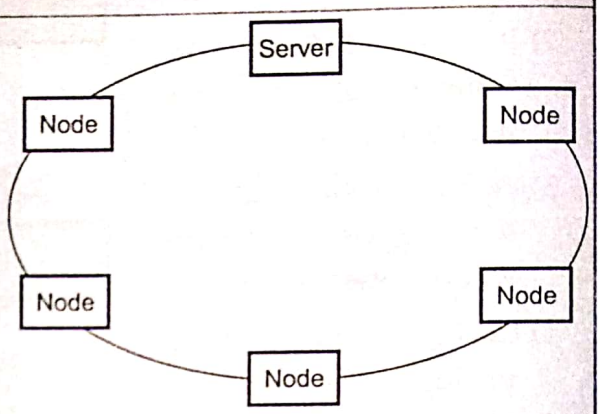
1) Troubleshooting is easy. 2) Isolation of network failures is easy.

Disadvantages of Mesh :

1) Difficulty of installation.
2) Costly because of maintaining redundant links.
3) Difficulty of reconfiguration.

Scanned with CamScanner

1.3.5 Comparison between Bus and Ring Topology

Sr. No.	Bus topology	Ring topology
1.		
2.	Bus requires proper termination. Cable cannot be left unterminated.	Termination is not required.
3.	Bus is a passive network topology.	Ring is an active network topology.
4.	There is loss in data integrity as the bus length increases.	Transmission errors are minimized because transmitted signal is regenerated at each node.
5.	It uses point to multipoint communication links.	It uses point-to-point communication links.
6.	Recommended when large number of devices are to be attached.	Recommended when moderate number of devices are to be attached.

1.3.6 Hybrid Topology

A hybird topology is a combination of two or more topologies. For example, bus topology connected in each branch of star network is shown in the Fig. 1.3.5. (Refer Fig. 1.3.5 on next page).

University Question

1. *What is topology ? Give different type of topology and its use.* **GTU : Dec.-11, Marks 5**

1.4 The Network Edge

1.4.1 End System, Clients and Servers

- In computer networks the computers connected to Internet are referred as **end systems**. The end systems are also called as **hosts** as they host application programs such as web browser program, a web server program an e-mail reader program.

Scanned with CamScanner

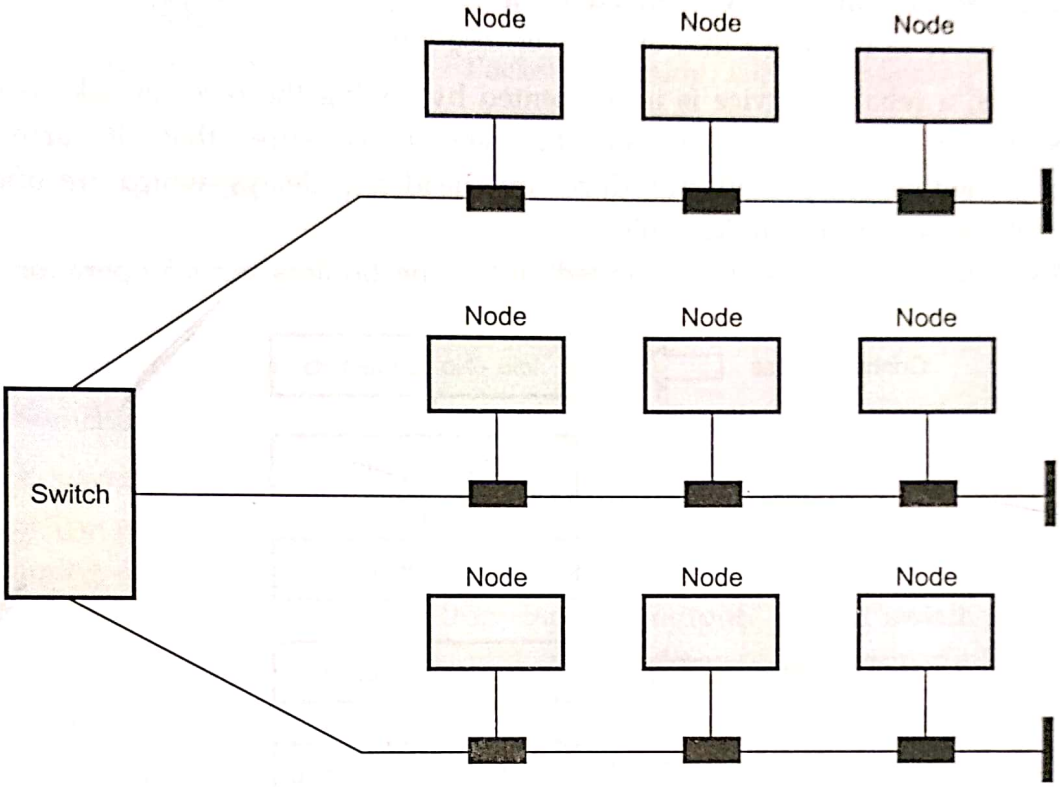


Fig. 1.3.5 Hybrid topology

- The hosts are further divided into clients and servers. Clients are desktop and mobile PCs whereas servers are more powerful devices such as web servers and mail servers.

Client program : A client program is a program running on one end system that requests and receives a service from a server program running on another end system.

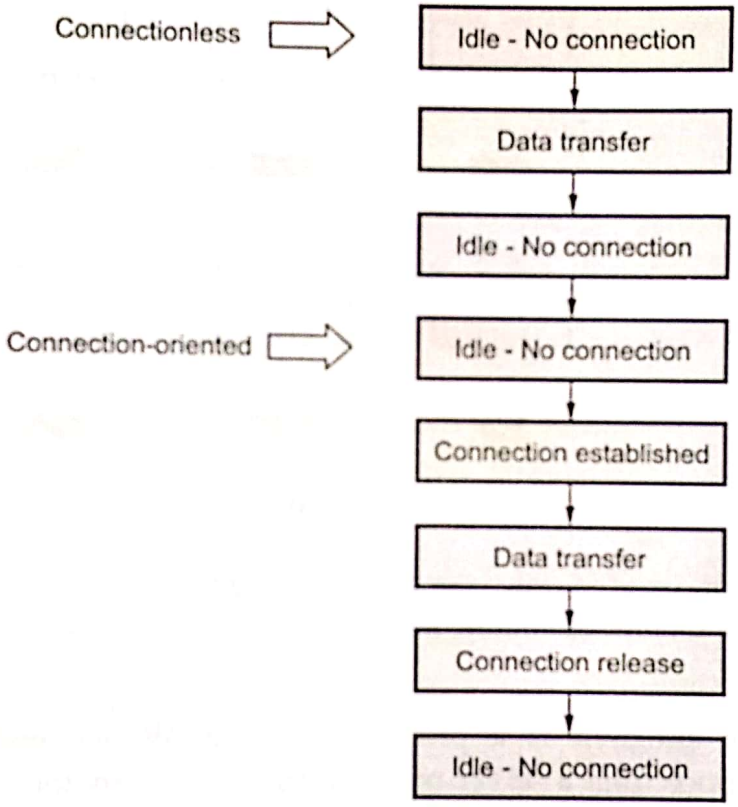
1.4.2 Connection Oriented and Connectionless Services

- Connection oriented and connectionless are the two types of services, that is offered by the layer.
- In **connection oriented,** direct path is established between source and destination. The telephone system is the example of the connection oriented service. This type of service provides a substantial amount of care for the user data.
- The **connectionless** (also called datagram) service goes directly from an idle condition into a data transfer mode, followed directly by the idle condition.
- The connectionless service is comparable to mailing a letter. Each message carries the full destination address, and each one is routed through the system independent of all the others.

Scanned with CamScanner

- Each service can be characterized by a Quality Of Service (QOS). Some services are reliable in the sense that they never lose data.
- Usually, a reliable service is implemented by having the receiver acknowledge the receipt of each message, so the sender is sure that it arrived. The acknowledgement process introduces overhead and delays, which are often worth it but are sometimes undesirable.

Fig. 1.4.1 shows the connection oriented and connectionless service operation.



Connectionless ⇨ No Acks, no flow control and no error control

Connection-oriented ⇨ Acks, flow control, error recovery

Fig. 1.4.1 Connectionless and connection oriented service

University Question

1. *Give difference between connection oriented versus connectionless services.*

GTU : Winter-13,14, Marks 7

1.5 Network Core

- The **network core** is referred as the mesh of routers that interconnect the Internet's end systems.

Scanned with CamScanner

- Different switching techniques are used for data transmission within the network. Basic methods of switching are : Packet switching, Circuit switching and Message switching.

1.5.1 Switching Fabric

- Component of packet switching are as follows :
  1. Input ports
  2. Number of output ports
  3. Switching fabric
  4. Routing processor
- Capacity of switch is the maximum rate at which it can move information, assuming all data paths are simultaneously active. Circuit switch must reject call if cannot find a path for samples from input to output . Packet switch must reject a packet if it can find a buffer to store it awaiting access to output trunk.
- Fig. 1.5.1 shows packet switch components.
- In packet switch, physical and data link function are performed by input ports. It constructs the packets from bits and perform error correction and detection.

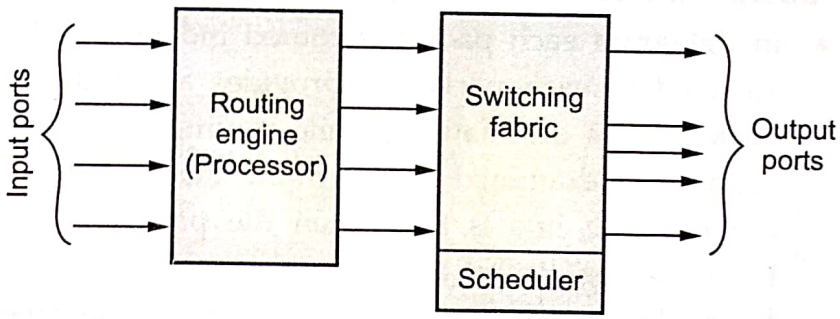


Fig. 1.5.1 Packet switch components

- It transfer data from input to output. It usually consists of links and switching elements.
- The routing engine looks-up the packet address in routing table and determines which output port to send the packet. It performs functions of network layer. Each packet is tagged with port number. The switch uses the tag to send the packet to the proper output port.
- Simplest switch fabric is a shared bus. Switch fabrics are created from certain building blocks of smaller switches arranged in stages.
- The simplest switch is a 2×2 switch, which can be either in the through or crossed position.

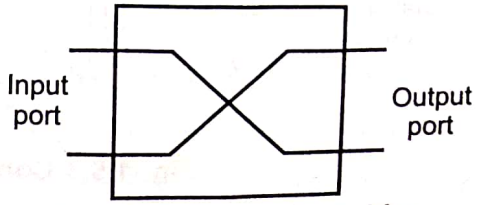


Fig. 1.5.2 Crossed position

Scanned with CamScanner

1.5.2 Packet Switching

- Packet switching is often used in computer networks where individual users have need of the channel intermittently. While using the channel the application requires high bandwidth, but most of the time, each user does not require that channel at all. Such applications, characterized by a high peak to average requirement for capacity, are called **bursty** and are ideal for packet switching.
- In packet switching, messages are broken into short blocks and interleaved with other messages. Thus, users queue for the channel and share it with one another efficiently. Data is sent in individual packets. Each packet is forwarded from switch to switch, eventually reaching its destination. Each switching node has a small amount of buffer space to temporarily hold packets. If the outgoing line is busy, the packet stay in queue until the line becomes available. Packet switching handles burstly traffic well.
- Packet switching method uses two routing approaches :
  1. **Datagram** and 2. **Virtual circuit.**

1) Datagram Packet Switching

- In **datagram** each packet is routed independently through the network. Header is attached to each packet. It provides all of the information required to route the packet to its destination. While routing the packet, the destination address in the header are examined to determine the next hop in the path to the destination. If the required line is busy then the packet is placed in the queue until the line becomes free. Packet share the transmission line with other packets. Then it deliver to the destination. Datagram approach is also called **connectionless**.
- Disadvantage of datagram approach is a lot of overhead because of independent routing. Another disadvantage is that packet may not arrive in the order at destination in which they were sent.
- Since each packet is routed independently, packets from the same source to the same destination may traverse through different paths. This is shown in Fig. 1.5.3.

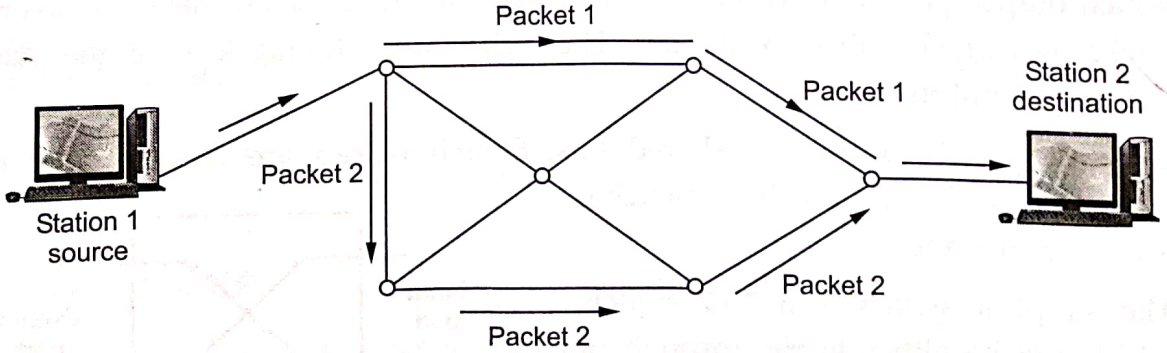


Fig. 1.5.3 Connectionless packet switching

Scanned with CamScanner

- The packets at station 2 or destination may arrive out of order, and resequencing may be required at the destination. At each node a routing table is maintained which specifies the next hops that is to be taken by packets for the given destination.

2) Virtual Circuit Packet Switching

- In **virtual circuit packet switching** a fixed path between a source and a destination is established prior to transfer of packets.
- Connection-oriented network is also known as **virtual circuit**. Virtual circuit is similar to telephone system. A route, which consists of a logical connection is first established between two users. The connection that is established is not a dedicated path between stations. The path is generally shared by many other virtual connections.
- The process is completed in three main phases -
  i) Establishment phase.
  ii) Data transfer phase.
  iii) Connection release phase.

i) Establishment phase :

- During setting up of logical connection, the two users not only agree to setup a connection between them but also decide upon the quality of service associated with the connection. After this the sequences of packetized information are transmitted bidirectionally between the nodes. The information is delivered to the receiver in the same order as transmitted by sender.

ii) Data transfer phase :

- During this phase it performs flow control and error control services.
- The error control service ensures correct sequencing of packets and correct arrival of packets.
- Flow control service ensures a slow receiver from being overwhelming with data from a faster transmitter.

iii) Connection release :

- When the station wish to close down the virtual circuit, one station can terminate the connection with a clear request packet. Fig. 1.5.4 shows the virtual circuit packet switching.

Scanned with CamScanner

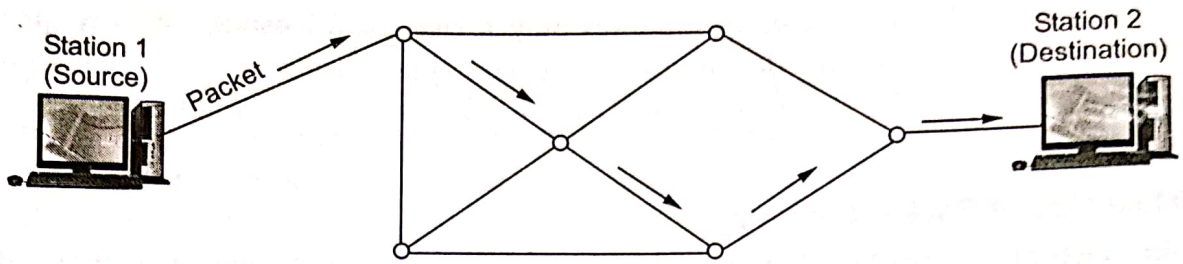


Fig. 1.5.4 Virtual circuit packet switching

1.5.2.1 Advantages of Packet Switching

1. Uses resources more efficiently.
2. Very little setup or tear down time.
3. It is more flexible. i.e. packets can be routed through any switching node.
4. Improved bandwidth.
5. Small sized packet reduces transmission delay.

1.5.2.2 Disadvantages of Packet Switching

1. Complex protocol for packet switching.
2. Algorithms are more complicated.
3. Difficults to bill customers.
4. Switching processor must be powerful.
5. Packets may lost during switching.

1.5.3 Circuit Switching

- The telephone system as it historically developed was designed for voice and analog signals. Sending data requires bandwidth. The amount of bandwidth needed is directly related to the data rate that is desired. An analog voice signal contains its data in a relatively narrow bandwidth, in proportion to the amount of data it carries.
- For voice signals, a relatively large amount of distortion is acceptable, since the human ear can understand voice even with distortion that looks severe to the eye. For digital signals, these distortions may cause the receiver to misinterpret the signal that is sent and so produce an error. The regular telephone loop from the local office to the phone is guaranteed by the phone company to have some specific characteristics. This type of line is the lowest performance line, called voice grade conditioning.
- Similar line characteristics are offered by telephone companies on the lines that go between phone company offices. These interoffice lines are called trunks. Any phone line can connect one user to another user through the phone system, the

Scanned with CamScanner

user has a line assigned randomly, through the phone offices. This is called the dial-up or switched network.

- Telephone networks are connection oriented because they require the setting up of connection before the actual transfer of information can take place.
- An end-to-end path setup beginning of a session, dedicated to the application, and then released at the end of session. This is called **circuit switching**. Circuit switching is effective for application which make comparatively steady use of channel. Fig. 1.5.5 shows the circuit switching.

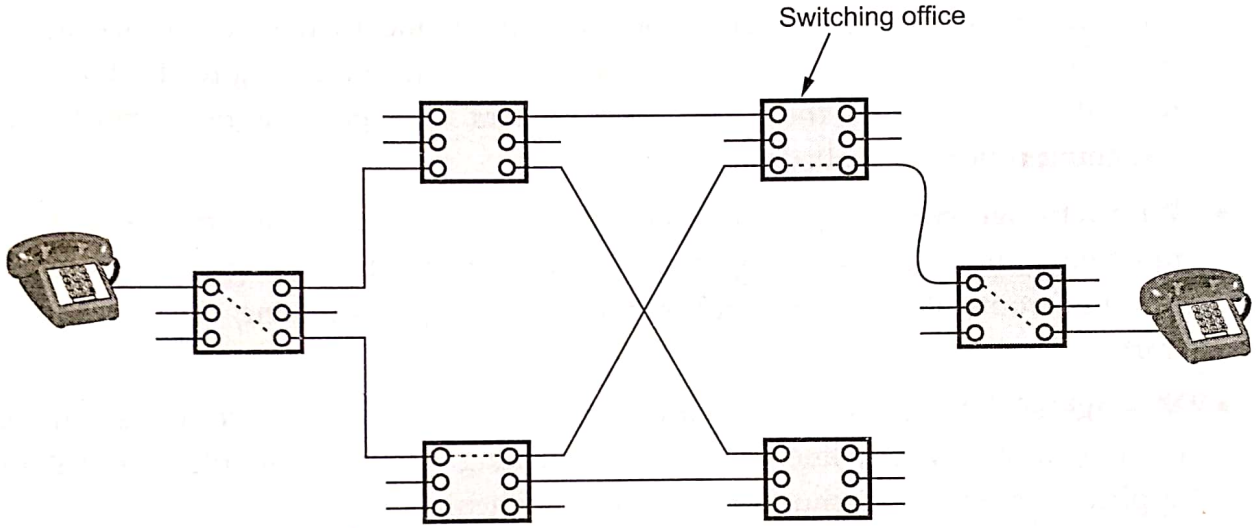


Fig. 1.5.5 Circuit switching

For application which need greater performance than these dial up lines can offer, telephone companies offer specially conditional lines. These lines both from the phone to the office and between phone offices, provide better frequency response and time delay characteristics. This kind of conditioned line is leased by the user. The term dedicated and leased are used when the phone company has set a side a conditional line for a communications link.

Advantages of Circuit Switching

1. Fixed bandwidth, guaranteed capacity.
2. Low variance end to end delay.

Disadvantages

1. Connection setup and tear down introduces extra overhead.
2. User pay for circuit, even when not sending data.
3. Other user cannot use circuit even if it is free of traffic.

Scanned with CamScanner

1.5.4 Message Switching

- Message switching is used to describe the telegraph network. When this form of switching is used, no physical copper path is established in advance between sender and receiver. 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 hope at a time. Each block is received in its entirely, inspected for errors, and then transmitted. A network using this technique is called a **store and forward network.**
- The message was punched on paper tape off line at the sending office and then read in and transmitted over a communication line to the next office along the way, where it was punched out on paper tape. An operator tore the tape off and read it in on one of the many tape readers, one per outgoing trunk. Such a switching office was called a torn tape office.
- With message switching, there is no limit on block size, which means that routers must have disks to buffer long blocks. It also means that a single block may tie up a router, router line for minutes, rendering message switching uses for interactive traffic.
- Message switching does not involve a call setup. It can achieve a high utilization of the transmission line. Message switching is not suitable for interactive applications. Fig. 1.5.6 shows the message switching.

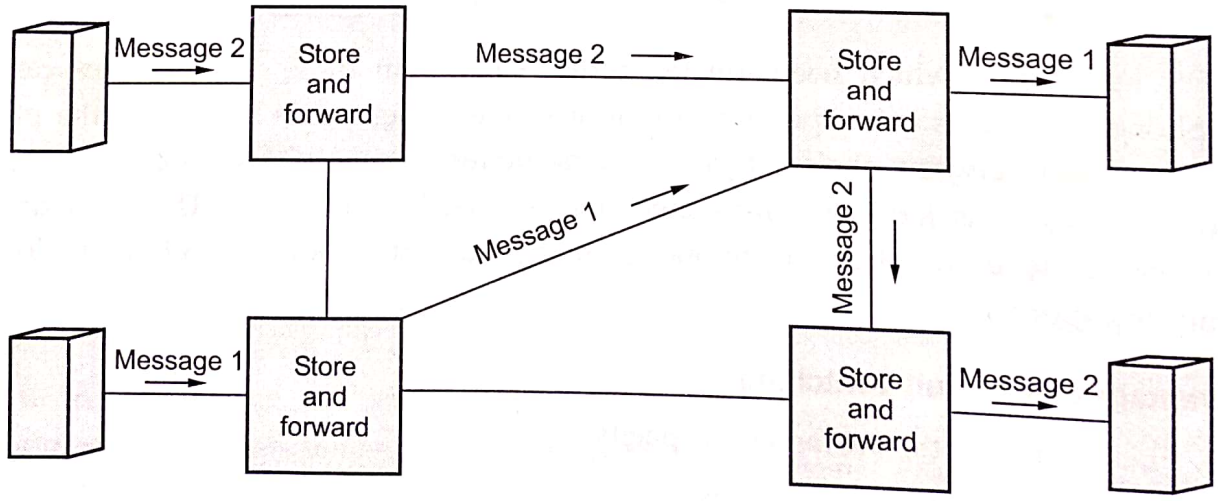


Fig. 1.5.6 Message switching

Advantages of Message Switching

1. Efficient traffic management.
2. Reduces network traffic congenstion.
3. Efficient use of transmission channel.

Scanned with CamScanner

Disadvantages of Message Switching

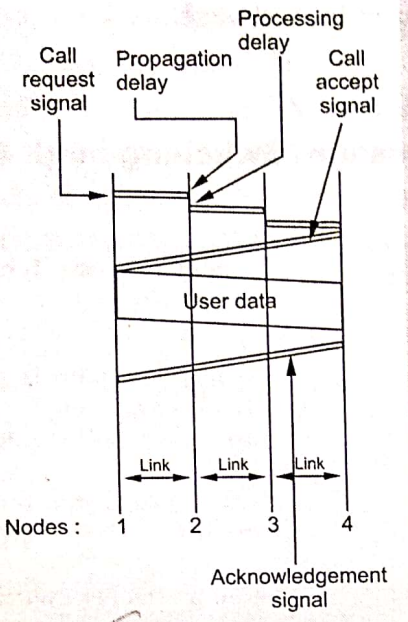
1. Because of store and forward, transmission delay is in deduced.
2. Each node requires large capacity for storing.

1.5.5 Comparison of Packet Switching, Message Switching and Circuit Switching

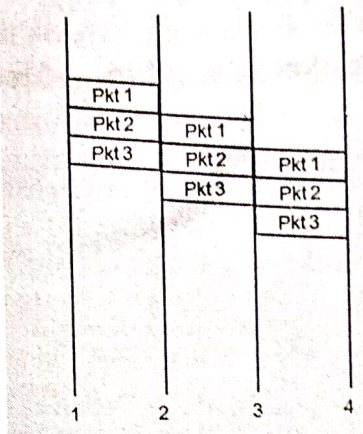
Sr. No.	Circuit switching	Packet switching	Message switching
1.	There is physical connection between transmitter and receiver.	No physical path is established between transmitter and receiver.	No physical path is set in advance between transmitter and receiver.
2.	All the packet uses same path.	Packet travels independently.	Packets are stored and forward.
3.	Needs an end to end path before the data transmission.	No needs of end to end path before data transmission.	Same as packet switching.
4.	Reverses the entire bandwidth in advance.	Does not reserve the bandwidth in advance.	Same as packet switching.
5.	Charge is based on distance and time, but not on traffic.	Charge is based on both number of bytes and connect time.	Charge is based on number of bytes and distance.
6.	Waste of bandwidth is possible.	No waste of bandwidth.	No waste of bandwidth.
7.	Congestion occur for per minute.	Congestion occurs for per packet.	No congestion or very less congestion.
8.	It cannot support store and forward transmission.	It support store and forward transmission.	It also support store and forward transmission.
9.	Not suitable for handling interactive traffic.	Suitable for handling interactive traffic.	Same as circuit switching.
10.	Recording of packet can never happen with circuit switching.	Recording of packet is possible.	Same as packet switching.

Scanned with CamScanner

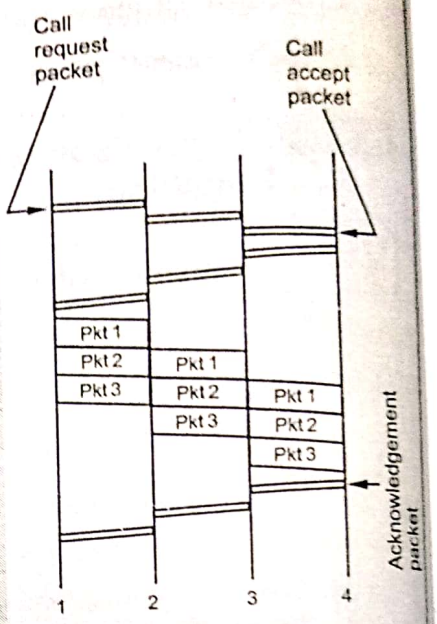
11. Timing diagram



Timing diagram



Timing diagram



1.6 Delay and Loss in Packet-Switched Networks

- A packet during its travel from one node to the subsequent node (host or router) it suffers from different types of delays at each node.
- Some inportant types of delays are :
  1. Processing delay
  2. Queuing delay
  3. Transmission delay
  4. Propagation delay
- All delays accumulated together and result in a larger delay called total nodal delay.

1.6.1 Processing Delay

- Processing delay is a nodal delay and it is defined as the time required examining the packet's header and determining where to direct the packet.
- The processing delay is denoted by d_{proc}.
- Processing delay also include delay due to the time needed to check for bit-level errors in the packet that occurred in transmitting the packet's bits from the upstream router to other router.

Scanned with CamScanner

1.6.2 Queuing Delay

- After nodal processing delay, the router directs the packet to the queue that precedes the link to subsequent router.
- The queuing delay is denoted by d_{queue}.
- The queuing delay is observed at the queue, the packet experiences a **queuing delay** as it waits to be transmitted over the link.
- The queuing delay of a specific packet depends on earlier-arriving packets that are queued and waiting for transmission across the link.
- The delay of a packet can vary significantly from packet to packet. If the queue is empty and no other packet is currently being transmitted, then packet's queuing delay zero.
- When the traffic heavy and various other packets are waiting to be transmitted, the queuing delay will be long..

1.6.3 Transmission Delay

- The **transmission delay** is defined as the amount of time required to transmit all of the packet bits over the link.
- The transmission delay is denoted by d_{trans}.
- The transmission delay is also called as store-and-forward delay.
- The transmisssion delay is expressed as the ratio of packet length (bits) to transmission rate of the link (bits/sec).

$$\text{Transmission delay} = \frac{\text{Packet length}}{\text{Transmisssion rate}} = \frac{L}{R}$$

1.6.4 Propagation Delay

- The propagation delay is defined as time required by a packet to propagate from transmitting node to the receiving node.
- The propagation delay is denoted by d_{prop}.
- The propagation speed of a packet depends on characteristic of physical medium of the link and the distance between the nodes.

1.6.5 Total Nodal Delay

- The total nodal delay exprienced by a packet is sum of processing delay, queuing delay, transmission delay and propagation delay within a network.
- The total delay is very significant parameter of a network.

Scanned with CamScanner

$$\text{Total nodal delay} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

1.7 Protocol Layers and Their Service Models

- A computer network must provide general, cost effective, fair and robust connectivity among a large number of computers. Designing a network to meet these requirements is no small task.
- To deal with this complexity, network designers have developed general blue prints - usually called network architectures. It guides the design and implementation of networks.

1.7.1 Layered Architecture

- Computer network is designed around the concept of layered protocols or functions. For exchange of data between computers, terminals or other data processing devices, there is data path between two computers, either directly or via a communication network.
- Following factors should be considered.
  1. The source system must either activate the direct data communication path or inform the communication network to the identity of the desired destination system.
  2. Provide for standard interface between network functions.
  3. Provide for symmetry in function performed at each node in the network. Each layer performs the same functions as its counter part in the other node of network.
- The network software is now highly structured.

1.7.2 Protocol Hierarchies

- Most of all networks are organized as a series of layers, each one built upon the one below it. Because of layer, it reduces the design complexity.
- In layer protocols, a layer is a service provider and may consists of several service functions. Function is a sub system of a layer.
- Each subsystem may also be made up of entities. An entity is a specialized module of a layer or subsystem.
- Name of the layer, total number of layer, function and content of each layer differ from network to network.
- Protocols are the rules that govern network communication.

Scanned with CamScanner

- Fig. 1.7.1 shows the five layer network.

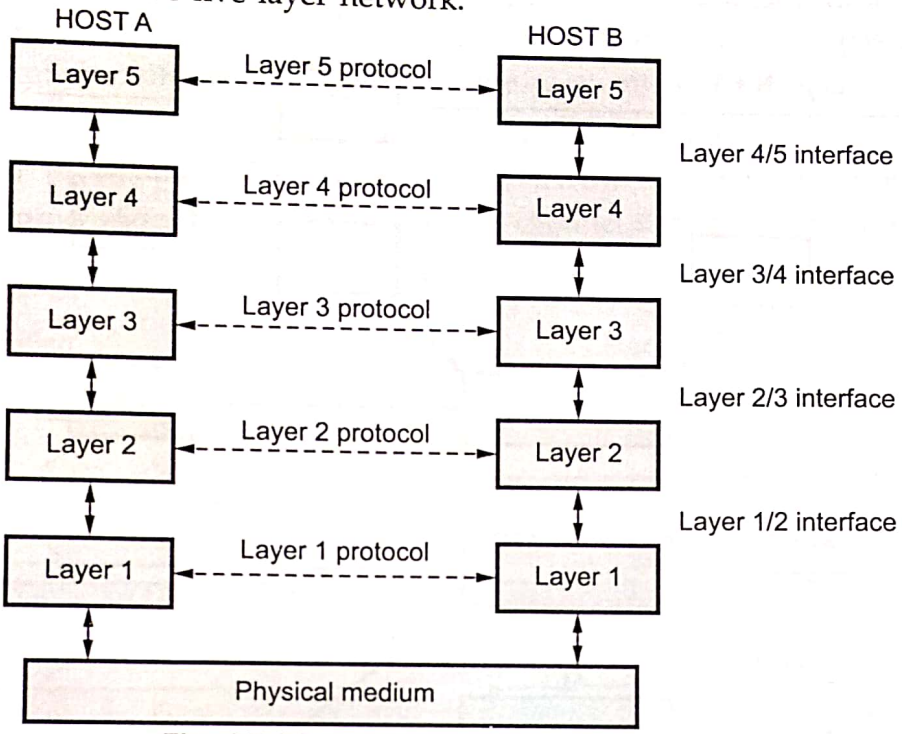


Fig. 1.7.1 Layers, protocols and interfaces

- Layer n on one node carries on a conversation with layer n on other node.
- The entities comprising the corresponding layers on different machine are called **peers.**
- The actual data flow is from upper layer to its below layer and then from physical medium to destination layer.
- Between each pair of adjacent layers is called **interface.** The interface defines which primitive operations and services the lower layer offers to the upper one.
- A set of layers and protocols is called a **network architecture.**

1.7.3 Interfaces and Services

- The process provides a common technique for the layer to communicate with each other. The standard terminology used for layered networks to request services is provided.
- In Fig. 1.7.2 the layers N+1, N and N–1 are involved in the communication process for layer communication, with each other.
- Following components are involved and their function is as follows :
  1. Service Data Unit (SDU)
  2. Protocol Control Information (PCI)
  3. Protocol Data Unit (PDU)
  4. Interface Control Information (ICI)
  5. Interface Data Unit (IDU)

Scanned with CamScanner

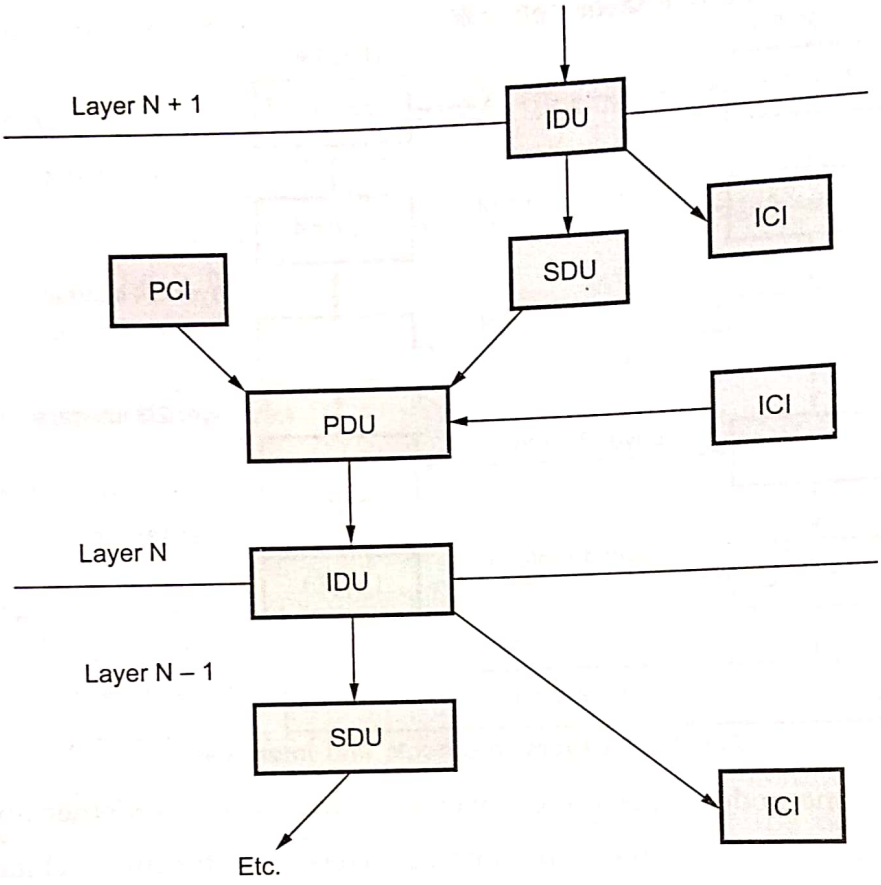


Fig. 1.7.2 Communication between layers

Sr. No.	Name	Function
1.	SDU	Transfer user data by layer N+1 to layer N and N–1.
2.	PCI	To perform service function, it is used to exchange information by peer entities at different sites on the network.
3.	PDU	Combination of the SDU and PCI.
4.	ICI	It passes temporary parameter between N and N–1 to invoke service function.
5.	IDU	The total unit of information transferred across the layer boundaries.

- When the IDU from layer N+1 passes to layer N, it becomes the SDU to that layer. PCI is added to SDU at layer N. ICI performs its function and is discarded. Another ICI is added to PDU at layer N and it becomes IDU to layer N–1. Thus a full protocol unit is passed through each layer.
- Each layer adds header to data. This header is used by the peer layer entity at another node of the network to invoke function. This process repeats itself through each layer.

Scanned with CamScanner

- As each unit traverses through the layer, it has a header added to it i.e. user data and header (SDU and PCI). This full protocol data unit is passed onto the communication path, where it arrives at the receiving site.
- In short, each layer added its header to user's data and passes to its next layer. This layer process on that data and adds its own header and provides to next layer for processing. Through transmission channel data passes to receiving site.
- Fig. 1.7.3 shows the communication between two sites in a network.

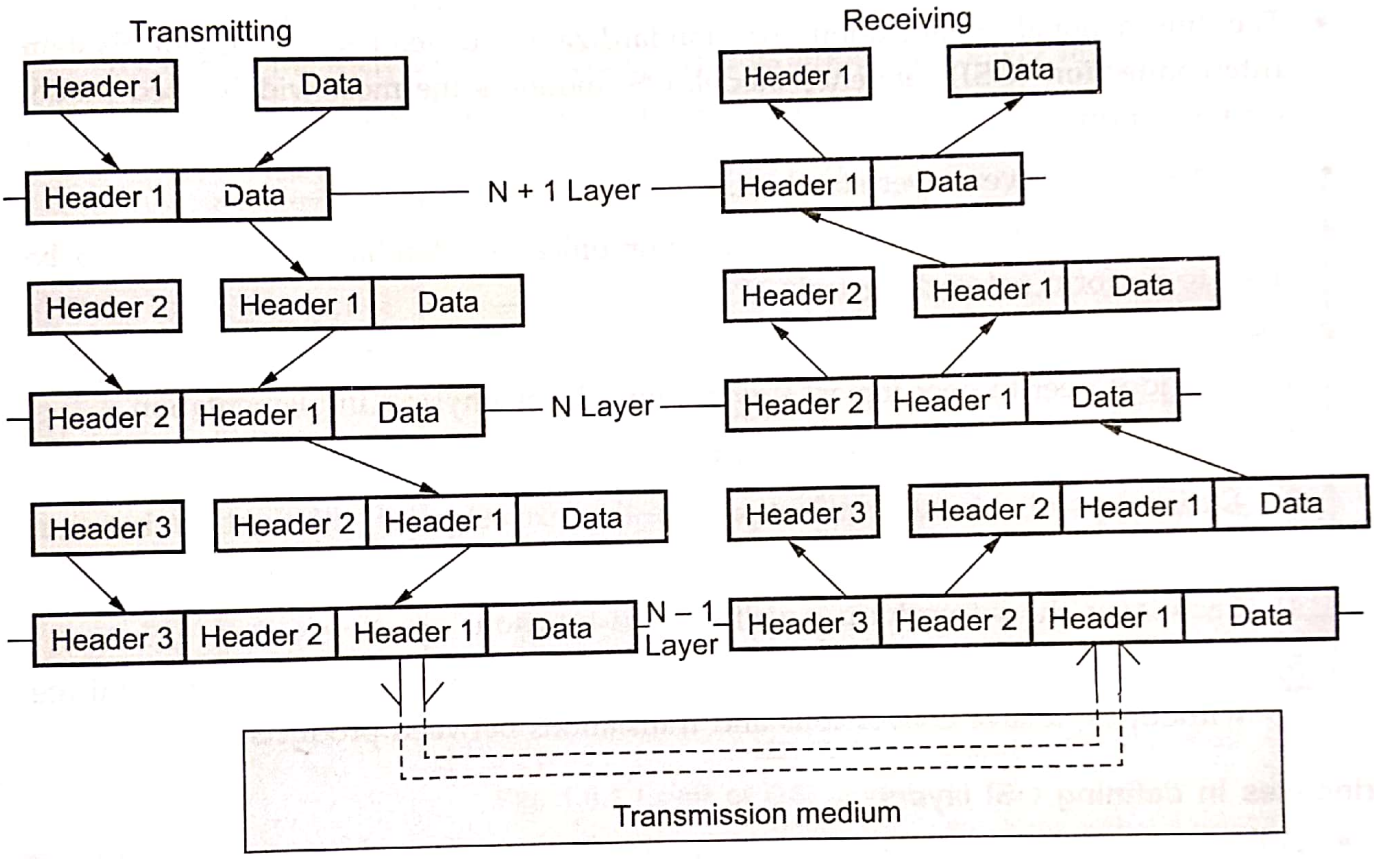


Fig. 1.7.3 Communication between two sites in a network

1.7.4 Relationship of Services to Protocols

- Service interface provides an entry point that users use to access the functionality exposed by the application.
- Service interface is usually network addressable.
- Service interface provides a much more coarse-grained interface while preserving the semantics and finer granularity of the application logic. It also provides a barrier that enables the application logic to change without affecting the users of the interface.
- The service interface should encapsulate all aspects of the network protocol used for communication between the user and service. For example, suppose that a

Scanned with CamScanner

service is exposed to consumers through HTTP over a TCP/IP network. User can implement the service interface as an ASP.NET component published to a well-known URL.

1.8 OSI Reference Model

GTU : Winter-13,14, Dec.-10,11, Summer-14

- The ISO was one of the first organizations to formally define a common way to connect computers. Their architecture, called the Open System Interconnection (OSI).
- The International organization for standardization developed the **Open System Interconnection (OSI)** reference model. OSI model is the most widely used model for networking.
- OSI model is a seven layer standard.
- The OSI model does not specify the communication standard or protocols to be used to perform networking tasks.
- OSI model provides following services.
  1) Provides peer-to-peer logical services with layer physical implementation.
  2) Provides standards for communication between system.
  3) Defines point of interconnection for the exchange of information between system.
  4) Each layer should perform a well defined function.
  5) Narrows the options in order to increase the ability to communicate without expansive conversions and translations between products.

Principles in defining OSI layers

- Following principles are used in defining the OSI layers.
  1. Do not create so many layers as to make the system engineering task of describing and integrating the layers more difficult than necessary.
  2. Create a boundary at a point where the description of services can be small and the number of interrelations across the boundary are minimized.
  3. Create separate layers to handle function that are manifestly different in the process performed.
  4. Collect similar functions into the same layer.
  5. Select the boundaries at a point which past experience has demonstrated to be successful.
  6. Create a layer of easily localized functions so that the layer could be totally redesigned and its protocols changed in a major way to take advantage of new

Scanned with CamScanner

advances in architecture, hardware or software technology without changing the services expected from and provided to the adjacent layers.

7. Create a boundary where it may be useful at some points in time to have the corresponding interface standardized.
8. Create a layer where there is a need for a different level of abstraction in the handling of data.
9. Allow changes of functions or protocols to be made within a layer without affecting other layers.
10. Create for each layer boundaries with its upper and lower layer only.

- Fig. 1.8.1 shows the OSI 7 layer reference model.

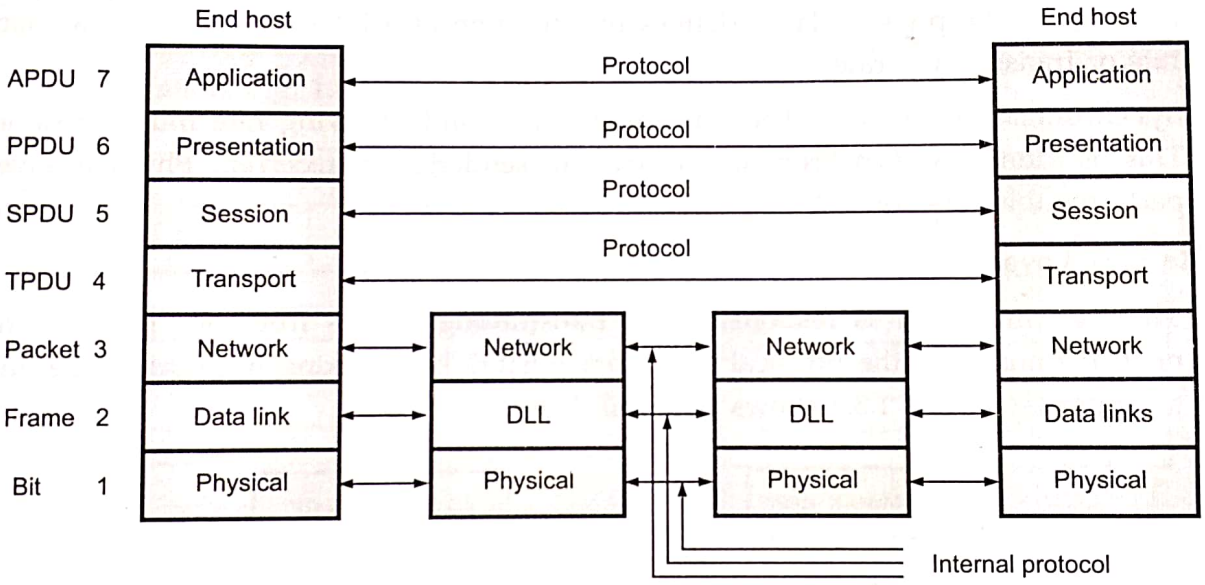


Fig. 1.8.1 Layer of OSI model

1.8.1 Layers in OSI Models

1. Physical Layer

- Physical layer is the lowest layer of the OSI model. Physical layer co-ordinates the functions required to transmit a bit stream over a communication channel. It deals

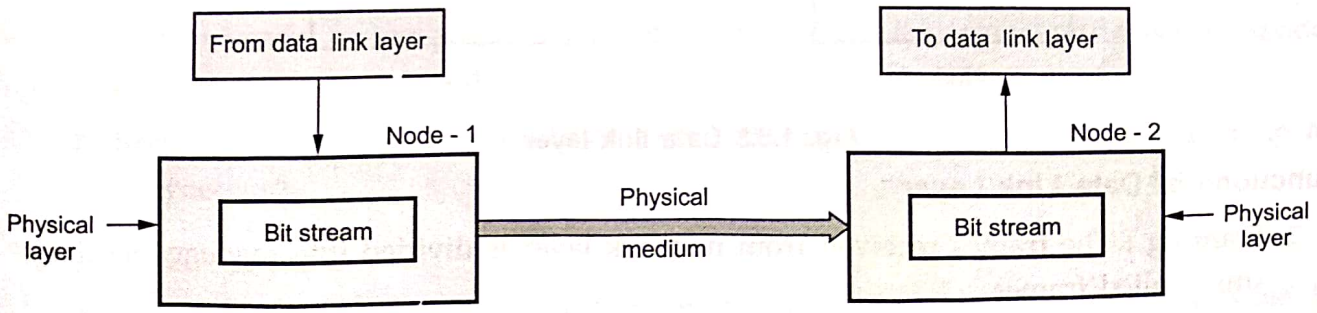


Fig. 1.8.2 Physical layer

Scanned with CamScanner

with electrical and mechanical specifications of interface and transmission media. It also deals with procedures and functions required for transmission.

- The position of physical layer with transmission medium and the next layer (data link layer) is shown in Fig. 1.8.2

Functions of Physical Layer

1. **Physical characteristics of interfaces and media :** The design issue of physical layer considers the characteristics of interface between devices and transmission media.
2. **Representation of bits :** Physical layer encodes the bit stream into electrical or optical signal.
3. **Data rate :** The physical layer defines the duration of a bit which is called as data rate or transmission rate.
4. **Synchronization of bits :** The transmission rate and receiving rate must be same. This is done by synchronizing clocks at sender and receiver. Physical layer performs this function.

2. Data Link Layer

- The data link layer is responsible for transmitting frames from one node to the next. It transforms the physical layer to a reliable link making it an error free link to upper layer. Fig. 1.8.3 shows data link layer.

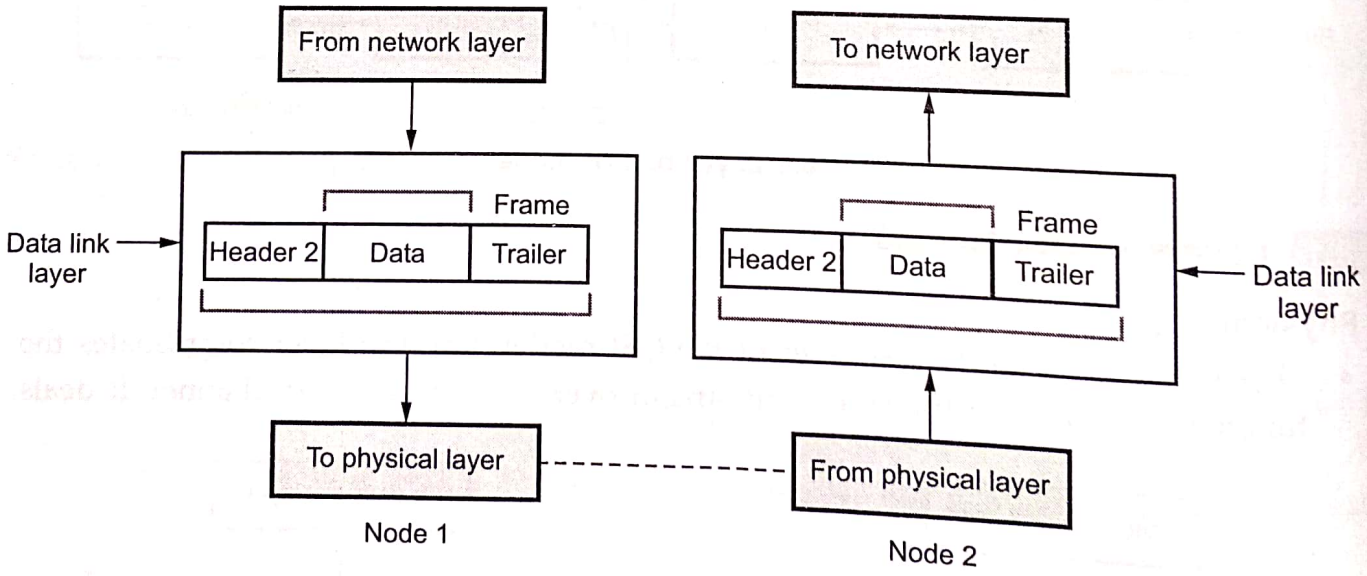


Fig. 1.8.3 Data link layer

Functions of Data Link Layer

1. **Framing :** The frames received from network layer is divided into manageable data units called frames.

Scanned with CamScanner

2. **Physical addressing :** When frames are to be sent to different LANs, the data link layer adds a header to the frame to define sender or receiver.
3. **Flow control :** When the rate of the data transmitted and rate of data reception by receiver is not same, some data may be lost. The data link layer imposes a flow control mechanism to prevent overwhelming the receiver.
4. **Error control :** Data link layer incorporates reliability to the physical layer by adding mechanism to detect and retransmit damaged or lost frames.
5. **Access control :** When multiple devices are connected to same link, the data link layer determines which device has control over link.

3. Network Layer :

- The network layer is responsible for the delivery of packets from the source to destination. Fig. 1.8.4 shows network layer.

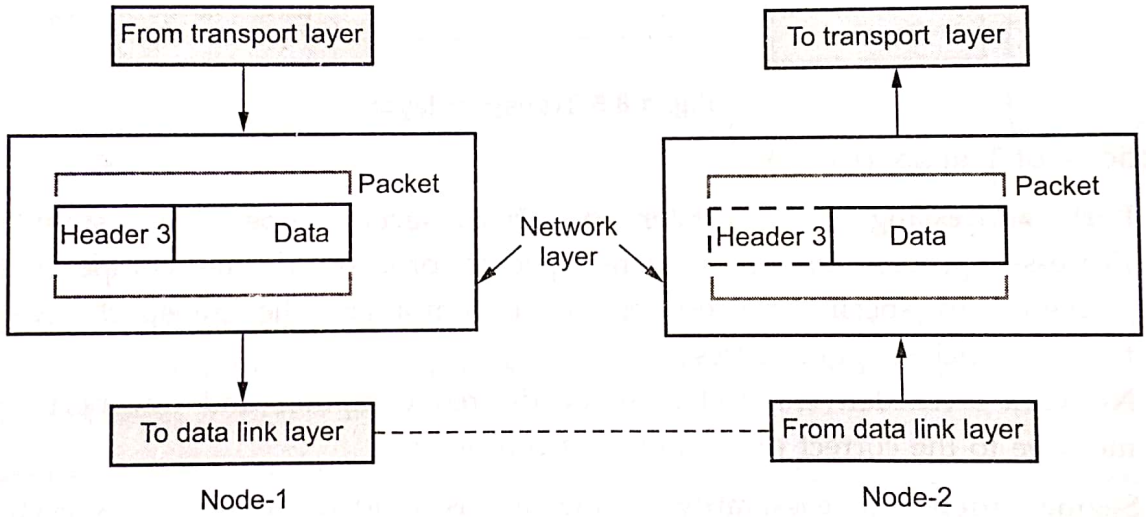


Fig. 1.8.4 Network layer

Functions of Network Layer

1. **Logical addressing :** Data link layer implements physical addressing. When a packet passes network boundary, an addressing system is needed to distinguish source and destination, network layer performs these function. The network layer adds a header to the packet of upper layer includes the logical addresses of sender and receiver.
2. **Routing :** Network layer route or switch the packets to its final destination in an internetwork.

4. Transport Layer :

- The transport layer is responsible for delivery of message from one process to another. The network does the host to destination delivery of individual packets

Scanned with CamScanner

considering it as independent packet. But transport layer ensures that the whole message arrives intact and in order with error control and process control. Fig. 1.8.5 shows transport layer.

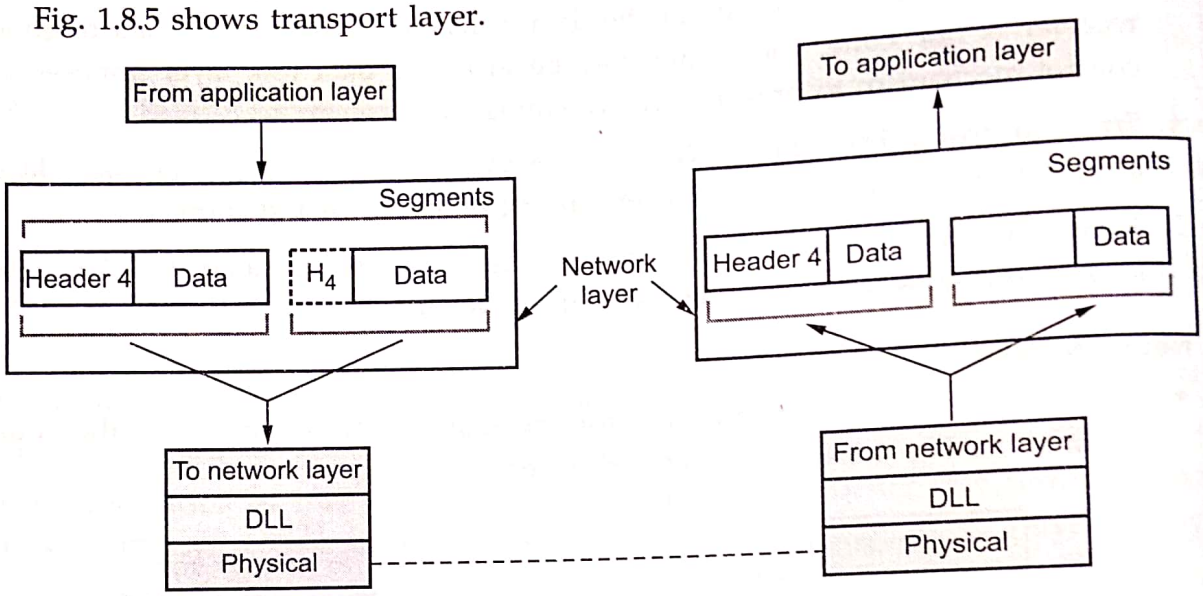


Fig. 1.8.5 Transport layer

Functions of Transport Layer

1. **Port addressing :** Computer performs several operations simultaneously. Process-to-process delivery means specific process of one computer must be delivered to specific process on other computer. The transport layer header therefore include port address.
   Network layer delivers packet to the desired computer and transport layer, gets message to the correct process on that computer.
2. **Segmentation and reassembly :** A message is divided into segments, each segment contains a sequence number which enables transport layer to reassemble at destination.
3. **Connection control :** Transport layer performs connectionless or connection oriented services with the destination machine.
4. **Flow control :** Transport layer performs end-to-end flow control while data link layer performs it across the link.
5. **Error control :** Error control at this layer is performed on end-to-end basis rather than across the link. The transport layer ensures error free transmission.

5. Session Layer :

- The session layer is network dialog controller i.e. it establishes and synchronizes the interaction between communication system. Fig. 1.8.6 shows session layer.

Scanned with CamScanner

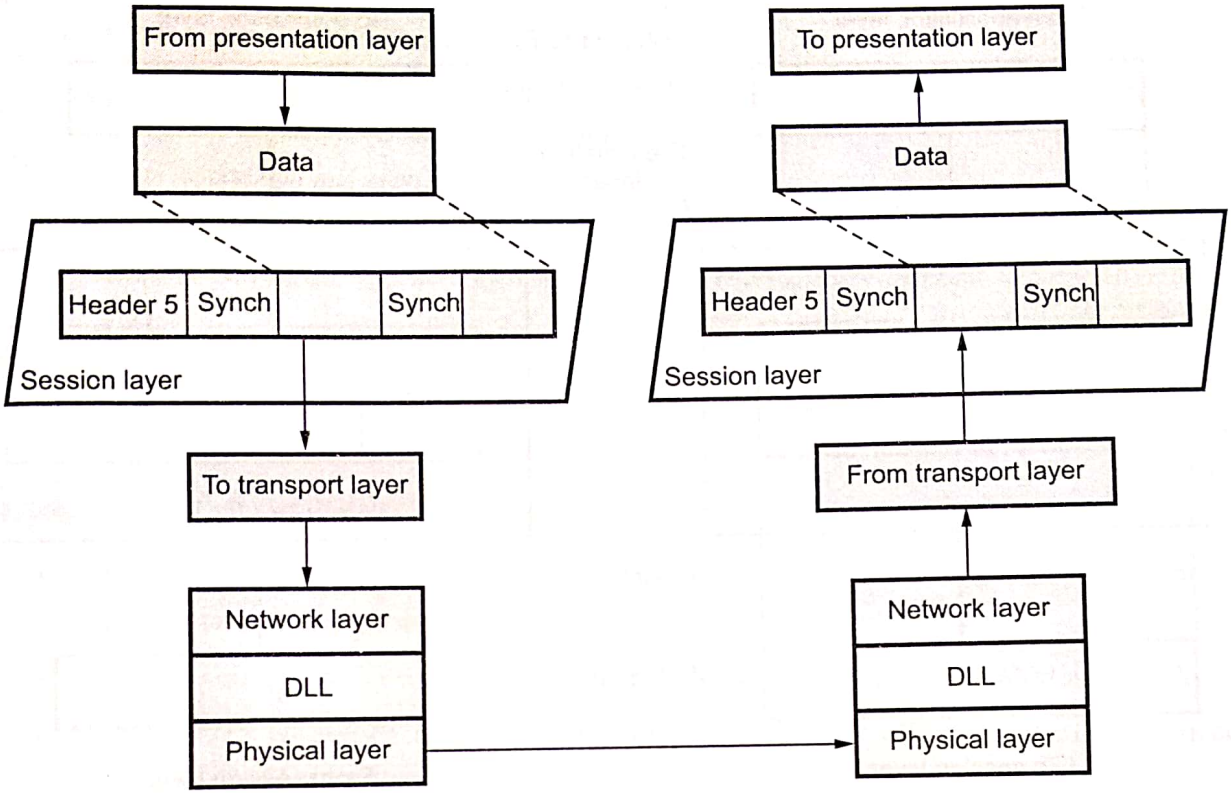


Fig. 1.8.6 Session layer

Functions of Session Layer

1. **Dialog control :** Communication between two processes take place in either half duplex or full-duplex mode. The session layer manages dialog control for this communication.
2. **Synchronization :** Session layer adds synchronization points into stream of data.

6. Presentation Layer :

- The presentation layer deals with syntax and semantics of the information being exchanged. Fig. 1.8.7 shows presentation layer.

Functions of Presentation Layer

1. **Translation :** Different computers use different encoding systems. The presentation layer maintains interoperability between the two encoding systems.
2. **Encryption :** Encryption is transforming sender information to other form to ensure privacy while transmission. Decryption is a reverse process.
3. **Compression :** Compression is a technique of reducing number of bits required to represent the data.

Scanned with CamScanner

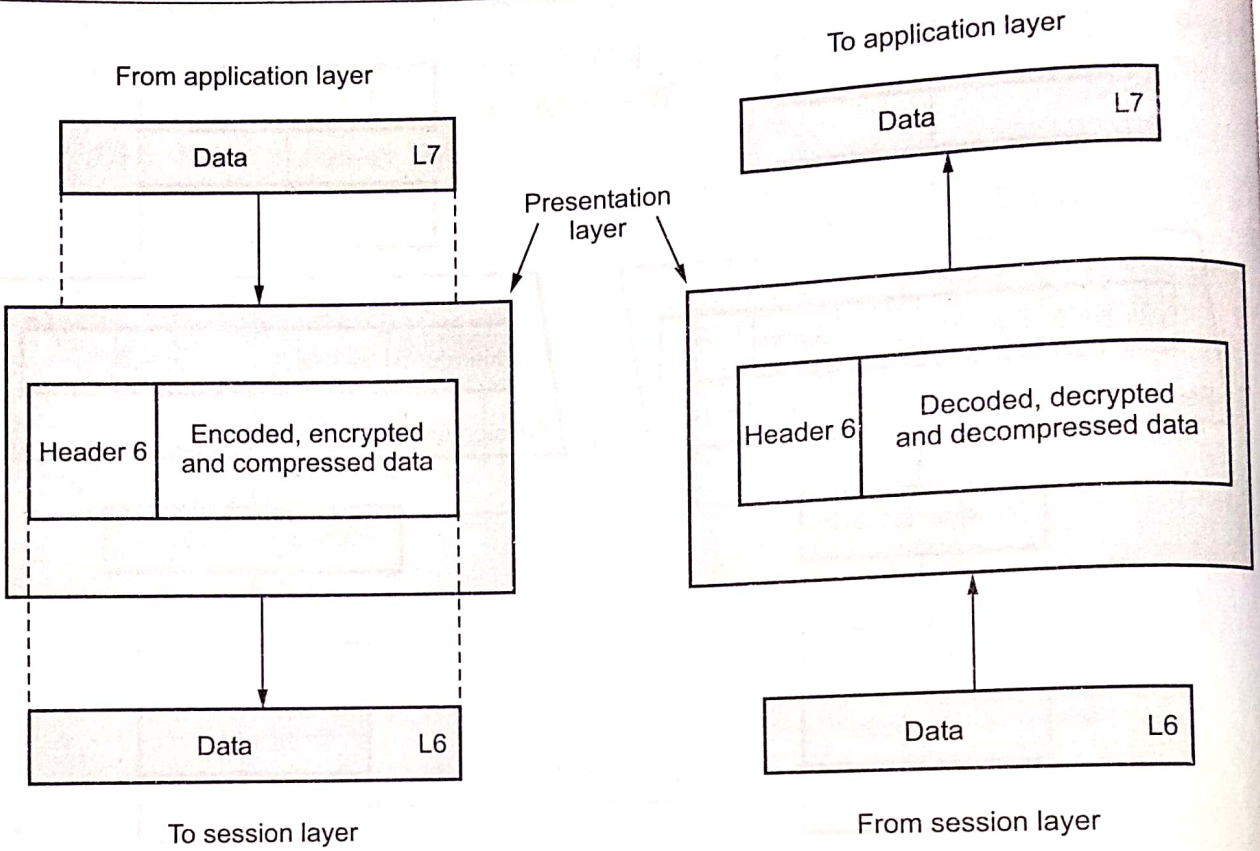


Fig. 1.8.7 Presentation layer

7. Application Layer

- Application layer is responsible for accessing the network by user. It provides user interfaces and other supporting services such as e-mail, remote file access, file transfer, sharing database, message handling (X.400), directory services (X.500).

Functions of Application Layer

1. **Network virtual terminal** : It is a software version of physical terminal that allows a user to log onto a remote host.
2. **File Transfer, Access and Management (FTAM)** : FTAM allows user to access files in remote hosts, to retrieve files and to manage files in remote computer.
3. **Mail services** : E-mail forwarding, storage are the services under this category.
4. **Directory services** : Directory services include access for global information and distributed database.

University Questions

1. *Draw diagram of OSI reference model ? What are the benefits of layering approach in OSI model ?*

GTU : Winter-14, Marks 7

Scanned with CamScanner

2. *Which of the OSI layers handles each of the following :*
   a) Determine which route through the subnet to use.
   b) Dividing the transmitted bit stream into frames.
   c) Encryption and compression of the information.
   d) Flow control between source and destination node. **GTU : Dec.-10, Marks 4**
3. *What is OSI model ? Draw diagram and explain physical, data link and network layer with its functions.* **GTU : Dec.-11, Marks 5**
4. *Explain for OSI reference model with diagram.* **GTU : Winter-13, Marks 8**
5. *Explain functions of different layers of OSI model.* **GTU : Summer-14, Marks 7**

1.9 TCP/IP Protocol

- The internet architecture, which is also sometimes called the TCP/IP architecture after its two main protocols.
- TCP/IP stands for Transmission Control Protocol / Internet Protocol.
- The TCP/IP reference model is a set of protocols that allow communication across multiple diverse networks.
- TCP/IP is normally considered to be a four layer system. Layers of TCP/IP are Application layer, Transport layer, Internet layer, Host to network layer.
- Host to network layer is also called physical and data link layer.
- The application layer in TCP/IP can be equated with the combination of session, presentation, application layer of the OSI reference model.
- Fig. 1.5.1 shows TCP/IP reference model.
- TCP/IP defines two protocol at transport layer : TCP and UDP.
- **User Datagram Protocol (UDP)** is connectionless protocol.
- UDP is used for application that requires quick but necessarily reliable delivery.
- Internet layer also called **network layer.** Internet layer handles communication from one machine to the other. Routing of packet takes place in internet layer.

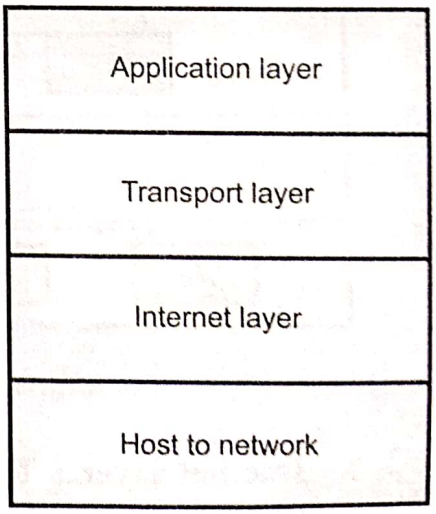


Fig. 1.9.1 TCP/IP reference model

- TCP/IP does not define any specific protocol in host to network layer. This layer is responsible for accpeting and transmitting IP datagrams. This layer normally includes the device driver in the operating system.

Scanned with CamScanner

- Detailed function of each layer is given below.
  1. **Application layer :** Application layer includes all process and services that use the transport layer to deliver data. The most widely known application protocols are : TELNET, File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) and Simple Network Management Protocol (SNMP). TELNET is the Network Terminal Protocol, which provides remote login over the network. FTP is used for interactive file transfer. SMTP delivers the electronic mail.
  2. **Transport layer :** Application programs send data to the transport layer protocols TCP and UDP. An application is designed to choose either TCP or UDP based on the services it needs.
- The transport layer provides peer entities on the source and destination hosts to carry on a conversation. Both ends protocol is defined in this layer. TCP is reliable connection oriented protocol that allows a byte stream originating on one computer to be delivered without error or any other computer in the internet. It converts the incoming byte stream into discrete message and passes each one onto the internet layer. At the destination side, the receiving TCP reassembles the received data or messages into the output format. TCP also handles flow control. It synchronizes between fast sender and slow receiver. UDP is a connectionless protocol. Sometimes this type of protocol is used for prompt delivery. The relation of the protocols is shown in the Fig. 1.9.2.

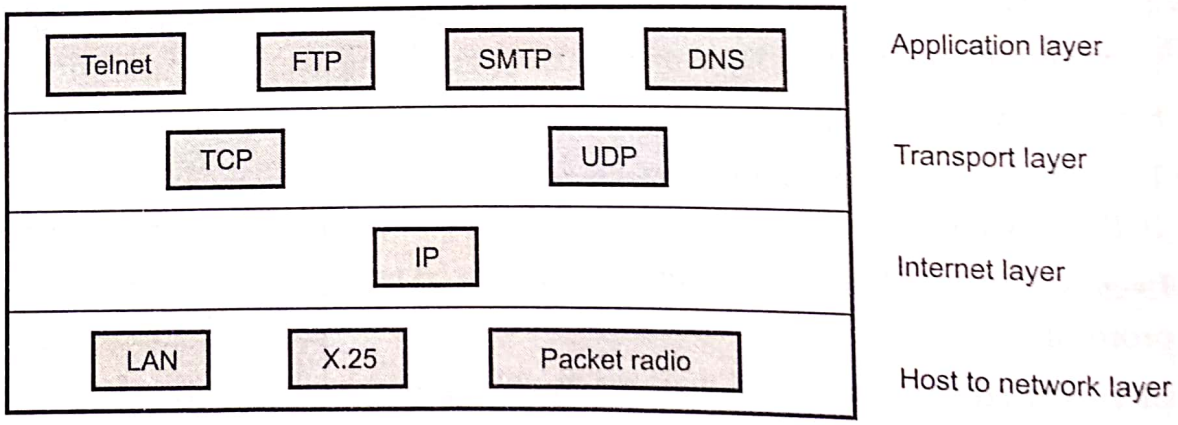


Fig. 1.9.2 Relation of protocol in TCP/IP model

  3. **Internet layer :** The Internet network level protocol (IP, ARP, ICMP) handle machine to machine communications.
- These protocols provide for transmission and reception of transport requests and handle network level control. The TCP/IP internet layer moves data from one host to another; even if the hosts are on different networks. The primary protocol used to move data is the Internet Protocol (IP), which provides the following services :

Scanned with CamScanner

a. **Addressing :** Determining the route to deliver data to the destination host.

b. **Fragmentation :** Breaking the messages into pieces if an intervening network cannot handle a large message.

- It provides a connectionless method of delivering data from one host to another. It does not guarantee delivery and does not provide sequencing of datagrams. It attaches a header to datagram that includes source address and the destination address, both of which are unique internet addresses.

4. **Host to network :** This layer is also called network interface layer. This layer is same as **physical and data link layer** of **OSI model. Host to network layer cannot define any protocol.** It is responsible for accpeting and transmitting IP datagrams. This layer may consist of a device driver in the operating system and the corresponding network interface card in the machine.

1.9.1 Comparison of the OSI and TCP/IP

Fig. 1.9.3 shows the OSI and TCP/IP model.

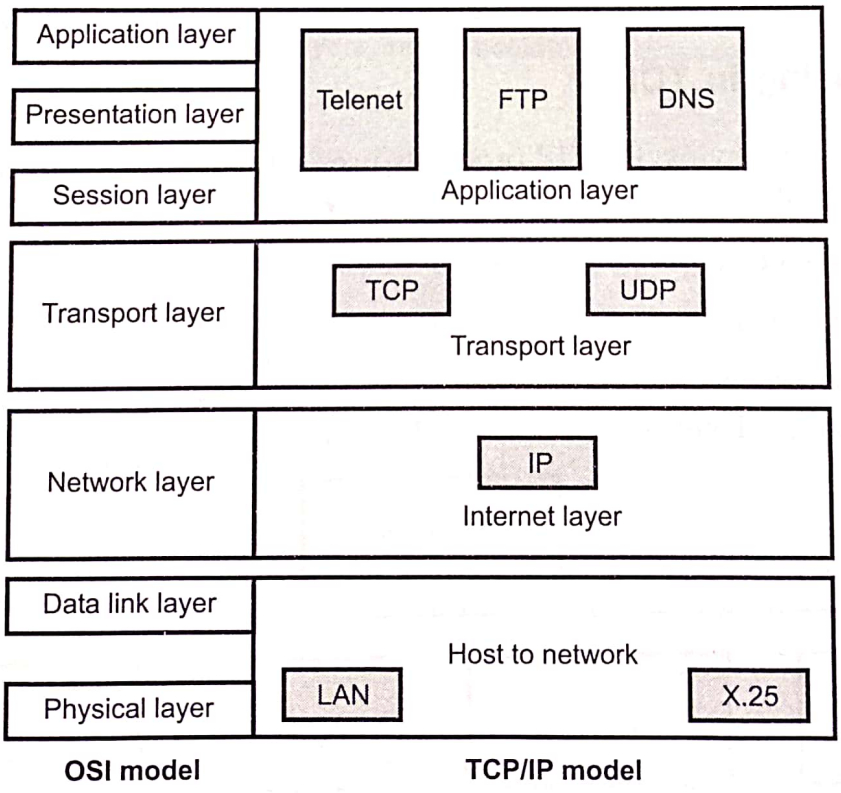


Fig. 1.9.3 OSI and TCP/IP model

Scanned with CamScanner

Sr. No.	OSI	TCP/IP
1.	7 layers	4 layers
2.	Model was first defined before implementation takes place.	Model defined after protocol were implemented.
3.	OSI model based on three concept i.e. service, interface and protocol.	TCP/IP model did not originally clearly distinguish between service, interface and protocol.
4.	OSI model gives guarantee of reliable delivery of packet.	Transport layer does not always guarantee the reliable delivery of packet.
5.	OSI does not support internet working.	TCP/IP support.
6.	Strict layering.	Lossely layered.
7.	Support connectionless and connection-oriented communication in the network layer.	Support only connection-oriented communication in the transport layer.

1.10 Addressing in TCP/IP

- An Internet employing TCP/IP protocols uses four levels of addresses :
  1. Physical (Link) addresses
  2. Logical (IP) addresses
  3. Port addresses
  4. Specific addresses
- Fig. 1.10.1 shows level of addresses in TCP/IP.

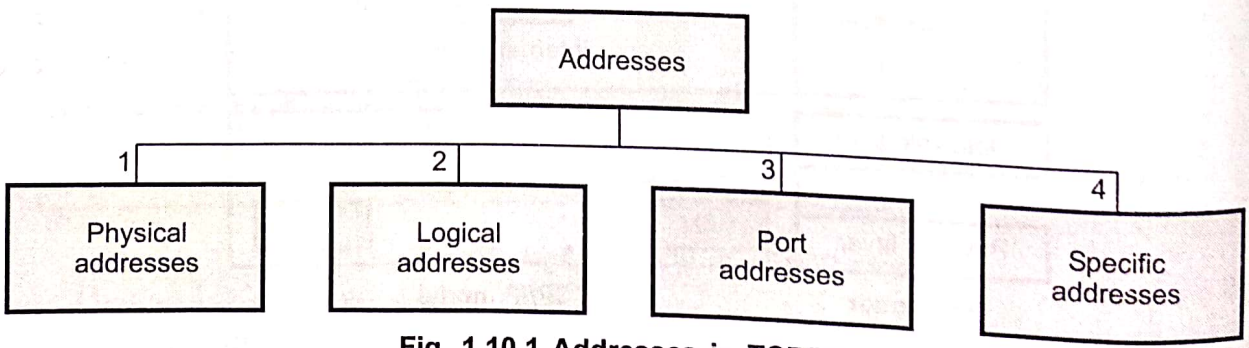


Fig. 1.10.1 Addresses in TCP/IP

- Each address type is related to a specific layer in TCP/IP architecture. Fig. 1.10.2 shows the relationship of layers and addresses in TCP/IP.

Scanned with CamScanner

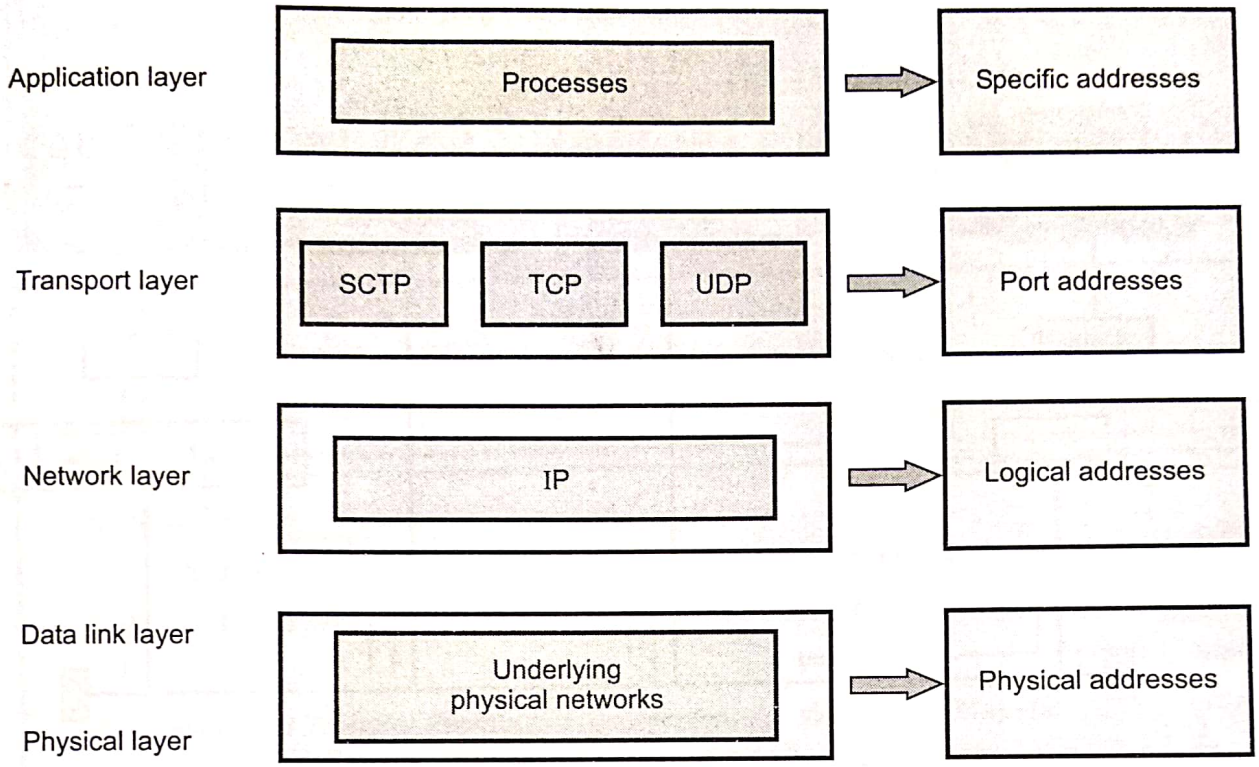


Fig. 1.10.2 TCP/IP layers and associated addresses

1.10.1 Physical Addresses

- The physical address is the lowest level address and is also referred as link address. The physical address of a node is defined by its LAN or WAN. The physical address is included in the frame by the data link layer.
- The size and format of physical addresses vary depending on the network. It has authority over the network. At data link layer, the frame contains physical (link) addresses in the header. The data link layer at sender receives data from upper layer, encapsulates the data in a frame, adds an header and trailer. Only the station having matched address with destination address accepts the frames. The frame is checked, the header and trailer are dropped and data is decapsulated and delivered to upper layer. (See Fig. 1.10.3 on next page)

1.10.2 Logical Addresses

- Logical addresses are independent of underlying physical networks. Since different networks can have different address formats hence a universal address system is required which can identify each host uniquely irrespective of underlying physical networks. Logical addresses are necessary for universal communications. It is a 32-bit address which uniquely defines host connected to Internet.
- The physical addresses changes from hop to hop, but the logical address usually remains the same.

Scanned with CamScanner

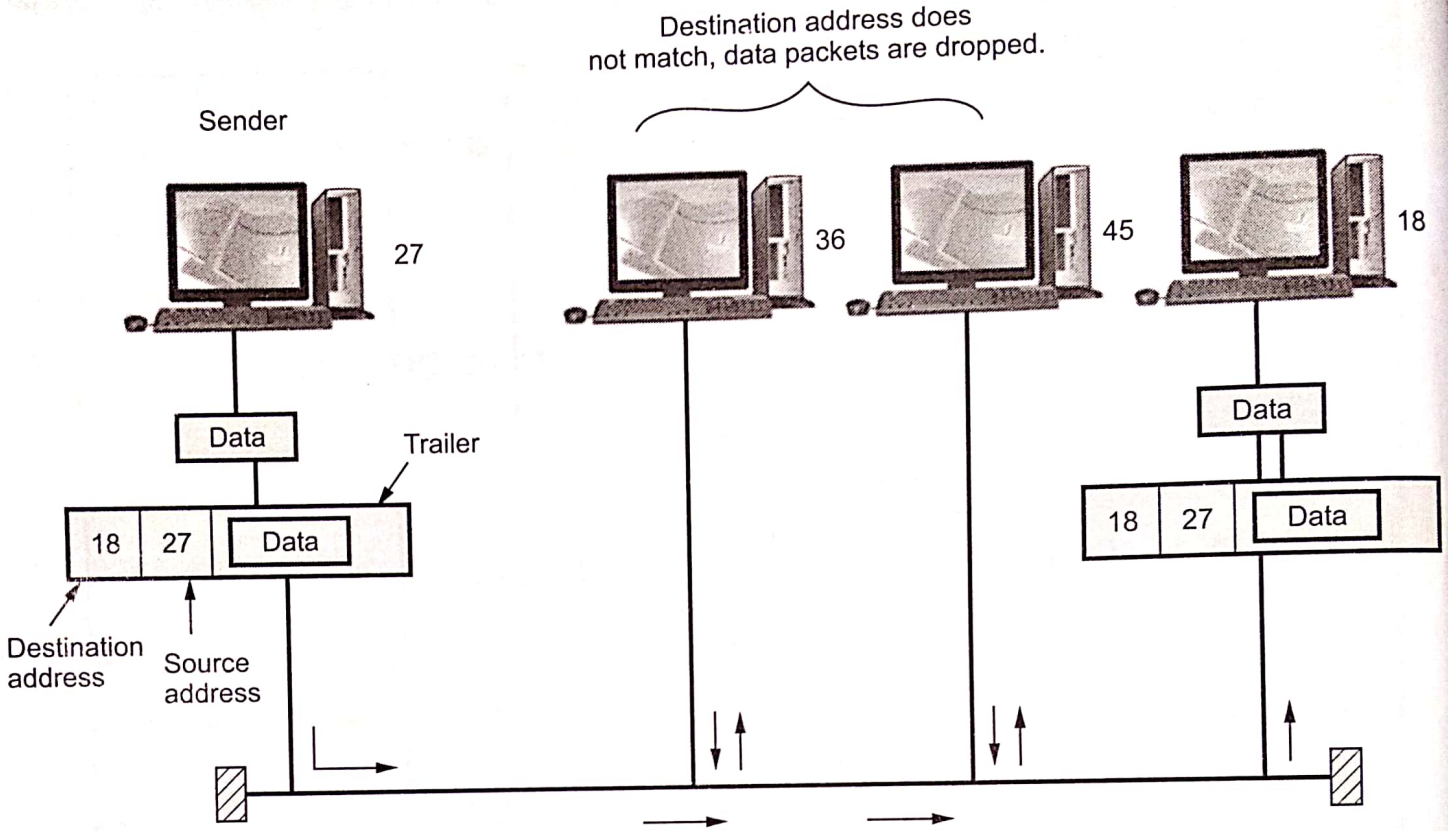


Fig. 1.10.3 Physical addresses

1.10.3 Port Addresses

- The IP address and physical address are necessary for data to travel from source to destination. But a communication process involves TELNET and FTP which requires addresses. In TCP/IP architecture, the label assigned to a process is called port address. In TCP/IP the port address is of 16-bit.

1.10.4 Specific Addresses

- Specific addresses are designed by users for some applications. For example, evilaas@in.com and the Universal Resource Locator (URL), www.vtubooks.com. The first example defines the recipient of e-mail and second example is used to find a document on the world wide web.
- The specific addresses gets changed to corresponding port and logical addresses by the station or host who sends it.

❑❑❑

Scanned with CamScanner

2 Application Layer

Syllabus

Principles of computer applications, Web and HTTP, E-mail, DNS, Socket programming with TCP and UDP.

Contents

2.1 Principles of Computer Applications. . . . . **Summer-14,** · · · · · · · · · · · · Marks 7

2.2 Electronic Mail **Winter-14, Dec.-10, 11, May-12,**
. . . . . **Winter-12,** · · · · · · · · · · · · · · Marks 7

2.3 Hypertext Transfer Protocol . . . . . **Winter-14, May-12,** · · · · · · · · Marks 7

2.4 Domain Name System . . . . . **Winter-12, 13, 14, Dec.-10, 11,** · · · ·
. . . . . **June-11,Summer-14,** · · · · · Marks 7

2.5 World Wide Web . . . . . **Dec.-11, Winter-12, 13** · · · · · Marks 7

2.6 Socket Programming with TCP and UDP

Scanned with CamScanner

2.1 Principles of Computer Applications

GTU : Summer-14

- Application layer provides the interface between the applications, use to communicate and underlying network.
- A process is a program in execution. When communicating processes are running on the same system, they communicate with each other using interprocess communication.
- Processes on two different end systems communicate with each other by message passing between computer networks.
- Fig. 2.1.1 shows communication for network application at application layer.

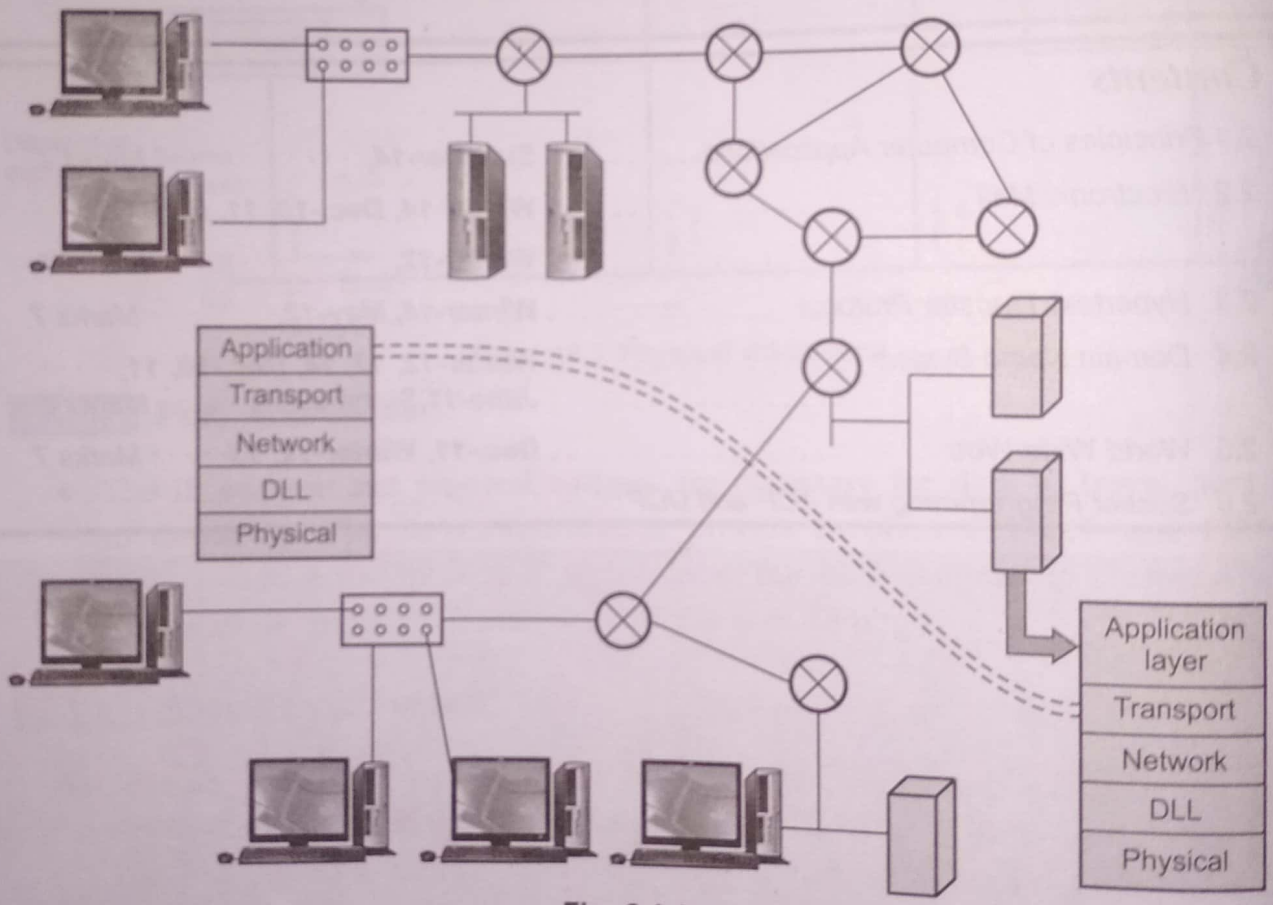


Fig. 2.1.1

- Networking applications have application layer protocols. They decide the rules for communication along with message format.

2.1.1 Application Layer Protocols

- World Wide Web, electronic mail system and domain name system are the traditional application of the application layer network.

Scanned with CamScanner

- Applications need their own protocols. These applications are part network protocol and part traditional application program.
- Here we study some of the most popular network applications available today. Two of the most popular applications are World Wide Web and Email system.
- Both of these applications use the request/reply method. The users send requests to servers, which then respond accordingly.
- These two application uses various protocol while exchanging the information. So it is important to distinguish between application protocols from application programs.
- Hyper Text Transport Protocol (HTTP) is an application protocol. HTTP is used to retrieve Web pages from remote servers.
- A web client uses application programs like Internet Explorer, Chrome, Firefox and Mozilla. All of them use the HTTP protocol for communication with web server.
- Widely used standardized application protocols are SMTP and HTTP
  1. Simple Mail Transfer Protocol is used to exchange electronic mail.
  2. HTTP is used to communicate between Web browsers and Web servers.

Client and Server Side Application

- Web browser is client side application of HTTP. Web server is server side application of HTTP.
- In electronic mail system, sending mail server implements the client side of SMTP and the receiving mail server implements the server side of SMTP.

Processes Communicating Across Network

- Two processes communicate with each other by sending and receiving messages. Socket is used for process communication. Socket is an interface between the application layer and transport layer within a machine.
- Socket is also referred as the application programmer's interface (API) between the application and the network. Multiple sockets might exist in each host. A port number identifies each such socket in each host.
- **Client process :** Process that initiates communication.
- **Server process :** Process that waits to be contacted.
- Application layer protocols used by both the source and destination device during a communication session. The protocols implemented on the source and destination host must match.

Scanned with CamScanner

- The client process begins the exchange by requesting data from the server. Serv responds by sending one or more streams of data to the client.

Addressing Processes

- A host (sender) uses the address of the destination host to specify where t message should sent. When data is received at the host, the port number examined to determine which application or process is the correct destination for the data.
- Each device on a network must be uniquely defined. In Internet applications, the destination host is identified by its IP address.
- Fig. 2.1.2 shows application processes, socket and transport protocol.

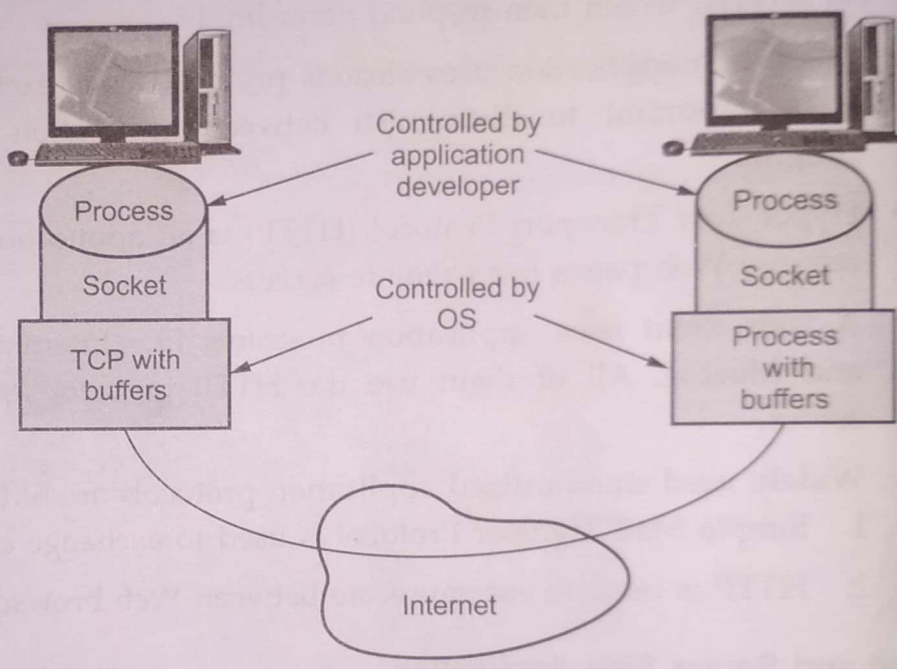


Fig. 2.1.2 Application processes, socket and transport protoc

2.1.2 Types of Services Required for Application

- Application service requirements are as follows :
  1. Data Loss
  2. Bandwidth
  3. Timing
- Following application requires reliable data transfers :
  a. File transfer b. E-mail
  c. Remote host access d. Instant messaging
  e. Financial application f. Web document transfer
- Loss of file data creates problem. Loss of data strongly depends upon t application and coding scheme used.

Bandwidth

- Some applications like multimedia require a minimum amount of bandwidth to b effective.

Scanned with CamScanner

- Many current multimedia applications are bandwidth sensitive.
- The application which requires little or less bandwidth, they are called *elastic applications*. Example of elastic applications are electronic mail, file transfer and remote access and web transfers.

Timing

- Some application requires low delay.
- Internet telephony, teleconferencing, multiplayer games requires tight timing constraints on data delivery.

Review Question

1. *Explain any two application layer protocol.* **GTU : Summer-14, Marks 7**

2.2 Electronic Mail

GTU : Winter-14, Dec.-10, 11, May-12, Winter-12

- E-mail is an asynchronous communication medium. Electronic mail is used for sending a single message that includes text, voice, video or graphics to one or more recipients.
- Electronic mail is fast, easy to distribute and inexpensive.
- Simple Mail Transfer Protocol (SMTP) is the standard mechanisms for electronic mail in the internet. SMTP is the TCP/IP mail delivery protocol.
- E-mail is not a real-time service in that fairly large delays can be tolerated.
- It is also not connection oriented in that a network connection does not need to be setup expressly for each individual message.
- Fig. 2.2.1 shows the high-level view of the internet e-mail system.
- Mail server handles incoming and outgoing mails.
- The Post Office Protocol (POP) servers store incoming mail while SMTP servers relay outgoing mails.
- The Internet Service Provider (ISP) probably runs both an SMTP server and POP server for its customers.

Following are the ways to access the e-mail.

1. Web based e-mail service.
2. E-mail through a LAN.
3. Unix shell account.
4. Using mail client.

Scanned with CamScanner

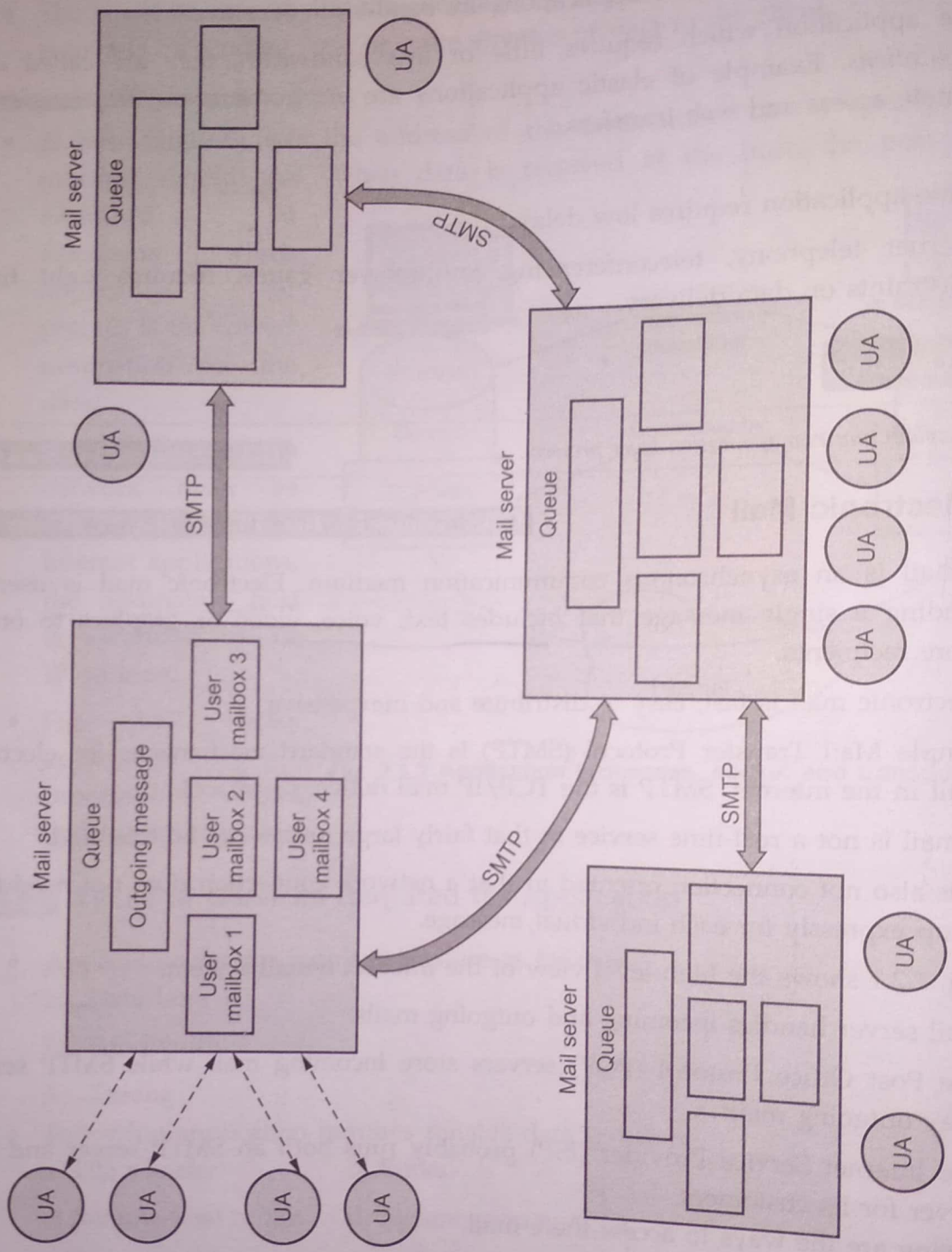


Fig. 2.2.1 View of e-mail system

Components

Three major components are

1. User agents.
2. Mail servers.
3. SMTP.

Scanned with CamScanner

Fig. 2.2.2 shows the components of an e-mail system.

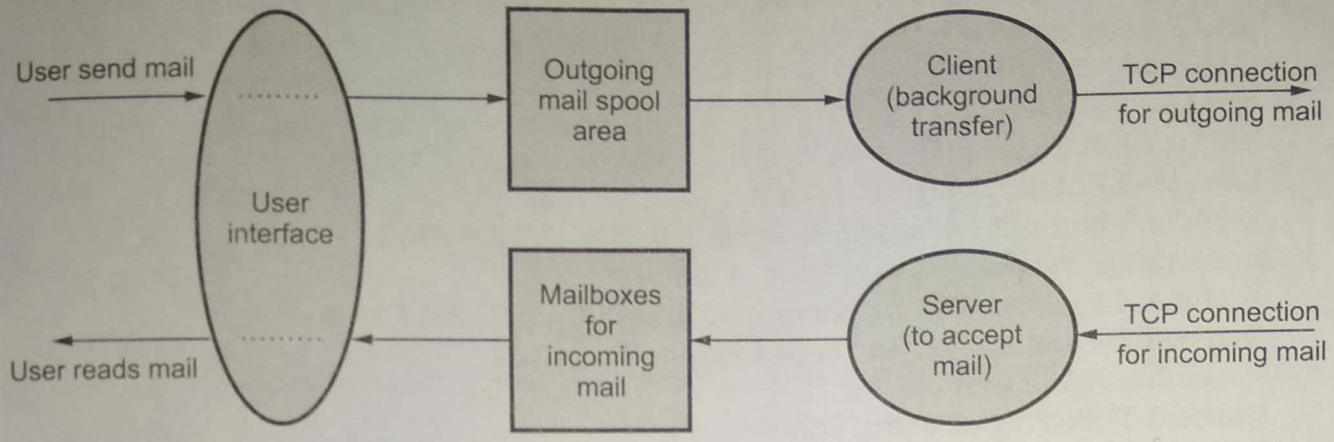


Fig. 2.2.2 Component of e-mail system

2.2.1 E-mail Addressing

- To send e-mail to some one, the Internet e-mail address must be known to sender. E-mail addresses look like this : vilas@hotmail.com.
- The e-mail address has two main parts, joined by @. In this example, vilas is the username. Username can contain numbers, underscores, periods and some other special characters. Commas, spaces and parentheses are not allowed.
- Hotmail.com is the host or domain name. E-mail address is case insensitive. **Vilas@hotmail.com** works just the same as **vilas@hotmail.com**. E-mail addresses do not have punctuation marks around them.

2.2.2 Message Headers

The message headers include the addresses of the receiver and the sender. Each header consists of the type of header, a colon, and the content of the header. Following is the sample of the complete header for a message.

Table shows the list of standard header.

```
Received:from del2.vsnl.net.in(del2.vsnl.net.in [202.54.15.30])
by giaspn01.vsnl.net.in (8.9.0/8.9.0) with ESMTP id MAA22885
for<siitpune@giaspn01.vsnl.net.in>;wed,19 jul 2000
      12:42:33+0530 (IST)
Received: from oemcomputer ([202.54.109.165])
by del2.vsnl.net.in (8.9.2/8.9.2)with SMTP id MAA12595
for<siitpune@giaspn01.vsnl.net.in>;wed, 19 Jul 2000
12:47:52-0500 (GMT)
Reply-To: <kanohar@del2.vsnl.net.in>
From:"SachinMahadik" <kanohar@del2.vsnl.net.in>
To: <siitpune@giaspn01.vsnl.net.in>
Subject: admission
Date: Wed, 19 Jul 2000 12:43:31 +530
```


Scanned with CamScanner

```
Message-ID:<LPBBKDKDNBJBIDPNDOLHOECOCBAA.kanohar@del2.vsnl.net.in>
MIME-version: 1.0
Content-Type: text/plain;
charset="iso-8859-1"
X-Priority: 3 (Normal)
X-MSMail-Priority: Normal
X-Mailer:Microsoft outlook IMO, Build 9.0.2416 (9.0.2910.0)
Importance: Normal
X-MimeOLE : Produced by Microsoft MimeOLE V5.00.2314.1300
Disposition-Notification-To:"BrijeshSinghal"
<kanohar@del2.vsnl.net.in>
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted:from quoted-printable to 8bit by
giaspn01.vsnl.net.in id MAA22885
X-UIDL: 69c2d7f9f63fef91eaf7c61f05d2b550
X-Mozilla-Status: 8003
```

2.2.3 Formatted E-mail

- E-mail that supports formatting such as boldface and underlining is a recent development. In the past, e-mail consists only of text characters. If both side supports the formatted e-mail, then both sides use send/receive formatted e-mail. The formatted e-mail comes in the following type.
  a) HTML
  b) Rich text
  c) Multipurpose Internet Mail Extension (MIME)
  d) MS word format
- HTML tags are just like web pages. It can include text formatting, numbering, bullets, horizontal lines, backgrounds, hyperlinks and HTML styles. It uses MIME protocol for sending. Rich text can be read by most word processing applications. MIME formatting are created just for e-mail.
- MIME formatting can include text formatting, pictures, video, sound, and other information. MS word format uses microsoft word and all of its features as your e-mail editor.
- To allow transmission of non-ASCII data through e-mail, the MIME used. MIME does not change SMTP or replace it. MIME allows arbitrary data that is to be encoded in ASCII and then transmitted in a standard e-mail message.
- Fig. 2.2.3 shows the MIME message that contains a photograph in standard GIF representation. The GIF image has been converted to a 7-bit ASCII representation using the base 64 encoding.

Scanned with CamScanner

```
From : Ravindra@hotmail.com
To : Avinash@hotmail.com
MIME-version : 1.0
Content-Type : imag/gif
Content-Transfer-Encoding : base64
.....data for the image.....
- - - - - - - - - - - - - - - - - - - - - -
```

Fig. 2.2.3 Example MIME message

- The MIME-version declares that the message was composed using version 1.0 of the MIME protocol. The content-type declaration specifies that the data is GIF image and the content-transfer-encoding header declares that base-64, encoding was used to convert the image to ASCII. To view the image, a receiver mail system must first convert from base 64 encoding back to binary.
- A content-type declaration must contain two identifiers, a content-type and a subtype, separated by a slash. In the example, image is the content type and gif is the subtype. The standard defines seven basic content types.

Sr. No.	Content type	Use
1.	Text	Textual (document)
2.	Image	Photograph
3.	Audio	A sound recording
4.	Video	A video recording including motion
5.	Application	Raw data for program
6.	Multipart	Multiple messages
7.	Message	An entire e-mail message

2.2.4 Functions of E-mail

- E-mail system support five basic functions. They are as follows -
  1. Composition
  2. Transfer
  3. Reporting
  4. Displaying
  5. Disposition

1. Composition : It is a process of creating messages and answers. Any text editor can be used for the body of the message. When answering a message, the e-mail system can extract the originator's address from the incoming e-mail.

Scanned with CamScanner

2. Transfer : It is moving messages from the originator to the receiver.
3. Reporting : It inform the originator what happened to the message. Whether, email is delivered or not delivered.
4. Displaying : Display is required for reading the email.
5. Disposition is the last step and related what the receiver does with the message after receiving it. It may be read and save or delete or forward the message.

2.2.5 User Agent and Message Transfer Agent

E-mail system consists of two subsystems.

1. User agent
2. Message transfer agent.

1. User Agent (UA)

- User agent is an interface between user and network application.
- It allow user to read and send e-mail. The user agents are local program that provide a command based, menu based or graphical method for interacting with the e-mail system.
- To send an e-mail message, a user must provide the data and the destination address. The destination address should be in proper format and the user agent can deal with destination address. Details of e-mail address, we already studied in email addressing section.
- Most e-mail system support mailing lists, so that a user can send the same message to a list of people with a single command.
- For reading e-mail, the user agent will look at the user's mail box for incoming e-mail before displaying anything on the screen. It display total number of new mail.

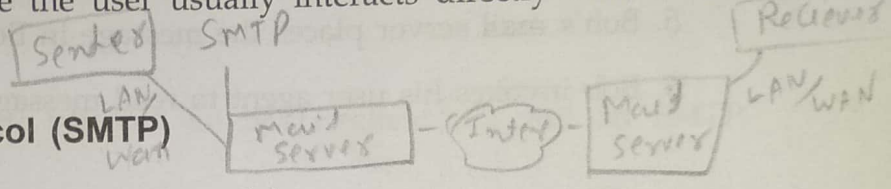
2. Message transfer agent

- Message Transfer Agent (MTA) move the messages from the source to the destination. MTA are system program that run in the background and move e-mail through the system.
- After writing the mail, user click of send icon. MTA activates at this time, MTA checks the destination address and transfer the mail to proper destination on the network.
- MTA use different types of protocol for moving the message from source to destination.

Scanned with CamScanner

1. It must handle temporary failures, if a destination machine is temporarily unavailable, it must spool the message on the local machine for later delivery.
2. MTA must distinguish between local and remote destinations.
3. It may have to deliver copies of a message to several machines.
4. It may allow mixing text, voice and video in a message as well as appending documents and files to a message.

- MTA works in background, while the user usually interacts directly with a user agent.



2.2.6 Simple Mail Transfer Protocol (SMTP)

- SMTP is application layer protocol of TCP/IP model.
- SMTP transfers message from sender's mail servers to the recipients mail servers.
- SMTP interacts with the local mail system and not the user.
- SMTP uses a TCP socket on port 25 to transfer e-mail reliably from client to server.
- E-mail is temporarily stored on the local and eventually transfered directly to receiving server.
- Client / Server interaction follows and commond/reponse paradigm.
  a] Commands are plain ASCII text.
  b] Responses are a status code and an optional phase.
  c] Command and response lines terminated with CRLF.
- Mail client application interacts with a local SMTP server to initiate the delivery of an e-mail message.
- There is an input queue and an output queue at the interface between the local mail system and the client and the server parts of the SMTP.
- The client is concerned with initiating the transfer of mail to another system while server is concerned with receiving mail. Before the e-mail message can be transferred, the application process must be set up a TCP connection to the local SMTP server. The local mail system retains a mailbox for each user into which the user can deposit or retrieve mail. Mail handling system must use a unique addressing system.
- Adderssing system used by SMTP consists of two parts : A local part and a global part. The local part is the user name and is unique only within that local mail system. Global part of the address is the domain name. Domain name is identity of the host, must be unique within the total internet.

Scanned with CamScanner

- SMTP uses different types of component. They are MIME and POP.

Scenario : Alice sends message to Bob

1. Alice uses User Agent (UA) to compose message and send to bob@technical.org.
2. Alice's UA sends message to her mail server, message placed in message queue.
3. Client side of SMTP opens TCP connection with Bob's mail server.
4. SMTP client sends Alice's message over the TCP connection.
5. Bob's mail server places the message in Bob's mailbox.
6. Bob invokes his user agent to read message.

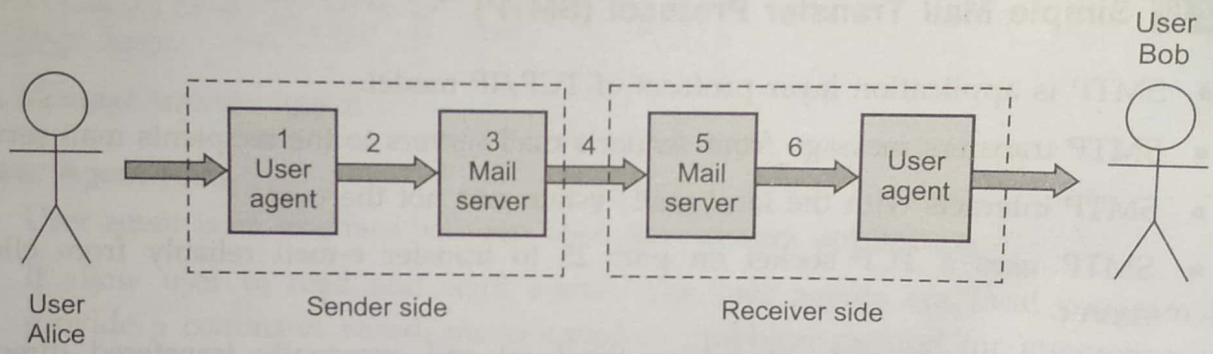


Fig. 2.2.4 Message Scenario

- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.

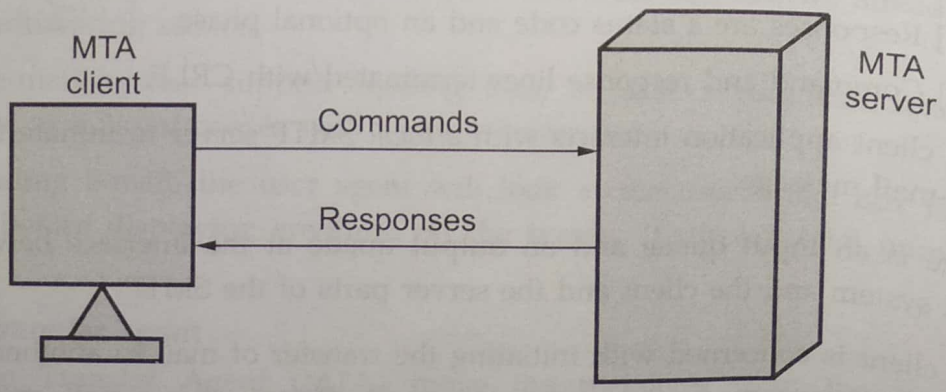


Fig. 2.2.5 Command / Response

- Each command or reply is terminated by a two character end of line token.
- Commands are sent from the client to the server. SMTP defines 14 commands. SMTP commands consist of human readable ASCII strings.

SMTP commands are,

i) **HELO :** Initiate a mail transaction, identifying the sender to the recipient.

Used by client by client to identify itself

Scanned with CamScanner

ii) **MAIL FROM :** Tells the remote SMTP that a new mail transaction is beginning.

iii) **RCPT TO :** The sending SMTP sends a RCPT command for each intended receiver.

iv) **DATA :** If accepted, the sender transfers the actual message. End of message is indicated by sending a " • " on a line by itself.

v) **QUIT :** Terminate the connection.

Sample SMTP Interaction

- Following are messages exchanged between an SMTP client (C) and an SMTP server (S).
- The host name of the client is iresh.fr and the host name of the server is sinhgad.edu.

S : 220 sinhgad.edu

C : HELO iresh.fr

S : 250 Hello iresh.fr, pleased to meet you

C : MAIL FROM : <rupali@iresh.fr>

S : 250 rupali@iresh.fr ... sender ok

C : RCPT TO : < rakshita@sinhgad.edu>

S : 250 rakshita@sinhgad.edu ..... Recipient ok

C : DATA

S : 354 Enter Mail, end with " · " on a line by itself

C : Do you like Apple ?

C : What about school ?

C : ·

S : 250 message accepted for delivery

C : QUIT

S : 221 sinhgad.edu closing connection

2.2.7 Multipurpose Internet Mail Extensions

- MIME is a supplementary protocol that allows non-ASCII data to be sent through SMTP.
- MIME defined by IETF to allow transmission of non-ASCII data via e-mail.
- It allows arbitary data to be encoded in ASCII for normal transmission.

Scanned with CamScanner

- All media types that are sent or received over the world wide web (www) are encoded using different MIME types.
- Messages sent using MIME encoding include information that describes the type of data and the encoding that was used.
- RFC822 specifies the exact format for mail header lines as well as their semantic interpretations.
- Fig. 2.2.6 shows the working of MIME.

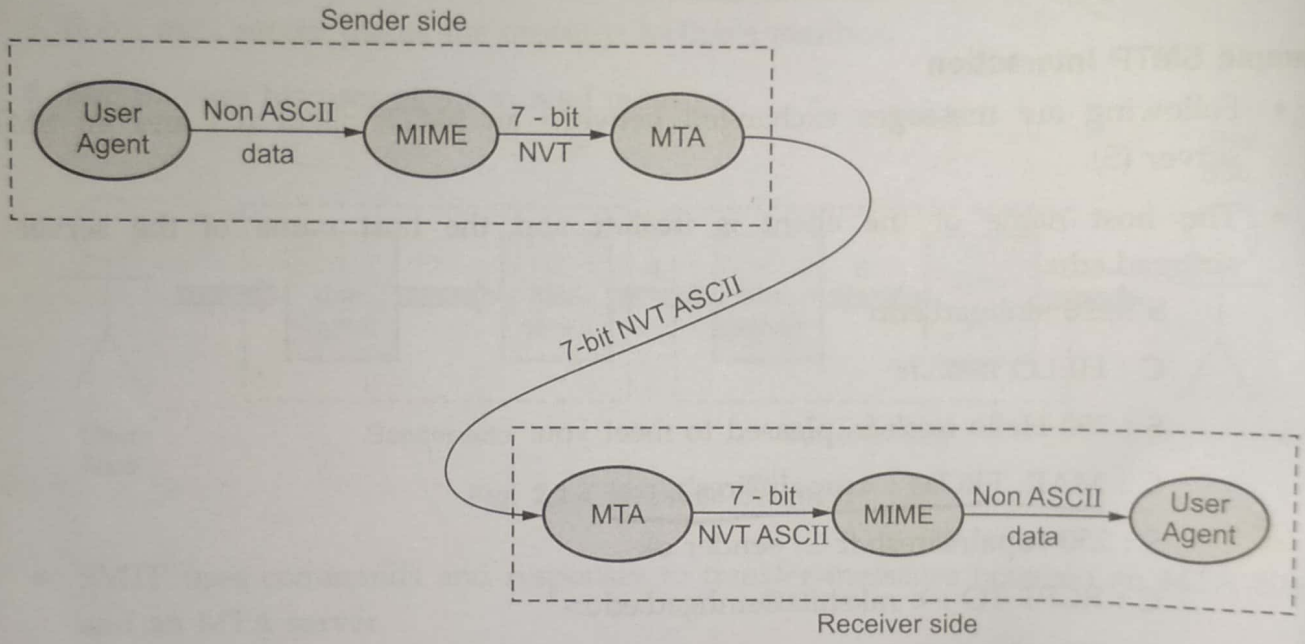


Fig. 2.2.6 MIME

- MIME define five headers.
  1. MIME - Version
  2. Content - Type
  3. Content - Transfer - Encoding
  4. Content - Id
  5. Content - Description

Mail Message Header

- From : iresh@e-mail.com
- TO : rupali@sinhgad.edu
- MIME - Version : 1.0
- Content - Type : image/gif
- Content - Transfer - Encoding : base64

Scanned with CamScanner

..... data for the image .....

....

....

MIME Types and SubTypes

- Each MIME content - type must contain two identifiers :
  - Content type
  - Content subtype
- There are seven standardized content-types that can appear in a MIME content - type declaration.

Type	Subtype	Description
Text	Plain	Unformatted text.
Multipart	Mixed	Body contains ordered parts of different data types.
	Parallel	Same as above, but no order.
	Digest	Similar to mixed, but the default is message.
	Alternative	Parts are different versions of the same message.
Video	MPEG	Video is in MPEG format.
Audio	Basic	Single channel encoding of voice at 8kHz. (Sound file)
Image	JPEG	Image is in JPEG format.
	GIF	Image is in GIF.
Message	Partial and external body	An entire e-mail message or an external reference to a message.
Application	Postscript	Adobe postscript.
	Octet stream	General binary data.

Content - Transfer Encoding

- This header defines the method to encode the messages into 0 and 1 for transport.

Content-Transfer-Encoding : < Type >

Scanned with CamScanner

The five types of encoding is listed below.

Type	Description
7-bit	ASCII characters and short lines.
8-bit	Non-ASCII characters and short lines.
Binary	Non-ASCII characters with unlimited length lines.
Base 64	6-bit blocks of data are encoded into 8-bit ASCII characters.
Quoted printable	Non-ASCII characters are encoded as an equal sign followed by an ASCII code.

2.2.8 Post Office Protocol (POP)

- Post Office Protocol 3 (POP3) is used to transfer e-mail messages from a mail server to mail client software.
- Fig. 2.2.7 shows working of POP3.
- POP3 begins when the user agent opens a TCP connection to the mail server on port 110.
- After TCP connection established, POP3 progresses three phases :
  i) Authorization
  ii) Transaction
  iii) Update
- In authorization phase, user agent sends a user name and a password to authenticate the user downloading the mail.
- In transaction phase, the user agent retrieves messages. In this phase, user agent can also mark messages for deletion, remove deletion marks.
- In update phase, it occurs after the client has issued the quit command, ending the POP3 session.
- POP3 has two modes : Delete mode and the keep mode.
- In the delete mode, mail is deleted from the mailbox after each retrieval.
- In the keep mode, the mail remains in the mailbox after retrieval.

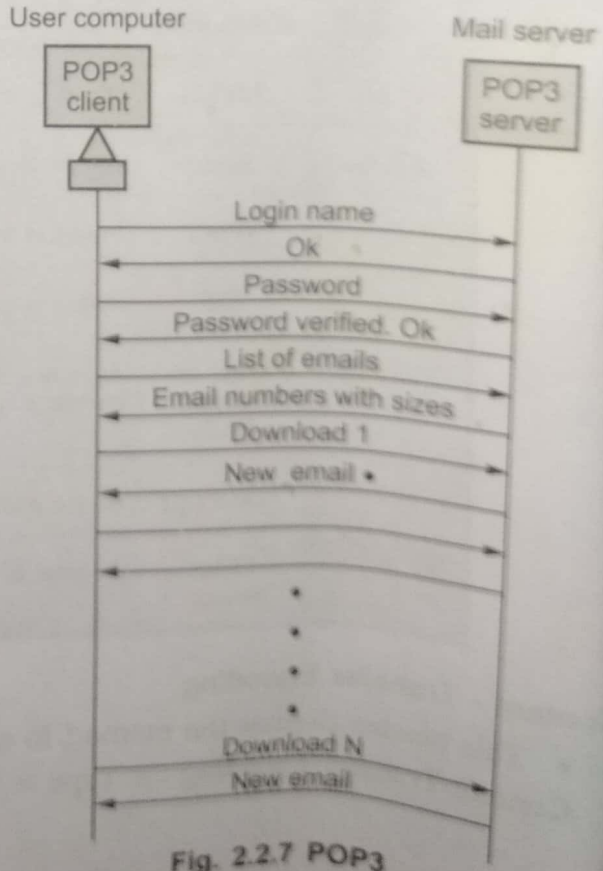


Fig. 2.2.7 POP3

Scanned with CamScanner

Limitations of POP3

1. POP3 does not allow the user to organize mail on the server, the user cannot have different folders on the server.
2. POP3 does not allow the user to partially check the contents of the e-mail before downloading.

2.2.9 IMAP

- IMAP is the Internet Mail Access Protocol. IMAP4 is more powerful and more complex. IMAP is similar to SMTP.
- IMAP allows users to store their email on remote server.
- It was designed to help the user who uses multiple computers.

Status line	HTTP / 1.1 300 ok
General headers	Date : Wed , 8 Oct 2014 13:00:13 GMT
	Connection : close
Entity headers	Server : Apache / 1.3.27
	Accept-range : bytes
	Content-type : text / html
	Content-length : 200
	Last-modified : 2 Oct 2014 13:00:13 GMT
Blank line	
Message body	<html> <head> <title> Welcome to the India <title> <head> <body>

- IMAP does not copy e-mail to the user's personal machine because the user may have several.
- An IMAP client connects to a server by using TCP.
- IMAP supports the following modes for accessing e-mail messages :
  i) Offline mode ii) Online mode iii) Disconnected mode

Offline mode : A client periodically connects to the server to download e-mail messages. After downloading, messages are deleted from the server. POP3 support this mode.

Online mode : Client process e-mail messages on the server. The e-mail messages are stored on the server itself but are processed by an application on the client's end.

Disconnected mode : In this mode, both offline and online modes are supported.

IMAP4 provides the following extra functions :

1. User can check the e-mail header prior to downloading.
2. User can partially download e-mail.
3. A user can create, delete or rename mailboxes on the mail server.
4. A user can create a hierarchy of mailboxes in a folder for e-mail storage.
5. User can search the contents of the e-mail for a specific string of characters.

Scanned with CamScanner

- The IMAP protocol provides commands to allow users to create folders and move messages from one folder to another.

Fig. 2.2.8 shows IMAP state transition diagram.

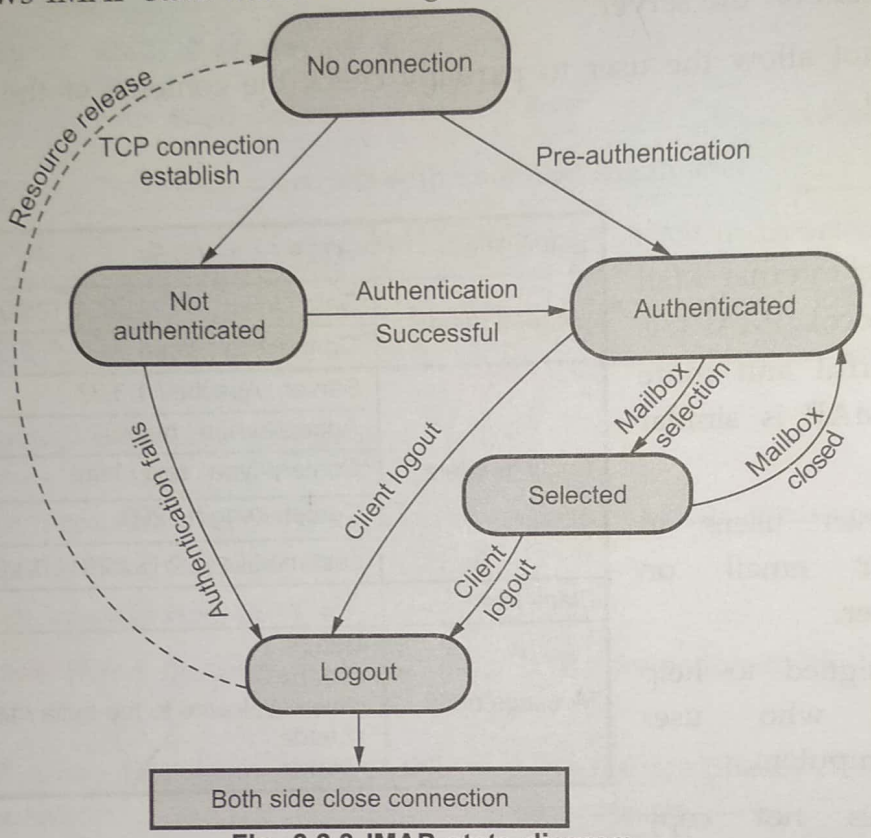


Fig. 2.2.8 IMAP state diagram

1. **Not authenticated :** Client provides authentication information to the server.
2. **Authenticated :** Server verify the information and client is now allowed to perform operations on a mailbox.
3. **Selected :** Client is allowed to access manipulated individual messages within the mailbox.
4. **Logout :** Client send logout command for closing IMAP session.

Review Questions

1. *Write note on following : 1) MIME* **GTU : Winter-14, Marks 3**
2. *Explain the e-mail architecture and services. Write short note on POP3 and MIME.* **GTU : Dec.-10, Marks 7**
3. *What is e-mail ? How it works ? Which protocol it uses ?* **GTU : Dec.-11, Marks 5**
4. *E-mail systems contain which two subsystems ? Write the five basic functions provided by e-mail system.* **GTU : May-12, Marks 7**
5. *Explain the basic functions of the e-mail system.* **GTU : Winter-12, Marks 7**

Scanned with CamScanner

2.3 Hypertext Transfer Protocol

GTU : Winter-14, May-12

- The standard web transfer protocol is Hyper Text Transfer Protocol (HTTP).
- The HTTP protocol consists of two fairly distinct items: The set of requests from browsers to servers and the set of responses going back the other way.
- All the newer versions of HTTP support two kinds of requests: Simple requests and full requests. A simple request is just a single GET line naming the page desired, without the protocol version. The response is just the raw page with no headers, no MIME, and no encoding. To see how this works, try making a Telnet connection to port 80 of www.w3.org and then type.
  GET /hypertext/www/TheProject.html
  but without the HTTP/1.0 this time. The page will be returned with no indication of its content type. This mechanism is needed for backward compatibility. Its use will decline as browsers and servers based on full requests become standard.
- Full requests are indicated by the presence of the protocol version on the GET request line. Requests may consist of multiple lines, followed by a blank line to indicate the end of the request. The first line of a full request contains the command (of which GET is but one of the possibilities), the page desired, and the protocol/version. Subsequent lines contain RFC 822 headers.
- Although HTTP was designed for use in the Web, it has been intentionally made more general than necessary with an eye to future object-oriented applications. For this reason, the first word on the full request line is simply the name of the **method** (command) to be executed on the web page (or general object).
- When accessing general objects, additional object-specific methods may also be available. The names are case sensitive, so, GET is a legal method but get is not.

HTTP Transaction

- HTTP uses the services of TCP. HTTP is a stateless protocol.
- The client initilizes the transaction by sending a request message. The server replies by sending a response.
- Fig. 2.3.1 shows HTTP transaction

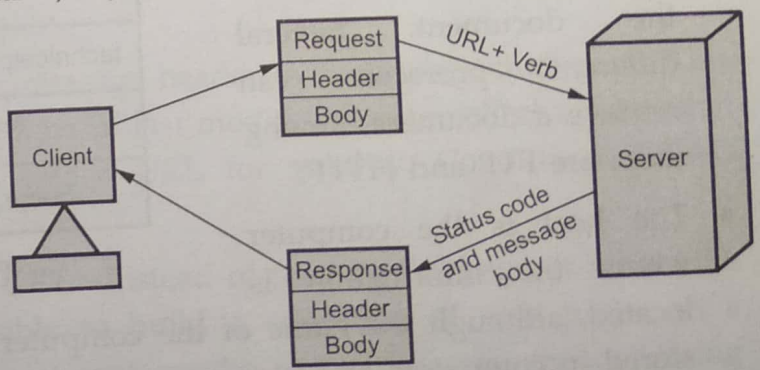


Fig. 2.3.1 HTTP transaction

Message

- HTTP messages are two types
  1. Request 2. Response

Scanned with CamScanner

- Both message type used same format.
- Request **message** consists of a request line, headers and a body. Fig. 2.3.2 shows request message.

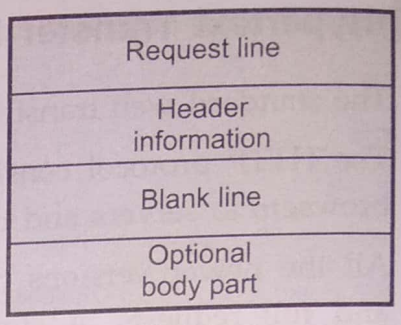


Fig. 2.3.2 Request message

Request line

- Request line defines the
  1. Request type
  2. Resource
  3. HTTP version
- Request type categorizes the request message into several methods for HTTP version 1.1.
- Fig. 2.3.3 shows the request line.
- URL is a standard for specifying any kind of information on the internet. The URL define four things.
  1. Method
  2. Host computer
  3. Port
  4. Path

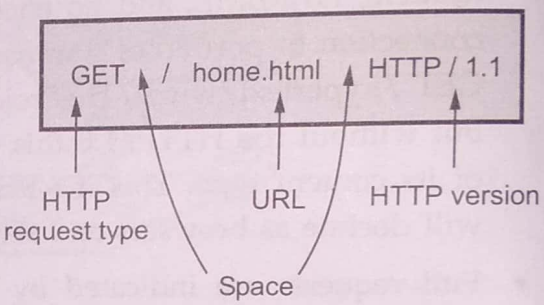


Fig. 2.3.3 Request line

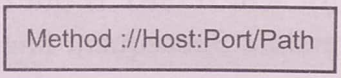


Fig. 2.3.4 (a) URL

Fig. 2.3.4 shows the URL.

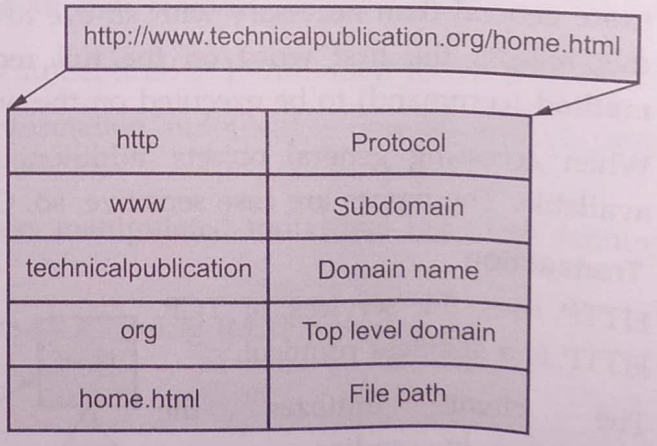


Fig. 2.3.4 (b) URL example

- The method is the prototocol used to retrieve the document. Several different protocols can retrieve a document, among them are FTP and HTTP.
- The host is the computer where the information is located, although the name of the computer can be alias. Web pages are usually stored in computers and computers are given alias names that usually begin with the character www.
- The URL can optionally contain the port number of the server.
- Path is the path name of the file where the information is located.
- The request type field in a request message defines several kinds of messages referred to as methods.

Scanned with CamScanner

Sr. No.	Method	Purposes
1.	GET	Used when the client wants to retrieve a document from the server. Server responds with the contents of the document.
2.	HEAD	Used when client wants some information about a document but not the document itself.
3.	POST	Used by the client to provide some information to the server i.e. input to the server.
4.	PUT	Used by the client to provide a new or replacement document to be stored on the server.
5.	PATCH	Similar to PUT except that the request contains a list of differences that should be implemented in the existing file.
6.	DELETE	Removes a document on the server.
7.	COPY	Copies a files to another location. URL gives the location of the source file.
8.	MOVE	Move a file to another location.
9.	LINK	Creates a link or links from a document to another location.
10.	UNLINK	UNLINK method deletes links created by the LINK method.
11.	OPTION	This method is used by the client to ask the server about available options.

- The GET method requests the server to send the page (by which we mean object in the most general case) suitably encoded in MIME. However, if the GET request is followed by an If-Modified-Since header, the server only sends the data if it has been modified since the data supplied. Using this mechanism, a browser that is asked to display a cached page can conditionally ask for it from the server, giving the modification time associated with the page. If the cache page is still valid, the server just sends back a status line announcing that fact, thus eliminating the overhead of transferring the page again.
- The HEAD method just asks for the message header, without the actual page. This method can be used to get a page's time of last modification, to collect information for indexing purposes, or just to test a URL for validity. Conditional HEAD request do not exist.
- The PUT method is the reverse of GET : Instead of reading the page, it writes the page. This method makes it possible to build a collection of web pages on a remote server. The body of the request contains the page. It may be encoded using MIME, in which case the lines following the PUT might include content type and authentication headers, to prove that the caller indeed has permission to perform the requested operation.
- Somewhat similar to PUT is the POST method. It too bears a URL, but instead of replacing the existing data, the new data is "appended" to it in some generalized

Scanned with CamScanner

sense. Posting a message to a news group or adding a file to a bulletin board system are example of appending in this context. It is clearly the intention here to have the web take over the functionality of the USENET news system.

- DELETE does what you might expect; it removes the page. As with PUT authentication and permission play a major role here. There is no guarantee that DELETE succeeds, since even if the remote HTTP server is willing to delete the page, the underlying file may have a mode that forbids the HTTP server from modifying or removing it.
- The LINK and UNLINK methods allow connections to be established between existing pages or other resources.

Response Message

- Fig. 2.3.5 shows the response message. It contains a status line, a header and body.
- Status line defines the status of the response message. It consists of the
  a. HTTP version b. Space
  c. Status code d. Space e. Status phrase

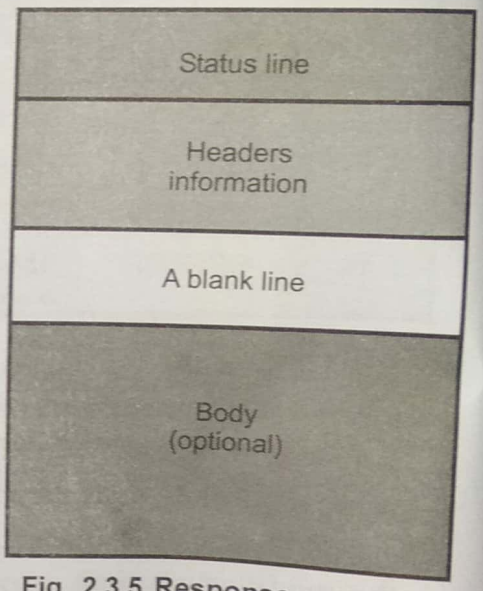


Fig. 2.3.5 Response message

Headers

- Header can be one or more header lines. Each header line is made of a header name, a colon, a space and a header value.
- The header exchange additional information between the client and the server.
- A header line belongs to one of four categories : general header, request header, response header and entity header.
- Fig. 2.3.6 shows the header format.
- *General header* includes general information about the message. Request and a response both contains general header.

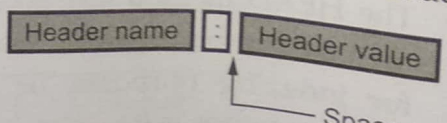


Fig. 2.3.6 Header format

- *Response header* can be present only in a response message. It specifies the servers configuration and special information about the request.
- Request header can be present only in a request message. It specifies the clients configuration and the client preferred document format.
- *Entity header* gives information about the body of the document. It is mostly present in response messages, some request message, such as POST and PUT methods, that contain a body also use this type of header.

Scanned with CamScanner

- Fig. 2.3.7 shows the headers.

Status line	HTTP/1.1 300 OK
General Headers	Date : Wed, 8 Oct 2014 13:00:13 GMT
	Connection : close
Response Headers	Server : Apache /1.3.27
	Accept-Ranges : bytes
Entity Headers	Content-Type : text/html
	Content-Length : 200
	Last-Modified : 2 Oct 2014 13:00:13 GMT
Blank Line	
Message Body	<html>
	<head>
	<title> Welcome to the India <title>
	<head>
	<body>

Fig. 2.3.7 Response message header

2.3.1 Persistent and Non-persistent Connection

- HTTP connections are of two types
  1. Persistent HTTP
  2. Non-persistent HTTP

Non-persistent connections

- In this type of connection, one TCP connection is made for each request / response.
- Suppose the page consists of a base HTTP file and ten JPEG images and that all 11 of these objects reside on the same server.
- Suppose the URL for the base HTML file is
  www.vtubooks.com / ITDept / home.index

The sequence of events are as follows :

1. The HTTP client initiates a TCP connection to the server www.vtubook.com on port number 80. It is default port number for HTTP.
2. HTTP client sends an HTTP request message to the server via the socket. Request message includes the path name/ITDept/home.index.

Scanned with CamScanner

3. HTTP server receives the request message via the socket.
4. HTTP server tells TCP to close the TCP connection.
5. HTTP client receives the response message. The TCP connection terminates.
6. The first four steps are then repeated for each of the referenced JPEG objets.

- As the brower receives the web pages, it displays the page to the user.

Round Trip Time (RTT)

- RTT is the time it takes for a small packet to travel from client to server and then back to the client.
- RTT includes packet propagation delays, packet queuing delays in intermediate routers and switches and packet processing delays.
- Fig. 2.3.8 shows operation when user clicks on a hyperlink.

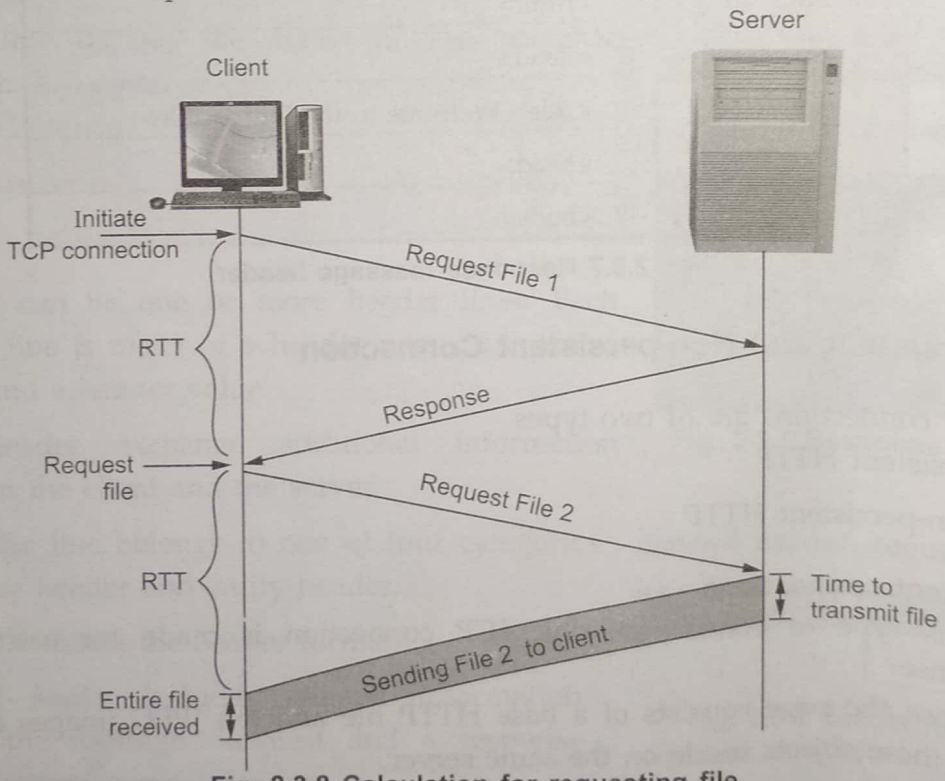


Fig. 2.3.8 Calculation for requesting file

- Browser to initiate TCP connection between the browser and the web server. It requires three way handshake.
- The client sends a small TCP segment to the server.
- The server acknowledges and responds with a small TCP segment.
- Finally, the client acknowledges back to the server.
- The initial design HTTP 1.0 uses nonpersistent connections. The TCP connection is closed after each request/response interaction.

Scanned with CamScanner

- Each subsequent request from the same client to the same server involves the setting up and tearing down of an additional TCP connection.

Disadvantages of non-persistent

1. TCP processing and memory resource wasted in the server and the client.
2. It requires delay of 2 RTT associated with the transfer of each object.
3. Each TCP connection setup involves the exchange of three segments between client and server machines.

Persistent connection

- HTTP 1.1. made persistent connections the default mode.
- The server now keeps the TCP connection open for a certain period of time after sending a response.
- This enables the client to make multiple requests over the same TCP connection and hence avoid the inefficiency and delay of the nonpersistent mode.

Types of persistent connections

- There are two versions of persistent connections :
  1. Without pipelining 2. With pipelining

Without pipelining

- The client issues a new request only when the previous response has been received.
- The client experiences one RTT in order to request and receive each of the referenced objects.
- Disadvantage : TCP connection is idle i.e. does nothing while it waits for another request to arrive. This idling wastes server resoruces.

With pipelining

- Default mode of HTTP 1.1. uses persistent connections with pipelining.
- Client issues a request as soon as it encounters a references. The HTTP client can make back to back requests for the referenced objects.
- It can make a new request before receiving a response to a previous request.
- When the server receives the back-to-back requests, it sends the objects back-to-back.
- It uses only one RTT.
- Pipelined TCP connection remains idle for a smaller fraction of time.

Scanned with CamScanner

- **Persistent HTTP** connections have a number of **advantages.**
  1. By opening and closing fewer TCP connections, CPU time is saved in routers and hosts.
  2. Requests and responses can be pipelined on a connection.
  3. Network congestion is reduced by reducing the number of packets caused by TCP opens.
  4. Latency on subsequent requests is reduced.

Proxy server

- HTTP supports the proxy servers. A proxy server is a computer that keeps copies of responds to recent requests.
- The HTTP client sends a request to the proxy server. The proxy server checks its cache.
- If the response is not stored in the cache, the proxy server sends the request to the corresponding server.
- Incoming responses are sent to the proxy server and stored for future requests from other clients.
- The proxy server reduces the load on the original server, decreases traffic and improves latency.
- To use proxy server, the client must be configured to access the proxy instead of the target server.

2.3.2 Difference between Persistent and Non-persistent

Sr. No.	Persistent HTTP	Non-persistent HTTP
1.	Persistent version is 1.1.	Non-persistent HTTP version is 1.0.
2.	It uses one RTT.	It uses two RTT.
3.	TCP connection is not closed.	TCP connection is closed after every request-response.
4.	Client make multiple request over the same TCP connection.	Client make multiple request over the multiple TCP connection.
5.	It is default mode.	It is not default mode.
6.	Request methods are GET, HEAD, POST, PUT, DELETE, TRACE and OPTIONS.	Request methods used are GET, POST and HEAD.

Scanned with CamScanner

Review Questions

1. *Write note on following : 1. HTTP* **GTU : Winter-14, Marks 4**
2. *Describe the built in HTTP request methods.* **GTU : May-12, Marks 7**

2.4 Domain Name System GTU : Winter-12,13,14, Dec.-10,11, June-11,Summer-14

- **Goal :** Assign meaningful high-level names to a large set of machines and handle the mapping of those names to a machine's **IP** address.
- The DNS is a distributed database that resides on multiple machines on the internet and used to convert between names and address and to provide e-mail routing information.
- DNS provides the protocol that allows the client and servers to communicate with each other.
- Domain names are case insensitive so **com** and **COM** mean the same thing.
- The **DNS** protocol runs over **UDP** and uses port 53.
- The **DNS** is specified in **RFC 1034** and **RFC 1035.**
- The **DNS** protocol is the application layer protocol.
- A full domain name is a sequence of labels separated by dots (.).
- The **DNS** name space is hierarchical and it is similar to the unix file system.

Originally, the internet was small and mapping between names and addresses was accomplished using a centrally-maintained file called *hosts.txt*. To add a name or change an address required contacting the central administrator, updating the table, and distributing it to all the other sites. This solution worked at first because most sites had only a few machines, and the table didn't require frequent changes.

The centrally-maintained table suffered from several drawbacks :

1. The name space was *flat*, and no two machines could use the same machine name.
2. As the internet grew, changes to the database took days to weeks to take effect.
3. The central site became congested with the increase in the number of sites retrieving copies of the current table.
4. The internet grew at an astonishing rate.

The Domain Name System (DNS) is a hierarchical, distributed naming system designed to cope with the problem of explosive growth :

1. It is *hierarchical* because the name space is partitioned into *subdomains*.

Scanned with CamScanner

2. It is distributed because management of the name space is delegated to local sites. Local sites have complete control (and responsibility) for their part of the name space. DNS queries are handled by servers called *name servers.*
3. It does more than just map machine names to internet addresses. For example, it allows a site to associate multiple machines with a single, mailbox name.

In the DNS, the name space is structured as a tree, with *domain names* referring to nodes in the tree. The tree has a *root,* and a *fully-qualified* domain name is identified by the *components* of the path from the domain name to the root.

Services provided by DNS :

- **Host aliasing :** A host with complicated hostname can have one or more alias names. DNS can be invoked by an application to obtain the canonical hostname for a supplied alias hostname as well as the IP address of the host.
- **Mail server aliasing :** DNS can be invoked by a mail application to obtain the hostname for a supplied alias hostname as well as the IP address of the host.
- **Load distribution :** DNS is also used to perform load distribution among replicated servers.

2.4.1 Components of DNS

DNS includes following components

1. Domain
2. Domain name
3. Name server
4. Name resolver
5. Name cache
6. Zone

1) For example, vtubooks.com is the site for technical publications. Here com is the domain.
2) Domain name is defined by the DNS as being the sequence of names and domain. For example, vtubooks.com could be domain name.
3) In name server, software (program) that maps names to addresses. It does this by mapping domain names to IP addresses.
4) Name resolver is a software that functions as a client interacting with a name server.
5) Name cache is the storage used by the name resolver to store information frequently used.
6) Zone is a contiguous part of a domain.

Scanned with CamScanner

2.4.2 DNS in the Internet

DNS is divided into three different sections in the internet i.e. Generic domain, Country domain and Inverse domain.

- Fig. 2.4.1 shows the DNS in the internet.

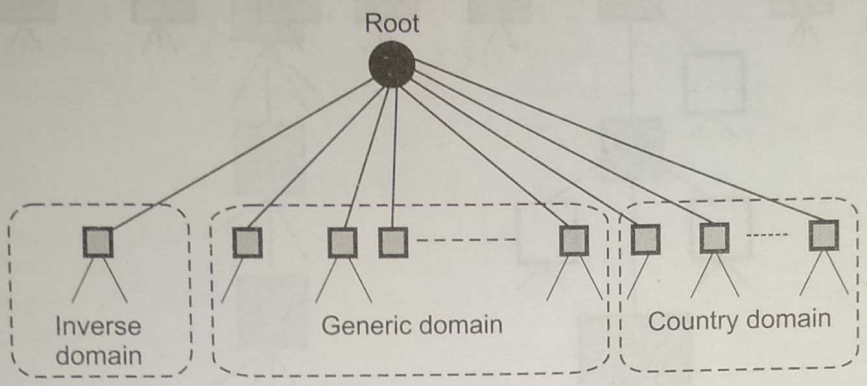


Fig. 2.4.1 DNS in the internet

Generic Domains

- Each node in the tree defines a domain, which is an index to the domain name space database.
- Generic domain labels are as follows

Sr. No.	Label	Description
1.	com	Commercial organization
2.	edu	Educational organization
3.	gov	Government Institutions
4.	int	International organizations
5.	mil	Military group
6.	net	Network support centers
7.	org	Nonprofit organization

Scanned with CamScanner

- Fig. 2.4.2 shows the generic domains

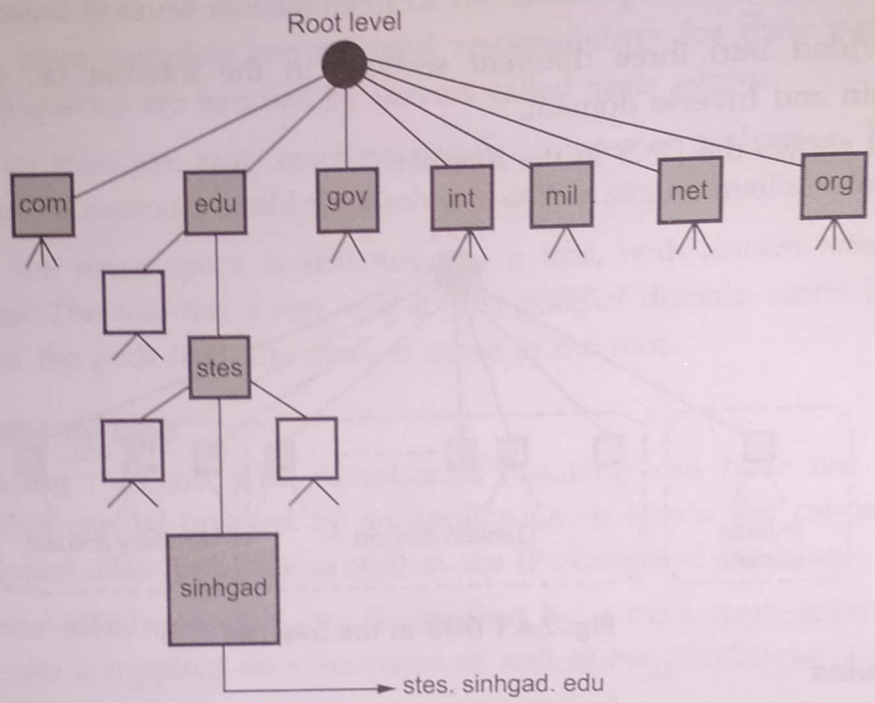


Fig. 2.4.2 Generic domains

Country Domains

- It uses two character country abbreviations at first level. Second level labels can be more specific, national destinations. For India, the country domain is in.
- Fig. 2.4.3 shows country domains.

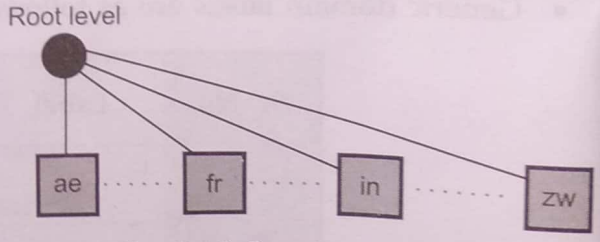


Fig. 2.4.3 Country domains

Inverse Domain

- Used to map an address to a name.
- Example : When a client send a request to the server for doing a particular task, server finds the list of authorized client. The list contains only IP address of the client.
- Server send a query to the inverse DNS server and ask for a mapping of address to name for authorized client list.
- The above query is called an inverse or pointer query.
- The pointer query is handled by the first level node called arpa. The second level is also one single node named in-addr. The rest of the domain defines IP addresses.

Scanned with CamScanner

Fig. 2.4.4 shows inverse domain.

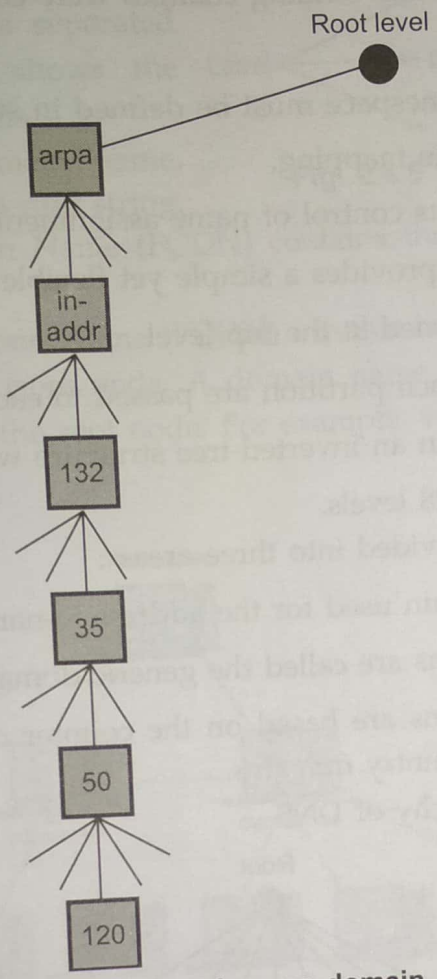


Fig. 2.4.4 Inverse domain

2.4.3 Name Spaces

- Name spaces are of two types : Flat name spaces and Hierarchical names.
- The name assigned to machines must be carefully selected from a name space with complete control over the binding between the names and IP addresses.

i) Flat name spaces :

- The original set of machines on the Internet used flat namespaces.
- These namespaces consisted of sequence of characters with no further structure.
- A name is assigned to an address.
- **Advantage :**
  1. Names were convenient and short.
- **Disadvantages :**
  1. Flat name spaces cannot generalize to large sets of machines because of the single set of identifiers.

Scanned with CamScanner

2. Single central name authority was overloaded.
3. Frequent name-address binding changes were costly and cumbersome.

ii) Hierarchical names

- The partitioning of a namespace must be defined in such a way that it :
  - Supports efficient name mapping.
  - Guarantees autonomous control of name assignment.
- Hierarchical namespaces provides a simple yet flexible naming structure.
- The namespace is partitioned at the top level.
- Authority for names in each partition are passed to each designated agent.
- The names are designed in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels.

The top level domains are divided into three areas :

1. Arpa is a special domain used for the address-to-name mappings.
2. The 3 character domains are called the generic domains.
3. The 2 character domains are based on the counter codes found in ISO 3166. These are called the country domains.

- Fig. 2.4.5 shows the hierarchy of DNS.

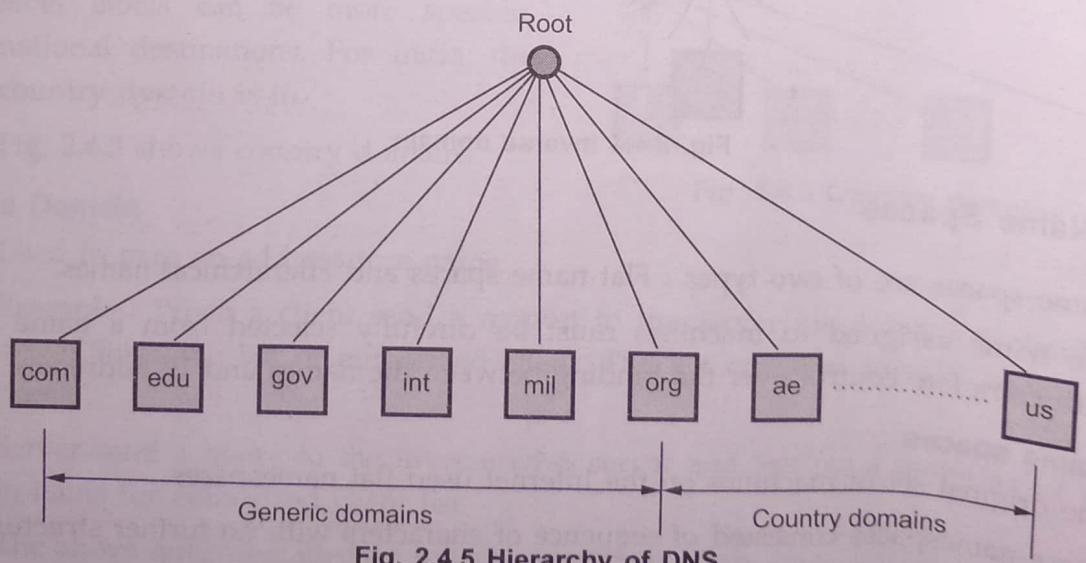


Fig. 2.4.5 Hierarchy of DNS

2.4.4 Domain Name Space

- In DNS, names are defined in an inverted tree structure with the root at the top. The tree can have only 128 levels : Level 0 to Level 127.
- Each node in the tree has a label, which is a string with a maximum of 63 characters. The root label is a null string , i.e. empty string.

Scanned with CamScanner

- Each node in the tree has a domain name, a full domain name is a sequence of labels separated by dots(.). Fig. 2.4.6 shows the domain names and labels.

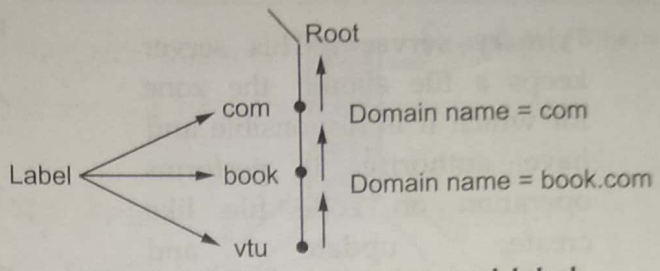


Fig. 2.4.6 Domain names and label

- In fully qualified domain name, label is terminated by a null string. Fully Qualified Domain Name (FQDN) contains the full name of host. All labels are part of FQDN.
- Partially Qulified Domain Name (PQDN) : In this label is not terminated by a null string. It always start from node. A domain name does not include all the levels between the host and the root node. For example, vtu.book.com.

Hierarchy of Name Servers

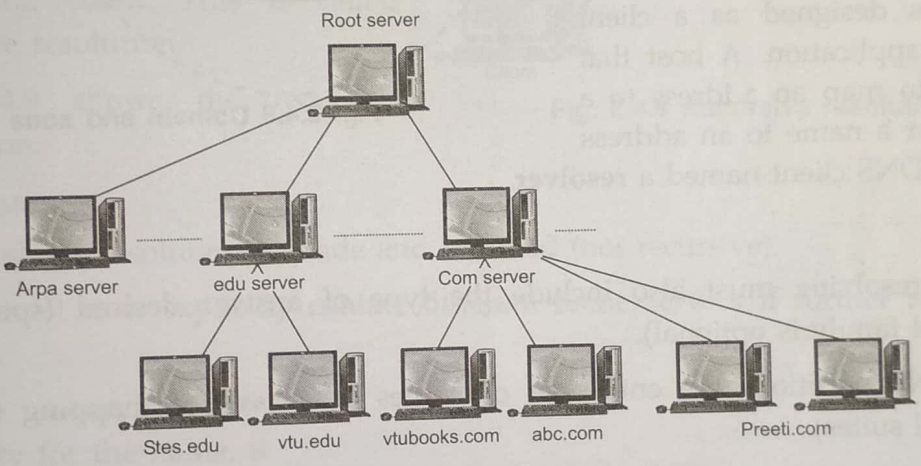


Fig. 2.4.7 Hierarchy of name server

- To distribute the information among many computers, DNS servers are used. Creates many domains as there are first level nodes. Fig. 2.4.7 shows hierarchy of name servers.
- **Zone :** Server have some authority and also responsible for operation. Server creates database, which is called zone file. Server maintain all the information about node of that domain.
- Fig. 2.4.8 shows domain with zone.
- Domain and zone are same if server accepts responsibillity for a domain and does not divide the domain into **subdomian.**
- Domain and zone are different, if a server divides its domain into subdomains and delegates part of its authority to other server.
- **Root server :** If zone consists of the full tree then that zone server is called root server. Root server do not maintain any information about domains.
- DNS uses two types of servers :
  1. Primary server 2. Secondary server

Scanned with CamScanner

- **Primary server :** This server keeps a file about the zone for which it is responsible and have authority. It performs operation on zone file like create, update and maintaining.
- **Secondary server :** It loads all information from the primary server. Secondary server can not perform any operation on zone file.

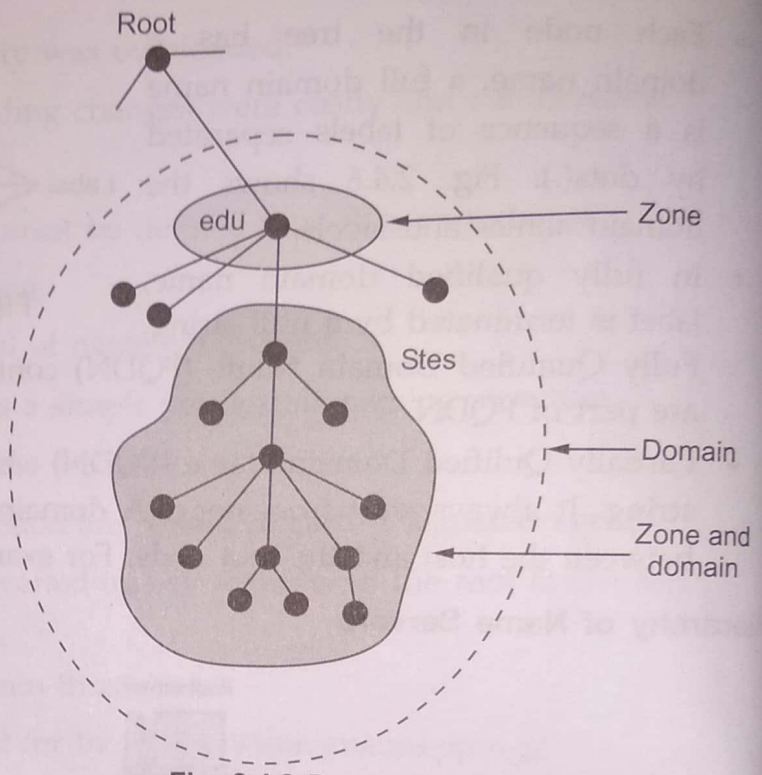


Fig. 2.4.8 Domain and zone

2.4.5 Resolution

- DNS is designed as a client server application. A host that needs to map an address to a name or a name to an address calls a DNS client named a **resolver**.

Working :

- Name resolving must also include the type of answer desired (specifying the protocol family is optional).
- The DNS partitions the entire set of names by class (for mapping to multiple protocol suites).
- Naming items is required since one cannot distinguish the names of subdomains from the names of individual objects or their types.

Mapping Domain Names to Addresses :

a) The DNS also includes an efficient, reliable, general purpose, distributed system for mapping names to addresses using an independent co-operative system called name servers.

b) Names Servers - are server programs that translate names-to-addresses (maps DN => IP addresses) and usually executes on a dedicated processor.

c) Name Resolvers - client software that uses one or more name servers in getting a mapped name.

d) Domain name servers are arranged in a conceptual tree structure that corresponds to the naming hierarchy.

Scanned with CamScanner

Recursive Resolution

- A client request complete translation.
- If the server is authority for the domain name, it checks its database and responds.
- If the server is not authority, it sends the request to another server and waits for the response.
- When the query is finally resolved, the response travel back until it finally reaches the requesting client. This is called recursive resolution.
- Fig. 2.4.9 shows the recursive resolution.

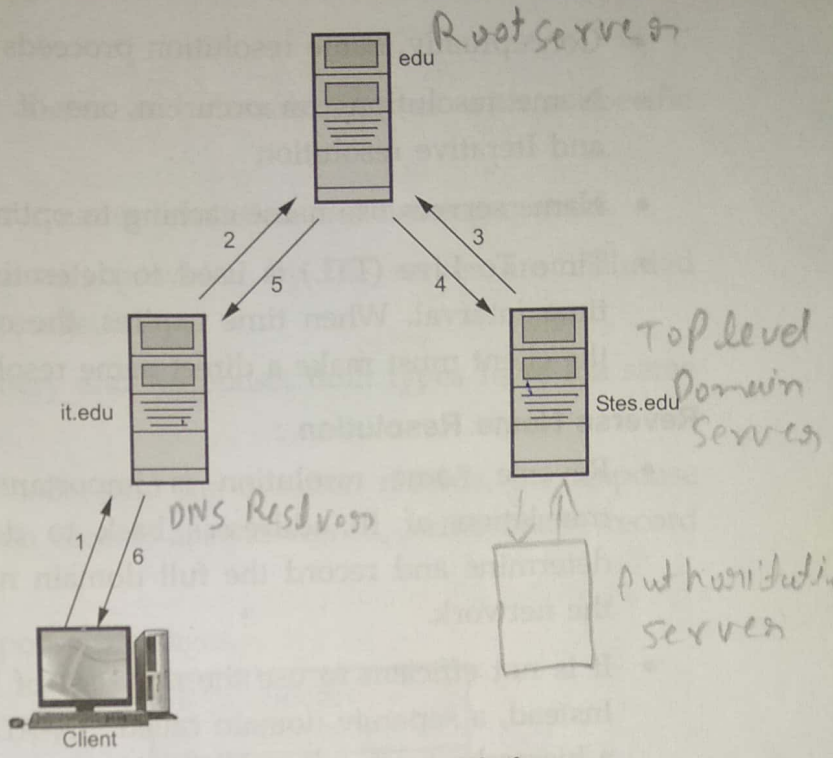


Fig. 2.4.9 Recursive resolution

Iterative Resolution

- Only a single resolution is made and returned (not recursive).
- Client must now explicitly contact different name servers if further resolution is needed.
- If the server is an authority for the name, it sends the answer. If it is not, it returns the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. This process is called iterative resolution because the client repeats the same query to multiple servers.
- Fig. 2.4.10 shows iterative resolution.

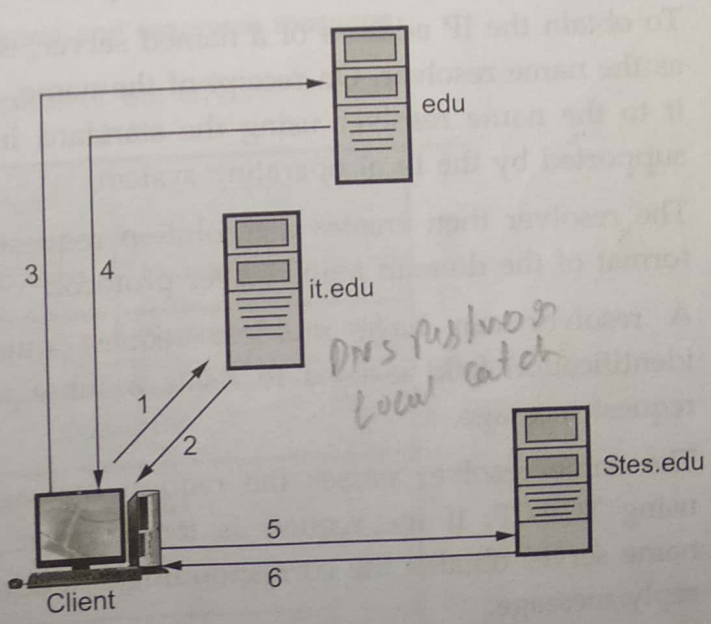


Fig. 2.4.10 Iterative resolution

Scanned with CamScanner

- Conceptually, name resolution proceeds in a top-down fashion.
- Name resolution can occur in one of two different ways : Recursive resolution and Iterative resolution
- Name servers use name caching to optimize search costs.
- Time To Live (TTL) is used to determine a guaranteed name binding during it's time interval. When time expires, the cache name binding is no longer valid, so the client must make a direct name resolution request once again.

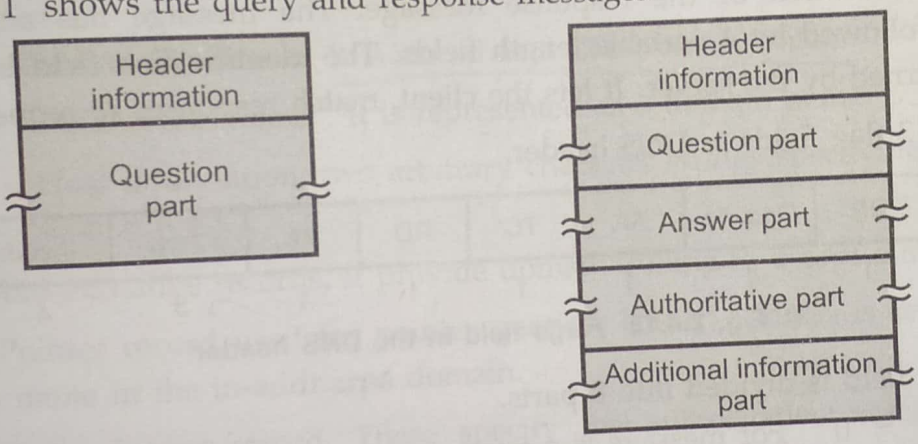
Reverse Name Resolution :

- Reverse name resolution is important task of DNS on the internet or the translation of IP addresses back to domain names. For example, servers can determine and record the full domain name of machine connecting to them over the network.
- It is not efficient to use the same set of DNS records for reverse name resolution. Instead, a separate domain called "IN-ADDR.ARPA" has been set aside to provide a hierarchy for translating IP addresses into names.
- A DNS lookup of "barg.oo.msstate.edu" would reveal it has the IP address "130.19.60.10". If one has the IP address and wishes to know the name, one must perform a DNS lookup of "10.60.19.130. in -addr.arpa", which will return the name.
- Reverse name resolution fields use the PTR resource record, which points to the correct position in the normal DNS space. The hierarchy under "IN-ADDR.ARPA" can be delegated of course just like any other domain.
- To obtain the IP address of a named server, each host has a client protocol known as the name resolver. On receipt of the name, the client application protocol posses it to the name resolver using the standard interprocess communication primitive supported by the local operating system.
- The resolver then creates a resolution request message in the standard message format of the domain name server protocol.
- A resolver can have multiple request outstanding at any time. Hence the identification field is used to relate a subsequent response message to an earlier request message.
- The name resolver passes the request message to its local domain name server using TCP/IP. If the request is for a server on this network, the local domain name server obtains the corresponding IP address from its DIB and returns it in a reply message.

Scanned with CamScanner

2.4.6 Message Format

- Messages are sent between domain clients and domain servers with a specific format.
- All messages of this format are used for name resolution and naming queries.
- Question sent by the client and answers provided by the server are included within different fields of the same message.
- DNS has two types of messages : Query and Response. Both types have the same format.
- The query message consists of the header and the question records, the response message consists of a header, question record, answer record, authoritative record and additional records.
- Fig. 2.4.11 shows the query and response messages.



(a) Query (b) Response

Fig. 2.4.11 Query and response message

- Fig. 2.4.12 shows the header format of the DNS.

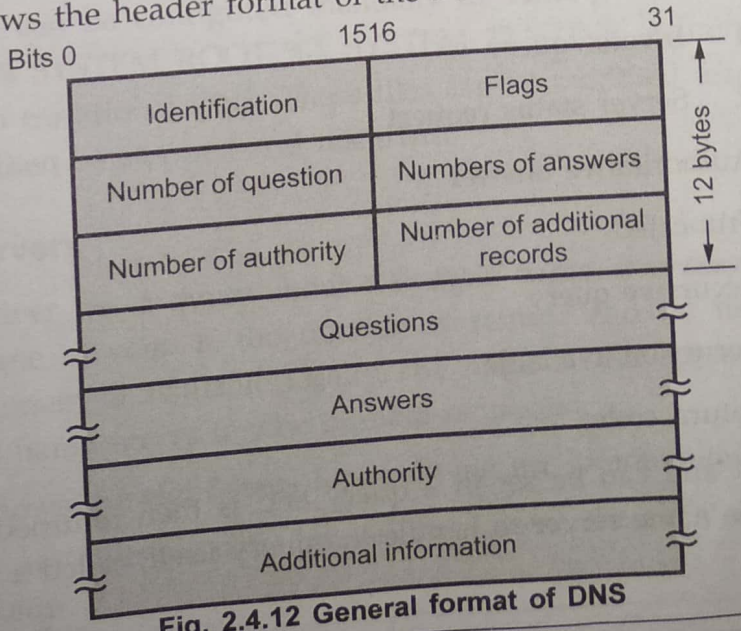


Fig. 2.4.12 General format of DNS

Scanned with CamScanner

- **Identification :** It is 16 bits fields and unique value used by the client to match responses to queries.
- **Flags :** It is the collection of subfields that define the type of messages and type of the answers requested and so on.
- Number of question record contains the number of queries in the question section of the message.
- Number of answer record contains the number of answer records in the answer section of the response message.
- Number of authority record contains the number of authority records in the authoritative section of the response message.
- Number of additional records contains the number of additional records in the additional section of the response message. The message has a fixed 12-byte header followed by 4 variable length fields. The identification field is set by client and returned by the server. It lets the client, match responses to requests.
- Fig. 2.4.13 flag fields in DNS header.

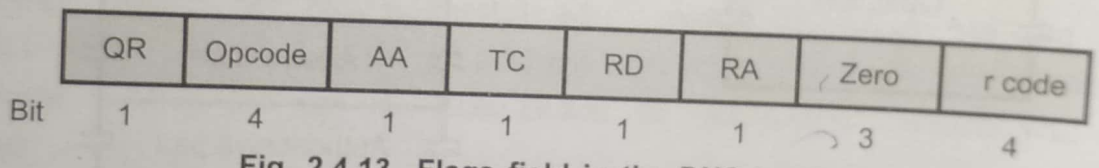


Fig. 2.4.13 Flags field in the DNS header

- The flags field is divided into 8 parts.

QR = 0 For message is a query

= 1 It is response

Opcode = 0 Standard query

= 1 Inverse query

= 2 Server status request

AA = Authoritative answer

TC = Truncated

RD = Recursive query

RA = Recursion available

r code = Return code

- RD field is 1-bit and can be set in a query and is then returned in the response. This flag tells the name server to handle the query itself, called a recursive query.

Scanned with CamScanner

- RA is a 1-bit field and set to 1 in the response if the server support recursion. There is a 3-bit field that must be zero.
- r code is a 4-bit field. The common value are 0 for no error and 3 for name error. A name error is returned only from an authoritative name server and means the domain name specified in the query does not exist.
- The next four 16-bit fields specify the number of entries in the four variable length fields that complete the record.

2.4.7 Resource Records

- Different types of resource records are used in DNS. An IP address has a type of A and PTR means pointer query.
- There are about 20 different types of resource records available. Some PR are listed below.

1) A = It defines an IP address. It is stored as a 32-bit binary value.
2) CNAME = "Canonical name". It is represented as a domain name.
3) HINFO = Host information, two arbitrary character strings specifying the CPU and operating system (OS).
4) MX = Mail exchange records. It provide domain willing to accept e-mail.
5) PTR = Pointer record used for pointer queries. The IP address is represented as a domain name in the in-addr.arpa domain.
6) NS = Name Server record. These specify the authoritative name server for a domain. They are represented as domain names.

A) Configuration of DNS :

The DNS server can be configured manually by editing files in the default WINNT installation path \% SYSTEM ROOT %\ SYSTEM 32 \ DNS. Administration is identical to administration in traditional DNS. These files can be modified using a text editor. The DNS service must then be stopped and restarted.

2.4.8 Name Servers

- When a resolver has a query about a domain name, it passes the query to one of the local name servers. If the domain is remote and no information about the requested domain is available locally, the name server sends a query message to the top level name server for the domain requested.
- Fig. 2.4.14 shows the eight steps for resolving the remote name.
- A resolver on flits.cs.vu.nl wants to know the IP address of the host linda.cs.yale.edu.

Scanned with CamScanner

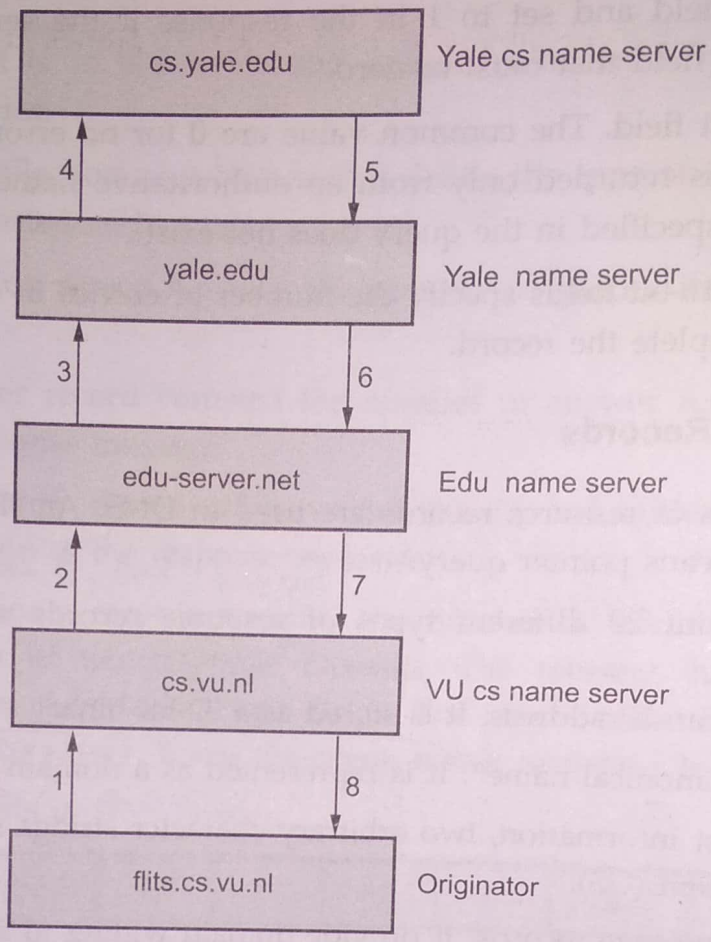


Fig. 2.4.14 Remote name resolve

Steps

1. It sends a query to the local name server cs.vu.nl. This query contains the domain name, sought, the type (A) and the class (IN).
2. and 3. Suppose the local name server has never had a query for this domain before and knows nothing about it. It may ask a few other nearby name servers, but if none of them know, it sends a UDP packet to the server for edu given in its database, edu-server.net. This server knows all its children, so it forwards the request to the name server for yale.edu.
4. This forwards the request to cs.yale.edu, which must have the authoritative resource records.
5. to 8. Each request is from a client to a server, the resource record requested works its way back in these steps.

2.4.9 LDAP

- LDAP is Lightweight Directory Access Protocol. It provides X-500 features. LDAP is an application-level protocol that is implemented directly on top of TCP.
- It stores entries, which is similar to objects. Each entry must have a distinguished name, which un-equally identifies the entry. Entries can also have attributes.

Scanned with CamScanner

- LDAP provides binary, string and time types. It allows the defination of object classes with attribute name of types. Entries are organized into a directory information tree, according to their distinguished names.
- LDAP defines a network protocol for carrying out data defination and manipulation.
- LDAP has been widely adopted, particularly for internet directory services. It provides secured access to directory data through authentication.

2.4.10 Dynamic Domain Name System (DDNS)

- DDNS is a service that maps internet domain names to IP addresses. DDNS serves a similar purpose to DNS : DDNS allows anyone hosting a Web or FTP server to advertise a public name to prospective users.
- Unlike DNS that only works with static IP addresses, DDNS works with dynamic IP addresses, such as those assigned by an ISP or other DHCP server. DDNS is popular with home networkers, who typically receive dynamic, frequently-changing IP addresses from their service provider.
- To use DDNS, one simply signs up with a provider and installs network software on their host to monitor its IP address.
- Compared to ordinary DNS, the disadvantage of DDNS is that additional host software, a new potential failure point on the network, must be maintained.

Review Questions

1. *What is the role of Domain Name Server (DNS) in Internet ? Explain the hierarchy of various domain names.* **GTU : Winter-14, Marks 7**
2. *How does DNS work ? Explain.* **GTU : Dec.-10, Marks 7**
3. *Explain DNS in detail with example.* **GTU : June-11, Marks 7**
4. *Explain : DNS and its advantages.* **GTU : Dec.-11, Marks 5**
5. *Explain the domain name system.* **GTU : Winter-12, Marks 7**
6. *What is a resource record ? How it is useful for DNS ?* **GTU : Winter-13, Marks 7**
7. *Explain : DNS and its use.* **GTU : Summer-14, Marks 4**

GTU : Dec.-11, Winter-12,13

2.5 World Wide Web

- World wide web is collection of millions of files stored on thousands of servers all over the world. These files represent documents, pictures, video, sounds, programs, interactive environments.
- Following are hardware, software and protocols that make up the web.

Scanned with CamScanner

1. A web server is a computer connected to the Internet that runs a program that takes responsibility for storing, retrieving and distributing some of the web files. A web client (web browser) is a computer that requests files from the web.
2. Well-defined set of languages and protocols that are independent of the hardware or operating system are required to run on the computers.
3. The Hyper Text Markup Language (HTML) is the universal language of the web.
4. Java is a language for sending small applications over the web. Java script is a language for extending HTML to embed small programs called scripts in web pages. The main purpose of Java and scripts is to speed up the interactivity of web pages.
5. VB script and Activex controls are microsoft system that work with IE.
6. Pictures, drawings, charts and diagrams are displayed on web using image formats such as JPEG and GIF formats.
7. The Virtual Reality Modeling Language (VRML) is the web's way of describing three-dimensional objects.

- A web page is an HTML document that is stored on a web server. A web site is a collection of web pages belonging to a particular organization.
- URL of these pages share a common prefix, which is the address of the home page of the size. Search engines are a bottom-up approach for finding your way around the web. Some search engines search only the titles of web pages. While other search every word. Keywords can be combined with Boolean operations, such as AND, OR and NOT, to produce rather complicated queries.
- Home page is the front door of a web site. When a person or organization says "My web site is at www.sangeeta.com", the URL to which they refer is the URL of the site's home page. The home page introduces the rest of the web site and provides links that leads to other pages on the site.

2.5.1 Web Browsers

It is a system of creating organizing and linking of document

- A web browser is a program. Web browser is used to communicate with web servers on the Internet, which enables it to download and display the webpages. Netscape Navigator and Microsoft Internet Explorer are the most popular browser softwares available in market. Browser interact with web as well as computer operating system and with other programs.

Scanned with CamScanner

- Internet explorer is the default browser in newly installed window 98 systems. Most browser windows have the same basic layout. Some of the basic elements are -
  - Menu bar
  - Tool bar
  - Address or location window
  - Viewing window
  - Status bar

Some web pages are divided into independent pages, called frames.

- The purpose of the web browser is to display web pages, which may either arrive over the Internet. Web browser can be used to view files of any common web format that are stored on the user system. Window, Macintosh and some Unix desktops support the default web browser.
- There are different ways for opening the web page.
  1. Enter its URL into the address or location box of a web browser.
  2. Select it from the list that drops down from the address.
  3. Link to it from another web page.
  4. Link to it from a mail message or newsgroup article.

2.5.2 Working of WWW

- www uses client-server interaction. The browser program acts as a client that uses the Internet to contact a remote server for a copy of the requested page. The server on the remote system returns a copy of the page along with the additional information.
- The additional information a www server returns tells the browser two important things.
  1) It describes how to display the information.
  2) It gives a URL for each selectable item on the page.
- When a browser receives a page from a remote server, it displays the page and then waits for the user to select one of the highlighted items. Once a user makes a selection, the browser consults the hidden information that arrived with the page to find the URL that corresponds to the selection. The browser then uses the Internet to obtain the newly selected page of information.
- Each URL uniquely identifies a page of information by giving the name of a remote computer, a server on that computer and a specific page of information available from the server. Fig. 2.5.1 illustrates how the URL encodes the information.

Scanned with CamScanner

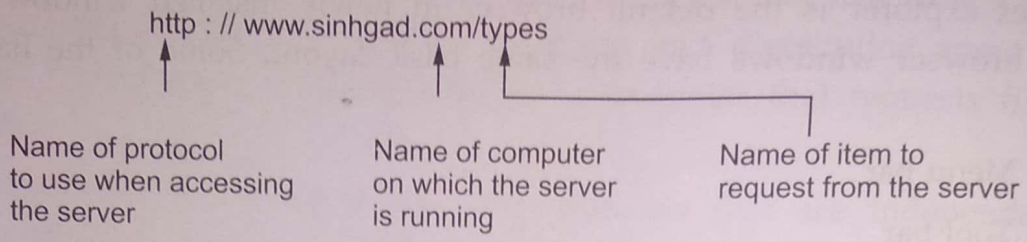


Fig. 2.5.1 Format of URL

- World Wide Web was developed using the client server architecture, which ensured cross-platform portability. The www is officially described as a "Wide area hypermedia information retrieval initiative". It is an information system that links data from many different Internet services under one set of protocols.
- Web clients, called browsers, interpret Hyper Text Markup Language documents delivered from web servers.
- The world wide web is a distributed, multimedia, hyper text system. It is distributed since information on the web can be located on any computer system connected to the Internet around the world.
- It is multimedia because the information it holds can be in the form of text, graphics, sound and video.
- Hyper text means that the information is available using hyper text technique, which involves selecting highlighted phrases or images that one selected retrieve information related to the selected highlighted subject.
- The information being retrieved can be information located any where in the world. The normal way to provide information on the world wide web is by writing documents in HTML (Hyper Text Markup Language).

1. The Client Side

- When a user clicks on a hyperlink, the brower carries out a series of steps in order to fetch the page pointed to.
  1. The brower determines the URL.
  2. The brower asks DNS for the IP address of www.vtubooks.com.
  3. DNS replies with 172.16.16.1.
  4. The brower makes a TCP connection to port 80 on 172.16.16.1.
  5. It then sends over a request asking for file/home/index.html.
  6. The www.vtubooks.com server sends the file/home/index.html.
  7. TCP connection is released.
  8. The brower displays all the text in home/index.html.
  9. The brower fetches and displays all images in this file.

Scanned with CamScanner

- Fig. 2.5.2 shows the web model.

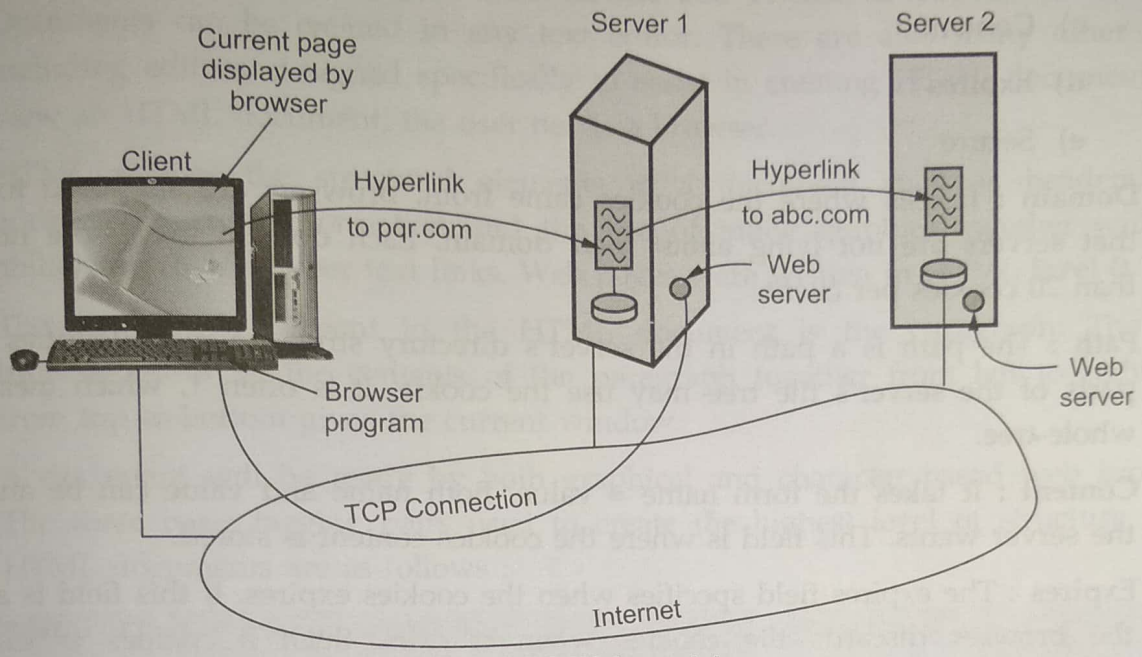


Fig. 2.5.2 Web model

2. The Server Side

- The steps that the server performs.
  1. Accept a TCP connection from a client browser.
  2. Get the name of the file required.
  3. Get the file.
  4. Return the file to the client.
  5. Release the TCP connection.

2.5.3 Statelessness and Cookies

- The web is basically stateless. There is no concept of a login session. The browser sends a request to a server and gets back a file. Then the server forgets that it has ever seen that particular client.
- When a client requests a web page, the server can supply additional information along with the requested page. This information may include a cookie, which is a small file. Browsers store offered cookies in a cookies directory on the client's hard disk unless the user has disabled cookies.
- Cookies are just files or strings, not executable programs. In principle, a cookie could contain a virus, but since cookies are treated as data there is no official way for the virus to actually run and do damage.
- A cookie may contain upto five fields.
  a) Domain

Scanned with CamScanner

b) Path

c) Content

d) Expires

e) Secure

a) Domain : It tells where the cookies came from. Browsers are supposed to check that servers are not lying about their domain. Each domain may store no more than 20 cookies per client.

b) Path : The path is a path in the server's directory structure that identifies which parts of the server's file tree may use the cookie. It is often 1, which means the whole tree.

c) Content : It takes the form name = value. Both name and value can be anything the server wants. This field is where the cookies content is stored.

d) Expires : The expires field specifies when the cookies expires. If this field is absent, the browser discards the cookies when it exits. Such a cookies is called a **nonpersistent cookie.** If a time and date are supplied, the cookie is said to be **persistent** and is kept until it expires.

e) Secure : This field can be set to indicate that the browser may only return the cookie to a secure server. This feature is used for e-commerce, banking and other secure applications.

Examples of cookies

Domain	Path	Content	Expires	Secure
toms-casino.com	/	customer ID=497793521	15-10-02, 17:00	Yes
joes-store.com	/	cart=1-00501; 1-07031; 2-13721	11-10-02, 15:20	No
sneaky.com	/	user ID=3456789	30-12-06, 11:00	Yes

2.5.4 Static Web Documents

- Hyper Text Markup Language (HTML) is intended as a common medium for typing together information from widely different sources. HTML documents are the Standard Generalized Markup Language (SGML) documents with generic semantic that are appropriate for representing information from a wide range of applications.

Scanned with CamScanner

- HTML documents are in plain text format that contain embedded HTML tags. Documents can be created in any text editor. There are also many other tools, including editors, designed specifically to assist in creating HTML documents. To view an HTML document, the user needs a browser.
- HTML defines the structural elements in a document such as headers, and addresses, layout information and the use of inline graphics together with the ability to provide hyper text links. Web pages were written in HTML level 0.
- The most basic element in the HTML document is the paragraph. The web browser flows all the contents of the paragraph together from left to right and from top to bottom gives the current window.
- A document will be ready by both graphical and character based web browser. The three basic tagging pairs used to create the highest level of structure in an HTML documents are as follows :

```
  <HTML> HTML documents </HTML>
  <HEAD> Header information of document </ HEAD>
  <BODY> Body of the HTML document </ BODY>
```

The general structure of the HTML is

```
<HTML>
 <HEAD>
  <TITLE>
       Title here
     </ TITLE>
     </ HEAD>
     <BODY>
          Body element and content
     </ BODY>
          </ HTML>
```

A simple HTML document is given below.

```
<HTML>
<HEAD>
<TITLE>  Communication Network </ TITLE>
</ HEAD>
<BODY>
<H/> Information about the communication network </H/>
<P> Information about the communication network is available
<A HREF :"http ://www.technicalpublicationspune.com"></A></P>
</BODY>
</HTML>
```

- Structural elements in the document are identified by Start and End tags. For example the <TITLE> and </TITLE> tags are used to specify the title of the document.

Scanned with CamScanner

- The <H/> and </H/> tags are used to define the first level heading. Headings are generated by an <Hn> tags, where n is a digit in the range 1 to 6. <H/> is the most important heading and <H6> is the less important. Typically the lower numbered heading will be displayed in a larger and heavier font.
- The browser may also choose to use different colors for each level of heading. Typically <H1> headings are large and bold face with at least one blank line above and below.
- In contrast <H2> headings are in a smaller font, and with less space above and below. The <BR>, <P> and <HR> tags all indicate a boundary between sections of text.
- The precise format can be determined by the style sheet associated with the page. The <BR> tag just forces a line break. <P> starts a paragraph, which might for example, insert a blank line and possibly some indentation. <HR> (horizontal-rule) tag forces the browser to generate a horizontal rule or line, across the display. It breaks pages into logical sections and is useful when creating forms. There is no equivalent vertical rule.

Advantages and Disadvantages of HTML

A) Advantages of HTML :

1. Applications are quickly developed, requiring substantially less time than is required when creating programs with languages such as C and Pascal.
2. Web applications are easy to maintain and update without disrupting the network data traffic.
3. Developing applications in HTML takes advantage of HTML general compatibility. Web applications can access other company data servers, such as FTP and WAIs databases.
4. Collecting information with HTML.
5. Web viewer request a document from a web site, its server sends the data and the connection between the two computers is dropped. This is the client server relationship. This relationship reduces the amount of time a server spends serving a client freeing it to serve other users.

B) Disadvantages :

1. **Locking :** HTML is not a compiled data format. Web pages cannot be locked. Users have free and open access to look at HTML sources.
2. **Security :** Information is easily accessible and travels unimpeded between hosts and desktops. The cost of this freedom is lack of inherent security.

Scanned with CamScanner

2.5.4.1 XML and XSL

- HTML does not provide any structure to web pages. HTML mixes the content with the formatting with web pages in HTML, it is very difficult for a program to search particular word.
- To overcome this problem, two new language Extensible Markup Language (XML) and Extensible Style Language (XSL) are used. The XML and XSL specifications are much stricter than HTML specification.
- Web pages in XML and XSL are still static since they simply contain instructions to the browser about how to display the page, just as HTML pages do. XML allows the web site designer to make up definition files in which the structures are defined in advance. Defination files can be included, making it possible to use them to build complex web pages.

2.5.4.2 XHTML

- XHTML is new web standard and should be used for all new web pages to achieve maximum portability across platforms and browsers.
- Difference between XHTML and HTML are as follows :
  1. XHTML pages and browsers must strictly conform to the standard.
  2. All tags and attributes must be in lower case.
  3. Closing tags are required even for </p>.
  4. Attributes must be contained within quotation marks.
  5. Tags must nest properly.
  6. Every document must specify its type.

2.5.5 Dynamic Web Documents

Server side dynamic web page generation. Fig. 2.5.3 shows processing of information in HTML.

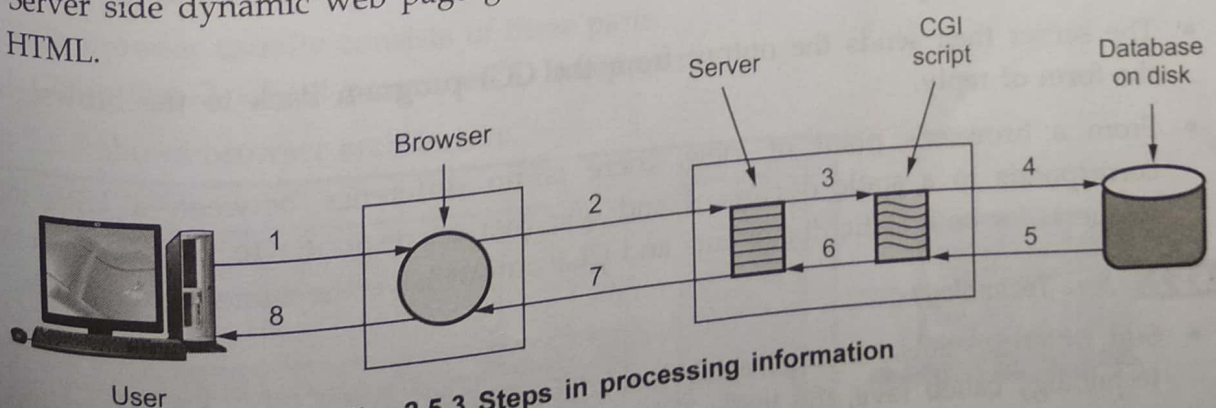


Fig. 2.5.3 Steps in processing information

1. User fills in form.
2. Form sent back.

Scanned with CamScanner

3. Handed to CGI.
4. CGI queries database.
5. Record found.
6. CGI builds pages.
7. Page returned.
8. Page displayed.

2.5.5.1 Common Gateway Interface

- CGI makes dynamic computation of web pages possible. It allows a web server to associate some URLs with computer program instead of static documents on disk.
- When a browser request one of the special URLs the server runs the associated computer program and sends the output from the program back to the user. A server can have an arbitrary number of CGI programs that perform different computations.

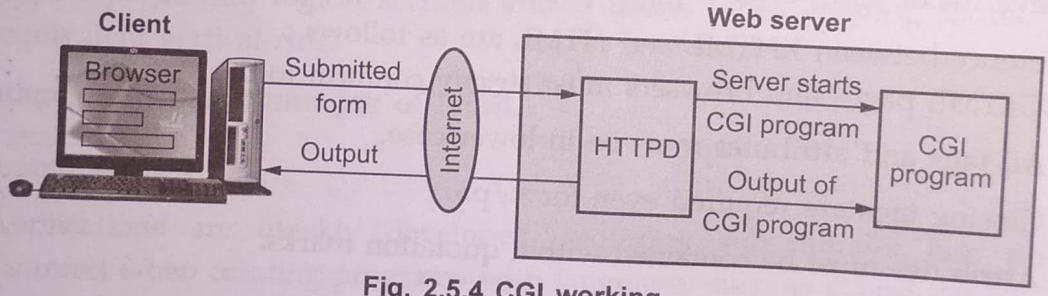


Fig. 2.5.4 CGI working

- The server uses the URL in the incoming request to determine which CGI program to run. CGI working is as follows : CGI program is part of a web server.
- The browser sends a request to the server. If the requested URL corresponds to a CGI program, the server starts the appropriate program and passes to the program a copy of the request.
- The server then sends the output from the CGI program back to the browser in the form of reply.
- From a browser's point of view, there is no difference between a URL that corresponds to a static document and one that corresponds to a CGI program. Requests for both static documents and CGI output have the same syntatic form.

2.5.5.2 Java Technology

- Sun Microsystems, Incorporated has developed a popular active document technology called Java, the technology can be used to create animated web pages, pages that interact with the user, or pages that use the screen in unexpected ways.

Scanned with CamScanner

- Java calls an active web page an *apple* ; the terminology is so widespread that most other vendors have either adopted it or chosen to use a minor variation.
- Java became popular for four reasons.
  1) The designers chose to make the Java language similar to a widely-used programming language, meaning that professional programmers could learn to write Java applets easily.
  2) No other active document technology was available.
  3) Because the Java system includes software to handle common tasks such as controlling the screen, a programmer can use predefined pieces to create a Java applet quickly.
  4) Java is so powerful that it provides more functionality than most other technologies. For example, Java can handle direct user interaction better than forms, can fetch a sequence of pages better than client-pull, can control multiple areas of the screen better than frames, and can manipulate a variety of data formats better than plugins. Thus, Java can substitute these by other technologies.
- Despite its many advantages over existing technologies, the strongest motivation for Java came from its ability to provide functionality that other technologies could not provide high quality animations. Because they use a computer's processing power to compute new images instead of trying to download them from a Web server, active document technologies like Java can change the display fast enough to present the illusion of smooth motion. Because none of the older Web technologies can provide the same functionality, many Web sites have been eager to use Java.

2.5.6 Browser Architecture

- Each browser usually consists of three parts.
  1. Controller 2. Client programs 3. Interpreters

Fig. 2.5.5 shows browser architecture.

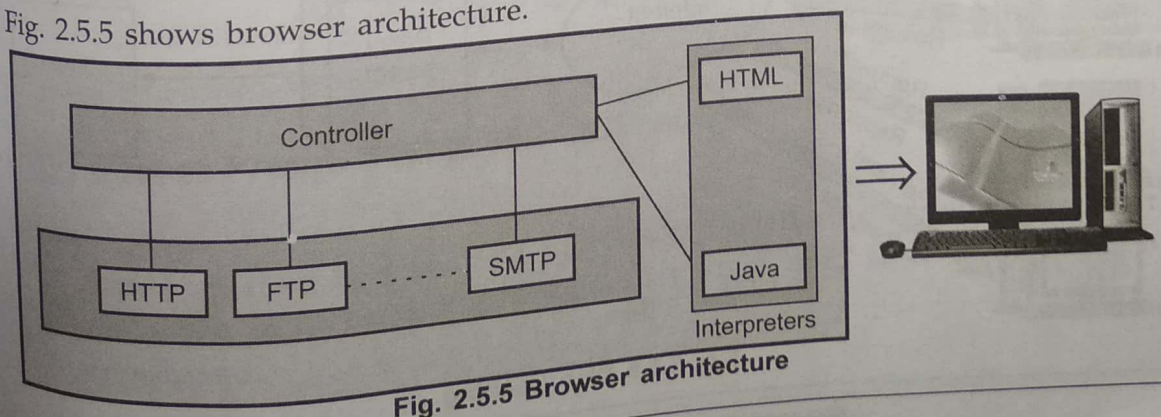


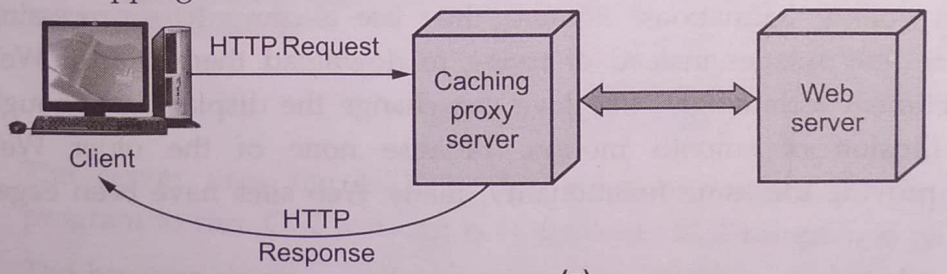
Fig. 2.5.5 Browser architecture

Scanned with CamScanner

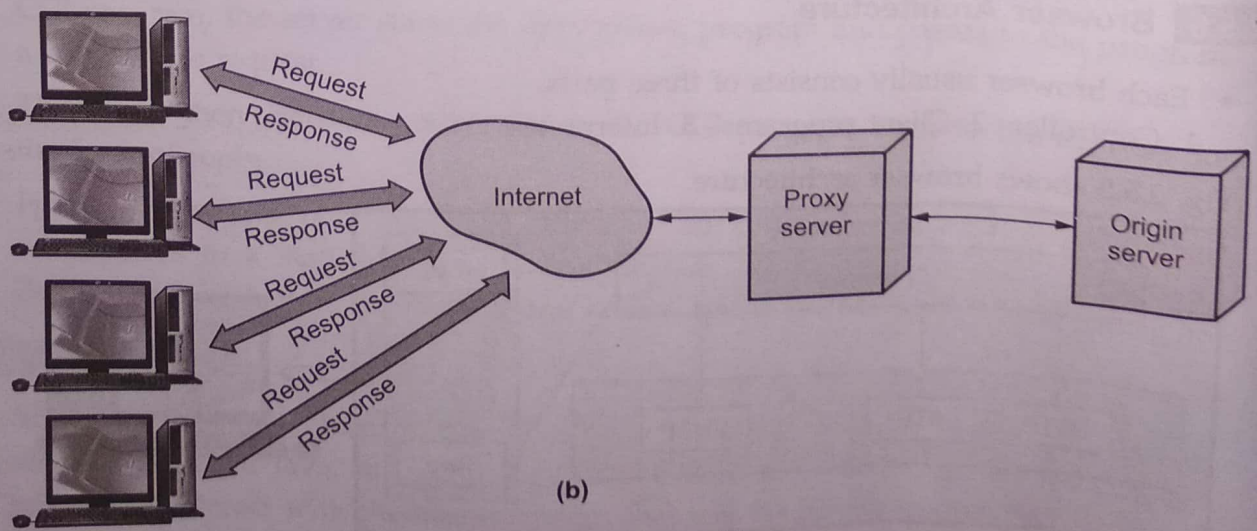
- The controller receives input from the keyboard or mouse and uses the client programs to access the document. After the document has been accessed, the controller uses one of the interpreters to display the document on the screen. Client program uses protocol such as HTTP, FTP or SMTP. The interpreter can be HTML or Java.

2.5.7 Caching in Web Browser

- Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address. DNA handles this with a mechanism called caching.
- When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
- If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem.
- To inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as unauthoritative.
- Caching speed up the resolution.
- Problem - if a server caches a mapping for a long time, it may send an outdated mapping to the client.



(a)



(b)

Fig. 2.5.6 Web Caching

Scanned with CamScanner

- Above problem is solved by two methods.
  1. Authoritative server always adds a piece of information to the mapping called time to live (TTL).
  2. DNS requires that each server keeps a TTL counter for each mapping it caches. The cache memory must be searched periodically and those mappings with an expired TTL must be purged.
- It satisfies the client request without involving origin server.
- User sets brower i.e. web accesses via cache.
- Brower sends all HTTP request to cache. If the object is in cache, it returns the object otherwise cache request the object from origin server.
- Fig. 2.5.6 shows the caching.

2.5.8 Uniform Resource Locators

- The Uniform Resource Locator (URL) is a standard for specifying any kind of information on the Internet.
- URL has three parts
  1. The protocol
  2. DNS name of the machine where the page is located.
  3. File name containing the page.
- The protocol is the client-server program used to retrieve the document.
- Host is the computer on which the information is located.
- The URL can optionally contain the port number of the server.
- File name gives where the information is located.

Review Questions

1. *Explain WWW and HTTP.* **GTU : Dec.-11, Marks 5**
2. *Explain the architectural overview of the world wide web.* **GTU : Winter-12, Marks 7**
3. *Give architectural overview of WWW.* **GTU : Winter-13, Marks 7**

2.6 Socket Programming with TCP and UDP

Socket

- Socket interface is a protocol independent interface to multiple transport layer primitives. In order to write applications which need to communicate with other applications.

Scanned with CamScanner

- Socket is an abstraction that is provided to an application programmer to send or receive data to another process.
- Data can be sent to or received from another process running on the same machine or a different machine.
- It is like an endpoint of a connection. It exists on either side of connection and identified by IP Address and Port number.
- Sockets works with UNIX I/O services just like files, pipes and FIFO.
- API stands for Application Programming Interface. It is an interface to use the network. Socket API defines interface between application and transport layer.
- The API defines function calls to create, close, read and write to/from a socket.

Advantages of using Socket Interface

- Syntax of the API functions is independent of the protocol being used. Ex:- TCP/IP and UNIX domain protocols can be used by applications using a common set of functions.
- Gives way to better portability of applications across protocol suites.
- Hides the finer details of the protocols from application programs thereby yielding faster and bug free application development
- Sockets are referenced through socket descriptors which can be passed directly to UNIX system I/O calls. File I/O and socket I/O are exactly similar from the programmer perspective.

Sockets versus File I/O

- Working with sockets is very similar to working with files. The socket ( ) and accept ( ) functions both return handles (file descriptor) and reads and writes to the sockets requires the use of these handles (file descriptors).
- In Linux, sockets and file descriptors also share the same file descriptor table. That is, if you open a file and it returns a file descriptor with value say 8, and then immediately open a socket, you will be given a file descriptor with value 9 to reference that socket.
- Even though sockets and files share the same file descriptor table, they are still very different. Sockets have addresses associated with them whereas files do not; notice that this distinguishes sockets form pipes, since pipes do not have addresses with which they associate.
- You cannot randomly access a socket like you can a file with lseek ( ). Sockets must be in the correct state to perform input or output.

Scanned with CamScanner

Socket Abstraction

- Socket is the basic abstraction for network communication in the socket API. Socket defines an endpoint of communication for a process.

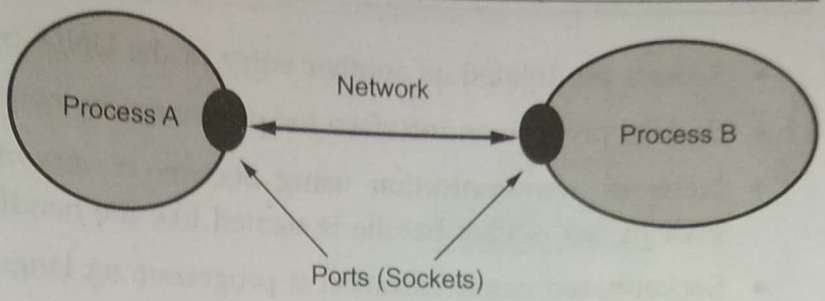


Fig 2.6.1 Socket and process

- Operating system maintains information about the socket and its connection. Fig. 2.6.1 shows the socket and process.

Socket Creation

int socket (int family, int type, int protocol);

Parameters:

1. family : AF_INET or PF_INET (These are the IP4 family)
2. type : SOCK_STREAM (for TCP) or SOCK_DGRAM (for UDP)
3. protocol : IPPROTO_TCP (for TCP) or IPPROTO_UDP (for UDP) or use 0

- If successful, socket ( ) returns a socket descriptor, which is an integer, and – 1 in the case of a failure.
- An example call:

```
if ((sd = socket(AF_INET, SOCK_DGRAM, 0) < 0)
{
        printf(socket() failed.);
        exit(1);
}
```

- Creating a socket is in some ways similar to opening a file. This function creates a file descriptor and returns it from the function call. You later use this file descriptor for reading, writing and using with other socket functions.
- Remember that the sockets API are generic. There must be a generic way to specify endpoint addresses. TCP/IP requires an IP address and port number for each endpoint address. Other protocol suites (families) may use other schemes.

2.6.1 TCP Socket

- In UNIX, whenever there is a need for IPC within the same machine, we use mechanism like signals or pipes. When we desire a communication between two applications possibly running on different machines, we need **Sockets**.

Scanned with CamScanner

- Sockets are treated as another entry in the UNIX open file table.
- Sockets provide an interface for programming networks at the transport layer.
- Network communication using Sockets is very much similar to performing file I/O. In fact, socket handle is treated like file handle.
- Socket-based communication is programming language independent.
- To the kernel, a socket is an endpoint of communication. To an application, a socket is a file descriptor that lets the application read/write from/to the network.
- A server (program) runs on a specific computer and has a socket that is bound to a specific port. The server waits and listens to the socket for a client to make a connection request.
- To review, there are five significant steps that a program which uses TCP must take to establish and complete a connection. The server side would follow these steps :
    1. Create a socket.
    2. Listen for incoming connections from clients.
    3. Accept the client connection.
    4. Send and receive information.
    5. Close the socket when finished, terminating the conversation.
- In the case of the client, these steps are followed:
    1. Create a socket.
    2. Specify the address and service port of the server program.
    3. Establish the connection with the server.
    4. Send and receive information.
    5. Close the socket when finished, terminating the conversation.
- Only steps two and three are different, depending on if it's a client or server application.
- Fig. 2.6.2 shows a timeline of the typical scenario that takes place between a TCP client and server.

(Refer Fig. 2.6.2 on next page.)

2.6.2 Socket Function

- To perform network I/O, process call the socket function specifying the type of communication protocol desired.
- The *socket ( )* system call creates an endpoint for communication and returns a descriptor.

Scanned with CamScanner

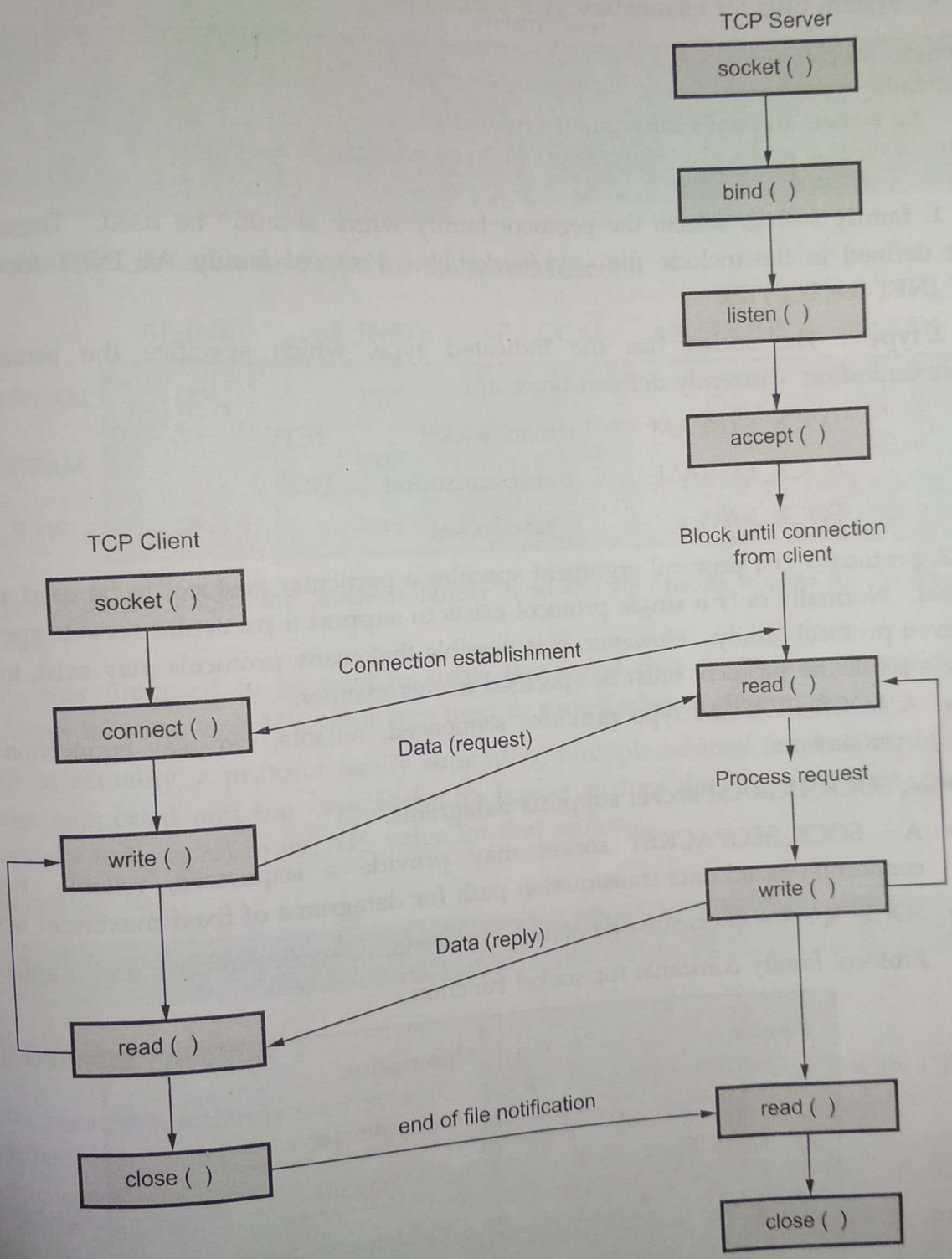


Fig. 2.6.2 Socket function for elementary TCP client server

Scanned with CamScanner

- *Socket( )* allocates resources needed for a communication endpoint - but it does not deal with endpoint addressing.
- System calls for elementary TCP sockets :

```
include <sys/types.h>
#include <sys/socket.h>
    int socket(int family,int type,int protocol );
returns on success: socket descriptor {a small nonnegative integer}
        on error:      -1
```

1. family : This selects the protocol family which should be used. These families are defined in the include file <sys/socket.h>. Protocol family AF_INET for Internet, PF_INET for TCP/IP).

2. type : The socket has the indicated type, which specifies the semantics of communication. Currently defined types are:

SOCK_STREAM	stream socket	TCP
SOCK_DGRAM	datagram socket	UDP
SOCK_RAW	raw socket	

3. protocol : The protocol argument specifies a particular protocol to be used with the socket. Normally only a single protocol exists to support a particular socket type within a given protocol family. However, it is possible that many protocols may exist, in which case a particular protocol must be specified in this manner.

- A SOCK_STREAM type provides sequenced, reliable, two-way connection based byte streams.
- A SOCK_DGRAM socket supports datagrams.
- A SOCK_SEQPACKET socket may provide a sequenced, reliable, two-way connection-based data transmission path for datagrams of fixed maximum length;
- SOCK_RAW sockets provide access to internal network protocols and interfaces.
- Protocol family constants for socket function.

Family	Description
AF_INET	IPv4 protocols
AF_INET6	Ipv6 Protocols
AF_ROUTE	Unix domain protocol
AF_ROUTE	Routing sockets
AF_KEY	Key Socket

Scanned with CamScanner

- Type of socket for socket function

Type	Description
SOCK_STREAM	Stream socket
SOCK_DGRAM	Datagram socket
SOCK_RAW	Raw socket

- Combination of family and type for the socket function.

	AF-INET	AF_INET6	AF_LOCAL	AF_ROUTE	AF_KEY
SOCK_STREAM	TCP	TCP	YES	–	–
SOCK_DGRAM	UDP	UDP	YES	–	–
SOCK_RAW	IPv4	IPv6	–	YES	YES

- The AF_prefix stands for "address family" and the PF_ prefix stands for "protocol family".
- PF_ was supposed to be used to create the socket that might support multiple address families, and AF_ value was used in socket address structures.
- But in actuality, a protocol family supporting multiple address families has never been supported and the <sys/socket.h> header defines the PF_value for a given protocol to be equal to the AF_ value for that protocol.
- Example:

```
if (( sd = socket (AF_INET, SOCK_STREAM, 0)) < 0)
        err_sys ("socket call error");
```

2.6.3 Connect Function

- The connect function is used by a TCP client to establish a connection with a TCP server.

```
#include <sys/socket.h>
int connect(int sockfd,const struct sockaddr *servaddr, socklen_t addrlen);
                                        Returns: 0 if OK, -1 on error
```

```
sockfd:   a socket descriptor returned by the socket function
*servaddr: a pointer to a socket address structure
addrlen:   the size of the socket address structure
```


Scanned with CamScanner

sockfd is a socket descriptor returned by the socket function. The second and third arguments are a pointer to a socket address structure and its size.

- connect( ) only returns when a connection is established or when an error occurs.
- The socket address structure must contain the IP address and port number of the server.
- The client does not have to call bind before calling connect: the kernel will choose both an ephemeral port and the source IP address if necessary.
- In the case of a TCP socket, the connect function initiates TCP's three-way handshake. The function returns only when the connection is established or an error occurs.
- There are several different error returns possible.
  1. If the client TCP receives no response to its SYN segment, ETIMEDOUT is returned. For example, in 4.4BSD sends one SYN when connect is called, another 6 seconds later, and another 24 seconds later. If no response is received after a total of 75 seconds, the error is returned.
  2. ECONNREFUSED : If the server's response to the client's SYN is an RST. RST indicates that no process is waiting for connections on the specified port. Considered hard error , i.e. error is returned to the client as soon as the RST is received. RST is a type of TCP segment that is sent by TCP when something is wrong. Three conditions that generates a RST:
- A SYN arrives for a port that has no listening server.
- A TCP wants to abort an existing connection.
- A TCP receives a segment for a connection that does not exist.
  3. EHOSTUNREACH or ENETUNREACH: If the SYN elicits an ICMP destination unreachable, the client kernel saves the message; i.e. considered soft error. It should not terminate connection attempt, keep trying. If no response after some fixed amount of time (say 75 secs), the saved message is returned to the process.
- Example :

```
if(connect(sd,(struct sockaddr*)&servaddr,sizeof(servaddr))!= 0)
    err_sys("connect call error");
```

2.6.4 Bind Function

- The servers bind to a particular port on their machines using the bind system call. This function has to be called only after a socket has been created and has to be passed the socket descriptor returned by the socket call. Again this binding on

Scanned with CamScanner

both the machines need not be in any particular order. Moreover the binding procedure on the client is entirely **optional**.

- The *bind* system call requires the address family, the port number and the IP address. The address family is known to be AF_INET, the IP address of the client is already known to the operating system. All that remains is the port number.
- The programmer can specify which port to bind to, but this is not necessary. The binding can be done on a random port as well and still everything would work fine. The way to make this happen is not to call *bind* at all. Alternatively *bind* can be called with the port number set to 0. This tells the operating system to assign a random port number to this socket. This way whenever the program tries to connect to a remote machine through this socket, the operating system binds this socket to a random local port. This procedure as mentioned above is not applicable to a server, which has to listen at a standard predetermined port.
- When a socket is created, it does not have any notion of end points addresses. An application calls bind to specify the local endpoint address for a socket. That is the bind function assigns a local port and address to a socket.

```
#include <sys/socket.h>
int bind (int sockfd, const strut sockaddr *myaddr,
                socklen_t addrlen)
```

- The second argument is a pointer to a protocol specific address and the third argument is the size of this address structure. Servers bind their well known port when they start.
- A process can bind a specific IP address to its socket. The IP address must belong to an interface host.

Uses of bind ( )

- Server would like to bind to a well known address (port number).
- Client can bind to a specific port.
- Client can ask the O.S. to assign any available port number.

2.6.5 Listen Function

- The listen function is called only by TCP server and it performs following functions.
  a. The listen function converts an unconnected socket into a passive socket, indicating that the kernel should accept incoming connection requests directed to this socket. In terms of TCP transmission diagram the call to listen moves the socket from the CLOSED state to the LISTEN state.

Scanned with CamScanner

b. The second argument to this function specifies the maximum number of connections that the kernel should queue for this socket.

```
#include <sys/socket.h>
int listen (int sockfd,int backlog);   returns 0 if OK -1on
                                                    error.
```

- This function is normally called after both the socket and bind functions and must be called before calling the accept function. The kernel maintains two queues and the backlog is the sum of these two queues. These are :
  a. An *Incomplete Connection Queue*, which contains an entry for each SYN that has arrived from a client for which the server is awaiting completion of the TCP three way handshakes. These sockets are in the SYN_RECD state.
  b. A *Completed Connection Queue* which contains an entry for each client with whom three handshakes has completed. These sockets are in the ESTABLISHED state.
- Fig. 2.6.3 .shows the two queues for a given listening socket.

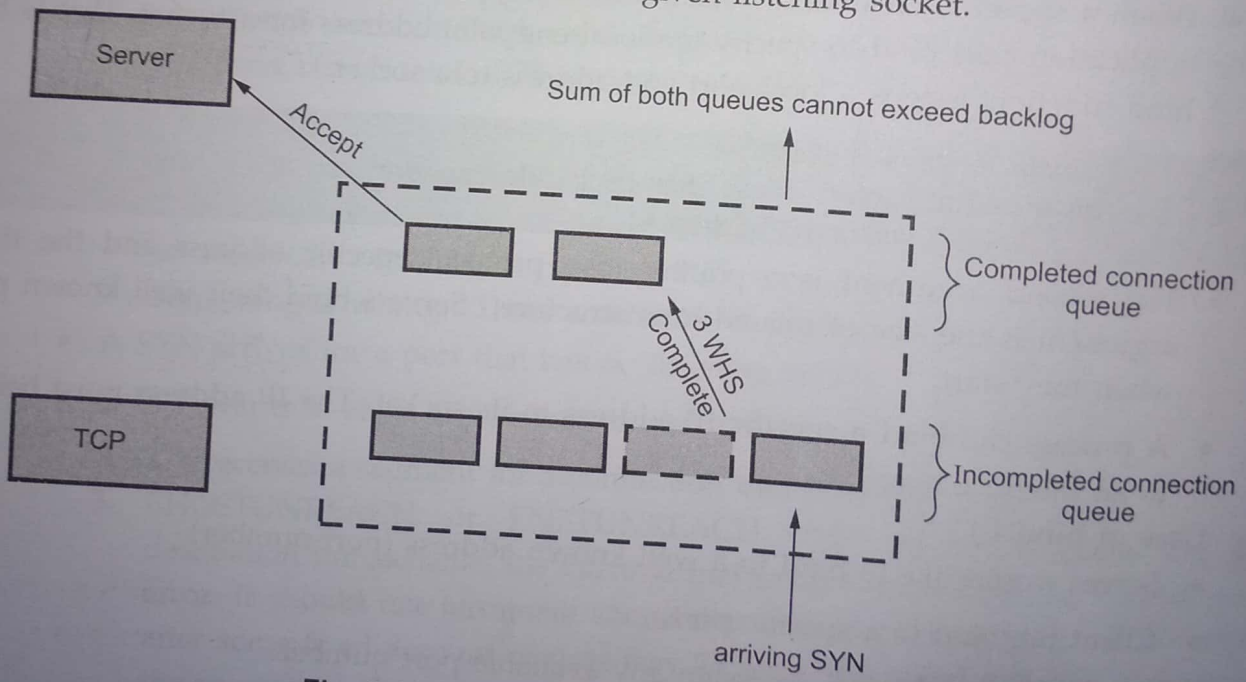


Fig. 2.6.3 Two queues maintained by TCP

- When a SYN arrives from a client, TCP creates a new entry on the incomplete queue and then responds with the second segment of the three way handshake. The server's SYN with an ACK of the clients SYN.
- This entry will remain on the incomplete queue until the third segment of the three-way handshake arrives or the entry times out. If the three-way hand shake completes normally, the entry moves from the incomplete queue to the completed queue.

Scanned with CamScanner

- When the process calls accept, the first entry on the completed queue is returned to the process or, if the queue is empty, the process is put to sleep until an entry is placed onto the completed queue. If the queues are full when a client arrives, TCP ignores the arriving SYN, it does not send an RST. This is because the condition is considered temporary and the client TCP will retransmit its SYN with the hope of finding room in the queue.
- Fig. 2.6.4 shows the TCP three way handshake and two queues for a listening socket.

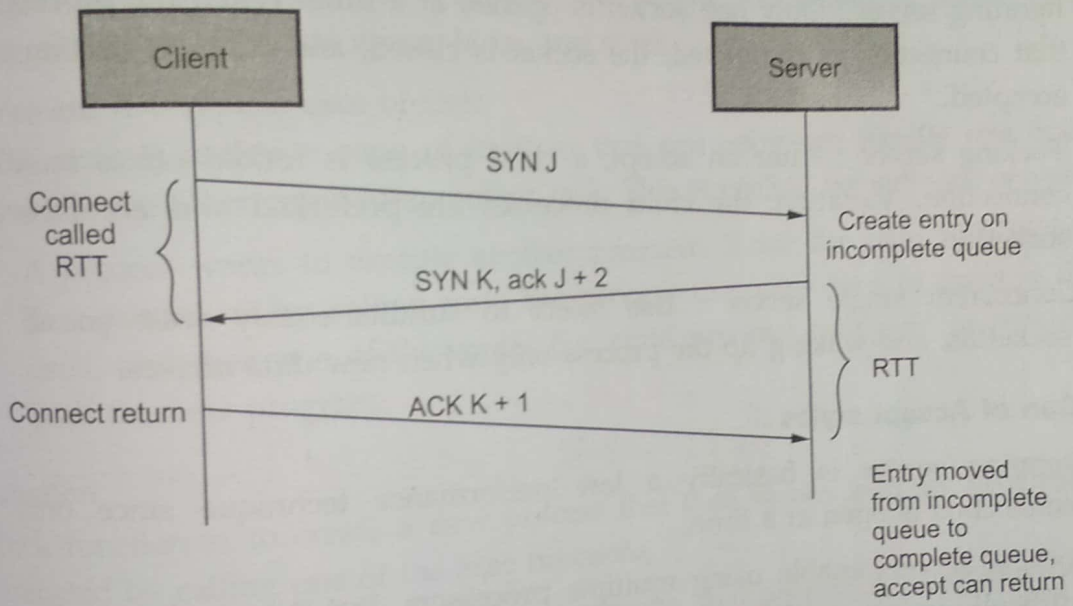


Fig. 2.6.4 TCP three way handshake and two queues for a listening socket

2.6.6 Accept Function

- *accept* is called by a TCP server to return the next completed connection from the completed connection queue. If the completed queue is empty, the process is put to sleep.

```
# include <sys/socket.h>
int accept ( sockfd, struct sockaddr * cliaddr, socklen_t *addrlen) ;
                return non negative descriptor if OK, -1 on error.
```

- The *cliaddr* and addrlen arguments are used to return the protocol address of the connected peer process. *addrlen* is a value-result argument before the call, we set the integer value pointed to by **addrlen* to the size of the socket address structure pointed to by *cliaddr* and on return this integer value contains the actual number of bytes stored by the Kernel in the socket address structure.
- If *accept* is successful, its return value is a brand new descriptor that was automatically created by the Kernel. This new descriptor refers to the TCP

Scanned with CamScanner

connection with the client. This function accepts extracts a connection on the buffer of pending connections in the system, creates a new socket with the same properties as skfd, and returns a new file descriptor for the socket.

- The *accept* call is a blocking system call. In case there are requests present in the system buffer, they will be returned and in case there aren't any, the call simply blocks until one arrives.

Accept styles

There are basically three styles of using accept:

1. Iterating server : Only one socket is opened at a time. When the processing on that connection is completed, the socket is closed, and next connection can be accepted.
2. Forking server : After an accept, a child process is forked off to handle the connection. Variation: the child processes are preforked and are passed the socketId.
3. Concurrent single server : Use select to simultaneously wait on all open socketIds, and waking up the process only when new data arrives.

Pro and Con of Accept styles

1. Iterating server is basically a low performance technique since only one connection is open at a time.
2. Forking servers enable using multiple processors. But they make sharing state difficult, unless performed with threads. Threads, however present a very fragile programming environment.
3. Concurrent single server : Reduces context switches relative to forking processes and complexity relative to threads. But it does not benefit from multiprocessors systems.

2.6.7 fork and exec Function

- The fork ( ) function is the only way in UNIX to create a new process. It is defined as follows:

```
#include <unist.h>
pid_t fork(void);
Returns: 0 in child, process ID of child in parent, 1 on error
```

- The new process created by fork is called the child process. This function is called once but returns twice.

Scanned with CamScanner

- The only difference in the returns is that the return value in the child is 0, whereas the return value in the parent is the process ID of the new child.
- The reason the child's process ID is returned to the parent is that a process can have more than one child, and there is no function that allows a process to obtain the process IDs of its children.
- The reason fork returns 0 to the child is that a process can have only a single parent, and the child can always call getppid to obtain the process ID of its parent.
- Both the child and the parent continue executing with the instruction that follows the call to fork. The child is a copy of the parent. For example, the child gets a copy of the parent's data space, heap, and stack.
- There are two typical uses of fork:
  1. A process makes a copy of itself so that one copy can handle one operation while the other copy does another task. This is typical for network servers.
  2. A process wants to execute another program. Since the only way to create a new process is by calling fork, the process first calls fork to make a copy of itself, and then one of the copies (i.e. child process) calls exec to replace itself with the new program.

exec function

- Fork function is to create a new process that then causes another program to be executed by calling one of the exec functions.
- When a process calls one of the exec functions, that process is completely replaced by the new program, and the new program starts executing at its main function.
- The process ID does not change across an exec, because a new process is not created; exec merely replaces the current process its text, data, heap, and stack segments with a brand new program from disk.
- There are six different exec functions. These six functions round out the UNIX system process control primitives.
- With fork, we can create new processes; and with the exec functions, we can initiate new programs. The exit function and the wait functions handle termination and waiting for termination.

```
#include <unistd.h>
int execl(const char *pathname,constchar *arg0,... /* (char*)0*/ );
int execv (const char *pathname, char *const argv[]);
int execle (const char *pathname, const char *arg0, ...
                /* (char *) 0, char *const envp[] */ );
int execve (const char *pathname,char *const argv[],char *const envp[]);
```


Scanned with CamScanner

```
int execlp(const char *filename,const char *arg0,.../*(char*)0*/);
int execvp (const char *filename, char *const argv[]);
```

All six return : 1 on error, no return on success

- The differences in the six exec functions are:
  1. Whether the program file to execute is specified by a filename or a pathname;
  2. Whether the arguments to the new program are listed one by one or referenced through an array of pointers;
  3. Whether the environment of the calling process is passed to the new program or whether a new environment is specified.
- These functions return to the caller only if an error occurs. Otherwise, control passes to the start of the new program, normally the main function.

2.6.8 Close Function

- The *close* ( ) function deallocate the file descriptor indicated by *fildes*. To deallocate means to make the file descriptor available for return by subsequent calls to *open* ( ) or other functions that allocate file descriptors. All outstanding record locks owned by the process on the file associated with the file descriptor shall be removed (i.e. unlocked).
- If *close*( ) is interrupted by a signal that is to be caught, it shall return -1 with *errno* set to EINTR and the state of *fildes* is unspecified. If an I/O error occurred while reading from or writing to the file system during *close*( ), it may return -1 with *errno* set to EIO; if this error is returned, the state of *fildes* is unspecified.
- When all file descriptors associated with a pipe or FIFO special file are closed, any data remaining in the pipe or FIFO shall be discarded. When all file descriptors associated with an open file description have been closed, the open file description shall be freed.
- If the link count of the file is 0, when all file descriptors associated with the file are closed, the space occupied by the file shall be freed and the file shall no longer be accessible.
- The normal UNIX close ( ) is also used to close a socket and terminate a TCP connection.

```
#include<unistd.h>
int close (int sockfd);
```

- The default action of close with a TCP socket is to mark the socket as closed and return to the process immediately. The socket descriptor is no longer usable by the process. i.e. it can not be used as an argument to read or write.

Scanned with CamScanner

getsockname ( ) and getpeername():

- These two functions return either the local protocol address associated with a socket or the foreign address associated with a socket.

```
#include <sys/socket.h>
int getsockname(int sockfd, struct sockaddr *localadddr,
                socklen_t  *addlen);
int getpeername(int sockfd, struct sockaddr *peeradddr,
                socklen_t  *addlen);
                    both return 0 if OK and –1 on error.
```

- The *getsockname( )* function retrieves the locally-bound name of the specified socket, stores this address in the **sockaddr** structure pointed to by the *address* argument, and stores the length of this address in the object pointed to by the *address_len* argument.
- If the actual length of the address is greater than the length of the supplied **sockaddr** structure, the stored address will be truncated. If the socket has not been bound to a local name, the value stored in the object pointed to by *address* is unspecified.
- Upon successful completion, 0 is returned, the *address* argument points to the address of the socket, and the *address_len* argument points to the length of the address. Otherwise, – 1 is returned and *errno* is set to indicate the error.
- The *getpeername( )* function retrieves the peer address of the specified socket, stores this address in the **sockaddr** structure pointed to by the *address* argument, and stores the length of this address in the object pointed to by the *address_len* argument.
- If the actual length of the address is greater than the length of the supplied **sockaddr** structure, the stored address will be truncated. If the protocol permits connections by unbound clients, and the peer is not bound, then the value stored in the object pointed to by *address* is unspecified.
- Upon successful completion, 0 is returned. Otherwise, – 1 is returned and *errno* is set to indicate the error.
- These functions are required for the following reasons.
  a. After connect successfully returns a TCP client that does not call **bind()**, **getsocketname()** returns the local IP address and local port number assigned to the connection by the Kernel
  b. After calling bind with a port number of 0, **getsockname()** returns the local port number that was assigned
  c. When the server is exceed by the process that calls **accept()**, the only way the server can obtain the identity of the client is to call **getpeername()**.

Scanned with CamScanner

2.6.9 UDP Socket Programming

-not necessary to establish connection

- There are some instances when it makes to use UDP instead of TCP. Some popular applications built around UDP are DNS, NFS and SNMP.
- Fig. 2.6.5 shows the interaction between a UDP client and server.
- Initially client does not establish a connection with the server. Instead, the client just sends a datagram to the server using the send to function which requires the address of the destination as a parameter.
- Similarly, the server does not accept a connection from a client. Instead, the server just calls the recvfrom function, which waits until data arrives from some client.

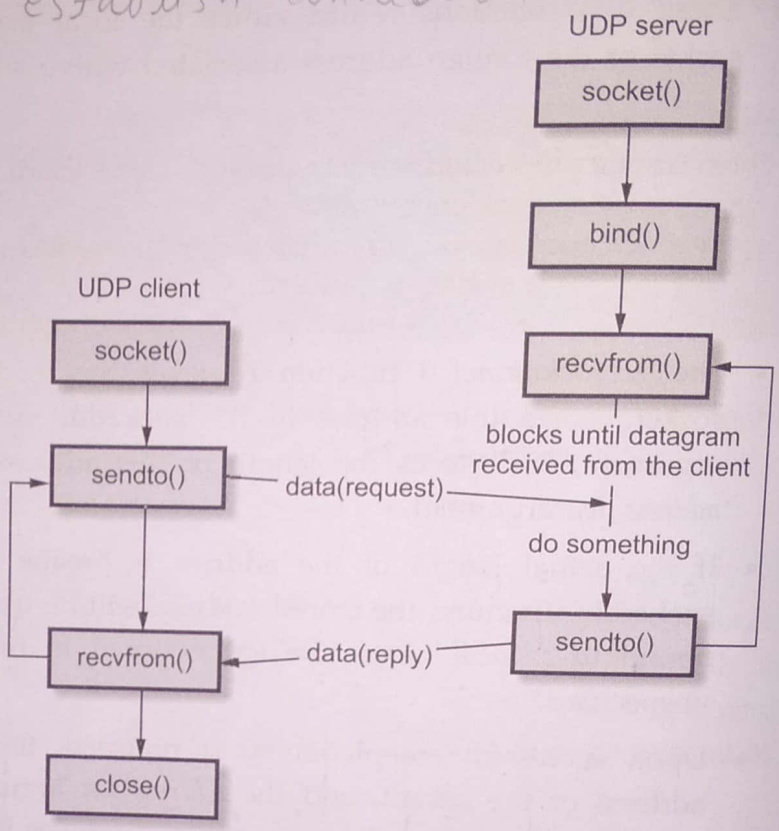


Fig. 2.6.5 UDP client-server

- *recvfrom* returns the protocol address of the client, along with the datagram, so the server can send a response to the client.

Steps on the client side are as follows:

1. Create a socket using the socket( ) function;
2. Send and receive data by means of the recvfrom ( ) and sendto( ) functions.

Steps on the server side are as follows:

1. Create a socket with the socket( ) function;
2. Bind the socket to an address using the bind( ) function;
3. Send and receive data by means of recvfrom( ) and sendto ( ).

2.6.9.1 The recvfrom ( ) Function

- This function is similar to the read ( ) function, but three additional arguments are required. The recvfrom ( ) function is defined as follows:

Scanned with CamScanner

```
#include <sys/socket.h>
ssize_t recvfrom (int sockfd, void* buff, size_t nbytes, int flags,
        struct sockaddr* from,
    socklen_t *addrlen);
```

- The first three arguments sockfd, buff, and nbytes, are identical to the first three arguments of read and write.
- sockfd is the socket descriptor, buff is the pointer to read into, and nbytes is number of bytes to read.
- The recvfrom function fills in the socket address structure pointed to with the protocol address of who sent the datagram. The number of bytes stored in the socket address structure is returned in the integer pointed by addrlen.
- The function returns the number of bytes read if it succeeds, -1 on error.

2.6.9.2 Sendto ( ) Function

- Sendto ( ) is similar to the send ( ) function, but three additional arguments are required. The sendto( ) function is defined as follows:

```
#include <sys/socket.h>
ssize_t sendto (int sockfd, const void *buff, size_t nbytes,
            int flags, const struct sockaddr *to,
            socklen_t addrlen);
```

- The first three arguments sockfd, buff, and nbytes, are identical to the first three arguments of recv.
- sockfd is the socket descriptor, buff is the pointer to write from, and nbytes is number of bytes to write.
- The sendto argument is a socket address structure containing the protocol address (e.g., IP address and port number) of where the data is sent. addrlen specified the size of this socket.

The function returns the number of bytes written if it succeeds, -1 on error.

❑❑❑

Scanned with CamScanner

3 Transport Layer

Syllabus

Introduction and transport layer services, Multiplexing and demultiplexing, Connection less transport (UDP), Principles of reliable data transfer, Connection oriented transport (TCP), Congestion control.

Contents

3.1 Introduction of Transport Layer . . . . . **Winter-14, Dec.-10, Summer-13,** . . . . . Marks 7

3.2 The Transport Layer Services . . . . . **June-11, Winter-12,** . . . . . Marks 7

3.3 Elements of Transport Protocols . . . . . **June-11, May-12,** . . . . . Marks 7

3.4 User Datagram Protocol . . . . . **Summer-13, Winter-13,** . . . Marks 7

3.5 Principle of Reliable Data Transfer . . . . . **Dec.-10,11, June-11, May-12, Winter-12, Summer-13,14,** Marks 7

3.6 Connection Oriented Transport (TCP) . . . . . **Dec.-11, Summer-13,14, Winter-12,14,** . . . . . Marks 7

3.7 Adaptive Retransmission

3.8 Congestion Control . . . . . **Dec.-10, Winter-12,14,** . . . Marks 14

3.9 Congestion Avoidance . . . . . **Winter-12,14, May-12,** . . . . Marks 7

3.10 Quality of Service . . . . . **Dec.-10, June-11, May-12, Winter-13,14,** . . . . . Marks 4

3.11 Performance . . . . . **Dec.-10, June-11,** . . . . . Marks 3

Scanned with CamScanner

3.1 Introduction of Transport Layer

GTU : Winter-14, Dec.-10, Summer-13

- A transport layer protocol provides for logical communication between application processes running on different hosts. The logical communication means that the communicating application processes are not physically connected to each other from the applications' viewpoint. Application processes use the logical communication provided by the transport layer to send messages to each other.
- Transport layer protocols are implemented in the end systems but not in network routers. Network routers only act on the network-layer fields. All transport layer protocols provide an application multiplexing/demultiplexing service.
- The transport service is said to perform "peer to peer" communication, with the remote transport entity. The data communicated by the transport layer is encapsulated in a transport layer PDU and sent in a network layer SDU. The network layer nodes transfer the transport PDU intact, without decoding or modifying the content of the PDU.
- The transport layer is the fourth layer in the OSI layered architecture. The transport layer is responsible for reliable data delivery. The upper-layer protocols depends heavily on the transport layer protocol. A high level of error recovery is also provided in this layer. This layer ensures, that packets are delivered error free, in sequence and with no losses or duplications.
- Transport layer functions :

1. This layer breaks messages into packets.
2. It performs error recovery if the lower layer are not adequately error free.
3. Function of flow control if not done adequately at the network layer.
4. Functions of multiplexing and demultiplexing sessions together.
5. This layer can be responsible for setting up and releasing connections across the network.

- Data link layer is responsible for delivery of frames between two neighbouring nodes over a link. So this iscalled *mode-to-node* delivery.
- Network layer is responsible for *host-to-host* delivery i.e. delivery of datagrams between two hosts. Fig. 3.1.1 shows data delivery.
- Transport layer is responsible for *process-to-process* delivery i.e. the delivery of a packet, part of a message one process to another.
- Client server paradigm is used for process to process communication. A process on the local machine (hot) called a *client* needs services from a process usually on the remote host called *server*.

Scanned with CamScanner

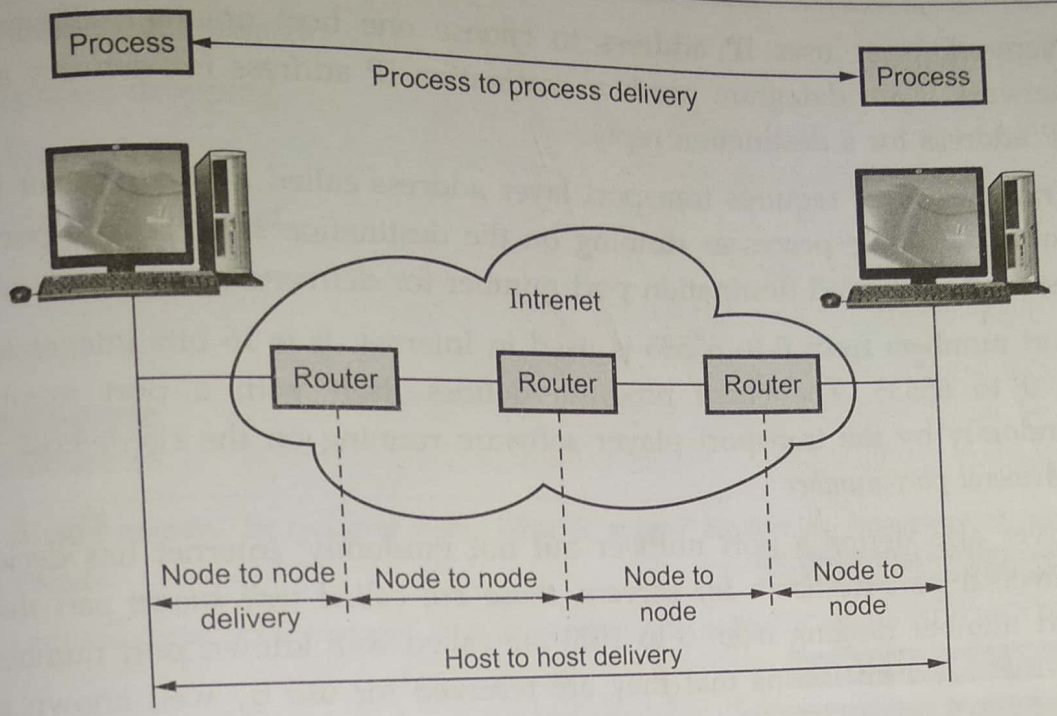


Fig. 3.1.1 Data delivery

- Nowadays, operating system support multiuser and multiprogramming environments. A remote computer can run several server programs at the same time. Following parameters are used for communication
  1) Local host
  2) Local process
  3) Remote host
  4) Remote process
- The services that a transport protocol can provide are often constrained by the service model of the underlying network layer protocol. If network layer protocol cannot provide delay or bandwidth guarantee for 4 PDU's sent between hosts, then the transport layer protocol cannot provide delay or bandwidth guarantees for messages sent between processes.
- Some of the services offered by the transport protocol even when the underlying network protocol does not offer the corresponding service at the network layer.
- A transport protocol can offer reliable data transfer service to an application even when the underlying network protocol is unreliable, even when the network protocol loses, garbles and duplicate packets.

Addressing method

- We need a address wherever to deliver something to one specific destination among may. All the layer uses different addressing methods.
- Data link layer uses MAC address to choose one node among several nodes, if the

Scanned with CamScanner

- Network layer uses IP address to choose one host among millions of host. In network layer, datagram needs a destination IP address for delivery and a source IP address for a destination reply.
- Transport layer requires transport layer address called a port number for selecting among multiple processes running on the destination host. Source port number is used for reply and destination port number for delivery.
- Port numbers from 0 to 65535 is used in Internet. It is 16 bits integer so the range is 0 to 65535. The client program defines itself with a port number, chosen randomly by the transport player software running on the client host. This is the *ephemeral port number*.
- Server also define a port number but not randomly. Internet has decided to use universal port numbers for servers, these are called *well known port numbers*. The port number ranging from 0 to 1023 are called well known port numbers and are restricted, which means that they are reserved for use by well known application protocols such as HTTP.

IANA Ranges

- The Internet Assigned Number Authority (IANA) has divided the port number into three ranges. They are

a) Well known ports

b) Registered ports

c) Dynamic ports

Sr. No.	Port Type	Range	Remark
1.	Well known port	0 to 1023	Assigned and controlled by IANA.
2.	Registered port	1024 to 49151	Not assigned and controlled by IANA. Only registered to prevent duplication
3.	Dynamic	49152 to 65535	Neither controlled nor registered. Used by any process. These are ephemeral ports.

- Properties of transport protocol

1. Ordered delivery of message.
2. Guarantees message delivery.
3. Protocol support large messages.
4. Each host support multiple application processes.
5. Protocol suport data transfer speed between sender and receiver

Scanned with CamScanner

Limitations of computer network

1. Message/packet dropping
2. Message reordering
3. Message size will be fixed.
4. Duplicate message delivery.
5. Long delay.

University Questions

1. *Enlist services provided by transport layer. What is socket ? Explain its importance at transport layer protocols.* **GTU : Winter-14, Marks 7**
2. *List the various duties of the transport layer and explain each in brief. Compare UDP and TCP.* **GTU : Dec.-10, Marks 7**
3. *List difference between transport layer and network layer.* **GTU : Summer-13, Marks 7**

3.2 The Transport Layer Services

GTU : June-11, Winter-12

- The transport protocol should provide to higher-level protocols. The transport entity that provides services to transport service users, which might be an application process. The hardware and software within the transport layer that does the work is called the transport entity. It can be in the operating system Kernel, in a separate user process or on the network interface card. The relationship of the network, transport and application layers in Fig. 3.2.1.

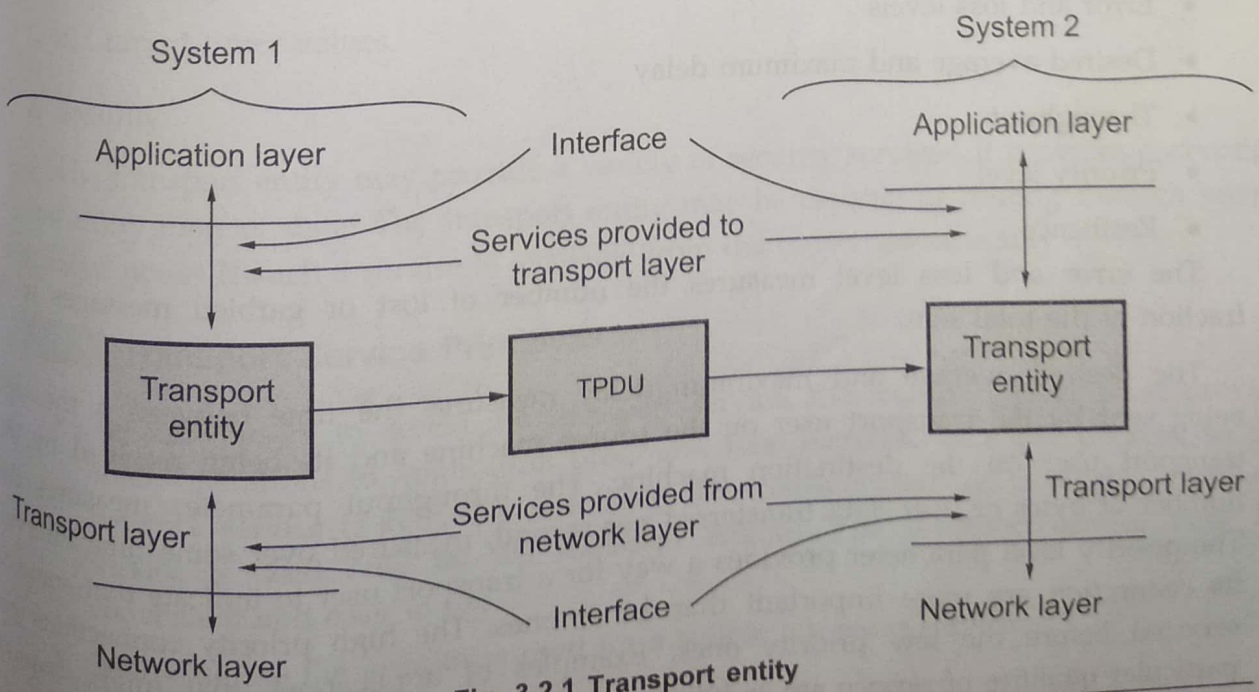


Fig. 3.2.1 Transport entity

Scanned with CamScanner

- The following categories of service are useful for describing the transport service.
  1. Type of service
  2. Quality of service
  3. Data transfer
  4. User interface
  5. Connection management
  6. Expedited delivery
  7. Status reporting
  8. Security

1. Type of service

It provides two types of services connection-oriented and connectionless or datagram service. A connection-oriented service provides for the establishment, maintenance and termination of a logical connection between transport service users. The connection-oriented service generally implies that the service is reliable. The connection-oriented service allows for connection-related features such as flow control, error control and sequenced delivery.

2. Quality of service

The transport protocol entity should allow the transport service user to specify the quality of transmission service to be provided. Following are the transport layer quality of service parameters

- Error and loss levels
- Desired average and maximum delay
- Throughput
- Priority level
- Resilience

The error and loss level measures the number of lost or garbled messages as a fraction of the total sent.

The desired average and maximum delay measures the time between a message being sent by the transport user on the source machine and its being received by the transport user on the destination machine. The throughput parameter measures the number of bytes of user data transferred per second, measured over some time interval. The priority level parameter provides a way for a transport user to indicate that some of its connection are more important than other ones. The high priority connections get serviced before the low priority ones. Examples of applications that might request particular qualities of service are as follows :

Scanned with CamScanner

a) A FTP might require high throughput.

b) A transaction protocol may require low delay.

c) An E-mail protocol may require multiple priority levels.

3. Data transfer

It transfers data between two transport entities. Both user data and control data must be transferred. Full duplex service must be provided. Half-duplex and simplex modes may also be offered.

4. User interface

There is not clear mechanism of the user interface to the transport protocol should be standardized.

5. Connection management

If connection-oriented service is provided, the transport entity is responsible for establishing and terminating connections. Symmetric connection procedure should be provided, which allows either TS user to initiate connection establishment.

6. Status reporting

It gives the following information.

a) Addresses

b) Performance characteristics of a connection

c) Class of protocol in use

d) Current timer values.

7. Security

The transport entity may provide a variety of security services. It provides encryption and decryption of data. The transport entity may be capable of routing through secure links or nodes if such a service is available from the transmission facility.

3.2.1 Transport Service Primitives

- To allow users to access the transport service, the transport layer must provide some operations to application programs. Real networks can lose packets, so the network service is generally unreliable. The transport service is reliable. Network service is used only by the transport entities. The transport service must be convenient and easy to use.
- Following are the primitives used for a simple transport service.

Scanned with CamScanner

Primitive	Packet sent	Meaning
LISTEN	(None)	Block until some process tries to connect.
CONNECT	CONNECTION REQ	Actively attempt to establish a connection.
SEND	DATA	Send information.
RECEIVE	(None)	Block until a DATA packet arrives.
DISCONNECT	DISCONNECTION REQ	This side wants to release the connection.

- At the transport layer, even a simple unidirectional data exchange is more complicated than at the network layer. Every data packet sent will also be acknowledged. These acknowledgement are managed by the transport entities using the network layer protocol and are not visible to transport user.
- Transport Protocol Data Unit (TPDU) for messages sent from transport entity to transport entity. TPDU are contained in the packets. Packets are contained in frames. When a frame arrives, the data link layer processes the frame header and passes the contents of the frame payload field up to the network entity. The network entity processes the packet header and passes the contents of the packet payload upto the transport entity.
- Fig. 3.2.2 shows the nesting of TPDU with packets.

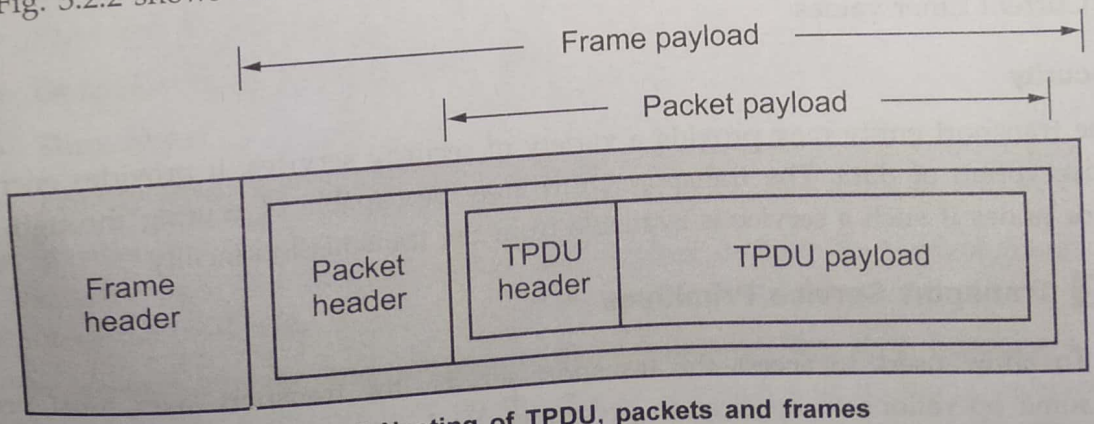


Fig. 3.2.2 Nesting of TPDU, packets and frames

- The client's CONNECT call causes a CONNECTION REQUEST TPDU to be sent to the server. When it arrives, the transport entity checks to see that the server is blocked on a LISTEN. It then unblocks the server and sends a CONNECTION

Scanned with CamScanner

ACCEPTED TPDU back to the client. When this TPDU arrives, the client is unblocked and the connection is established.

- Data can be exchanged using the SEND and RECEIVE primitives. When the TPDU arrives, the receiver is unblocked. It can then process the TPDU and send a reply.
- When a connection is no longer needed, it must be released to free up table space within the two transport entities. Disconnection has two types : Asymmetric and symmetric.
- Fig. 3.2.3 shows the state diagram for a simple connection management scheme.

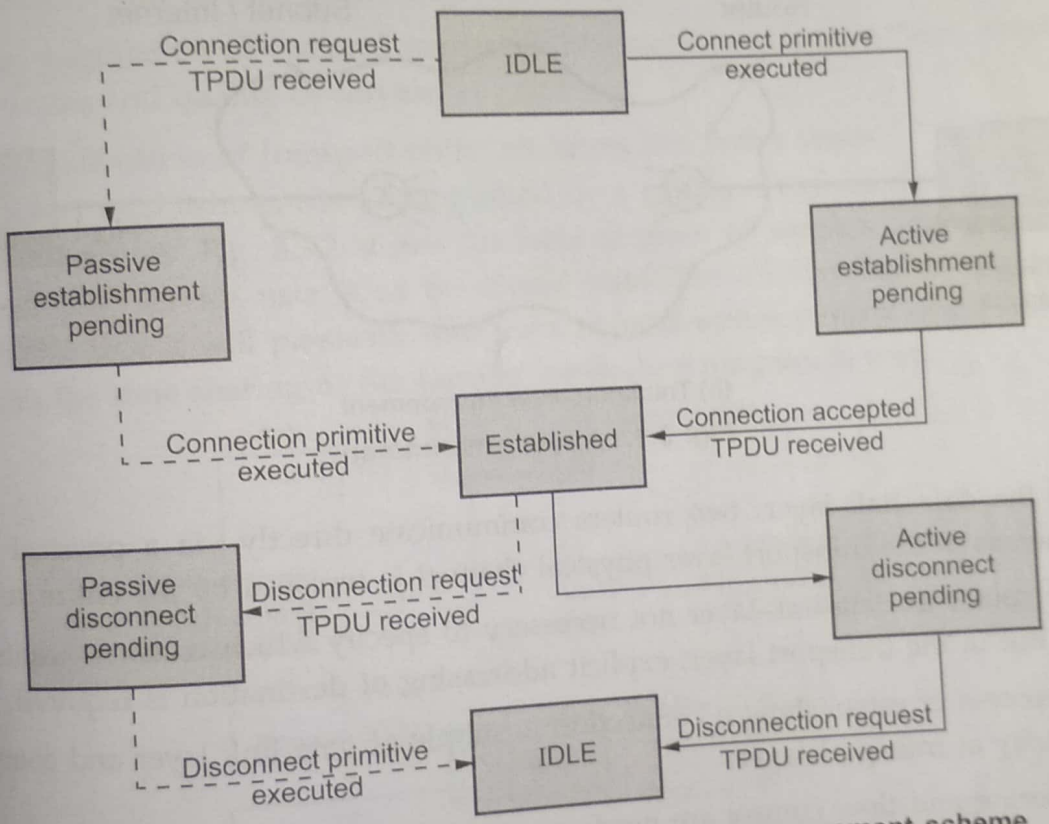


Fig. 3.2.3 State diagram for a simple connection management scheme

University Questions

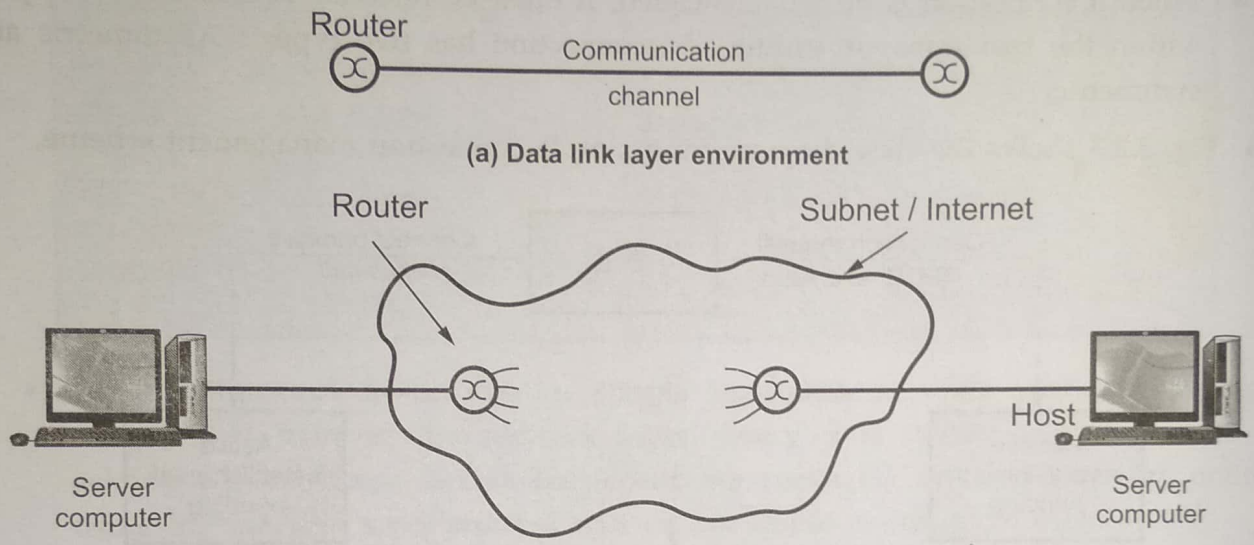
1. *Generate the CRC code for 1101011011. The divisor is 10011. Append CRC with data and check at receiver side whether any error exists or not.* **GTU : June-11, Marks 4**
2. *Write about elements of transport layer 1) Connection establishment 2) Connection release.* **GTU : June-11, Marks 7**
3. *Explain the basic five service primitives of the transport layer protocol.* **GTU : Winter-12, Marks 7**

Scanned with CamScanner

3.3 Elements of Transport Protocols

GTU : June-11, May-12

- The transport service is implemented by a transport protocol used between the two transport entities. Fig. 3.3.1 shows the environment of DLL and transport layer.



(a) Data link layer environment

(b) Transport layer environment

Fig. 3.3.1 Layer environment

- At the data link layer, two routers communicate directly via a physical channel, whereas at the transport layer physical channel is replaced by the entire subnet.
- For router in data link layer not necessary to specify which router it wants to talk to. But in the transport layer, explicit addressing of destination is required.
- A process of establishing a connection is simple at data link layer and complicated initially at transport layer.
- Buffering and flow control are needed in both layers.

3.3.1 Addressing

- When a user of a given transport entity wishes to establish a connection with a user of some other transport entity. The source user needs to be specified by all the informations, user identification, transport entity identification, station address and network number. Typically, the user address is specified as station or port. The port variable represents a particular TS user at the specified station, in OSI this is called a **Transport Service Access Point (TSAP).**
- Each station having only one transport entity, so a transport entity identification is not needed. The address should include a designation of the type of transport protocol e.g. TCP, UDP. In the case of single network, station identifies an

Scanned with CamScanner

attached network device. In internet, station is a global internet address. Transport layer simply passes the station portion of the address down to the network service. Port is included in a transport header to be used at the destination by the destination transport protocol.

3.3.2 Connection Establishment

For supporting connection-oriented service, the establishment and releasing of connection is required. The connection establishment serves three main purposes.

1. It allows each end to assure that the other exists.
2. It allows negotiation of optional parameter like maximum segment size, maximum window size and quality of service.
3. It triggers allocation of transport entity resources like buffer space.

- Connection establishment is accomplished by a simple set of user commands and control segments. Fig. 3.3.2 shows the state diagram of simple connection. Firstly the transport service user is in the closed state. The TS (Transport Service) user can signal that it will passively wait for a request with a passive open command. For this the time sharing or file transfer application program is used.

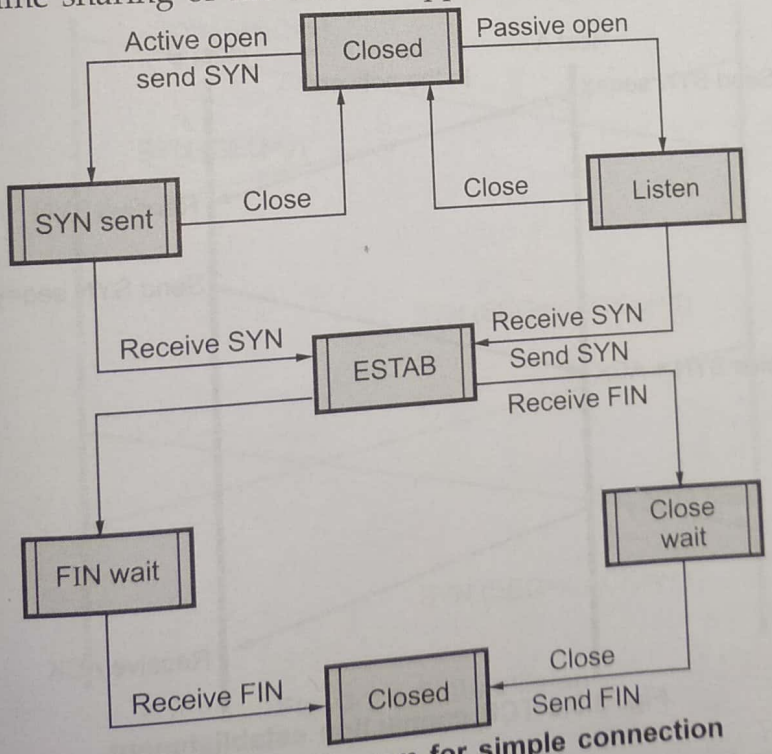


Fig. 3.3.2 State diagram for simple connection

- After a passive open command is issued, the transport entity creates a connection object of some sort that is in the listen state. After issuing an active open command the TS user may open a connection from the closed state. The transport entity then establishes the connection with a destinated user, which then triggers

Scanned with CamScanner

the transport entity to send an SYN segment. The SYN segment is carried to the receiving transport entity and interpreted as a request for connection to a particular port.

- The connection will establish, if the destination transport entity is in the listen state.
- Signal the TS user that a connection is open.
- Send an SYN as confirmation to the remote transport entity.
- Put the connection object in an ESTAB state.
- TCP provides a connection-oriented service over packet switched networks. Connection-oriented implies that there is a virtual connection between two end points.
- There are three phases in any virtual connection. These are the connection establishment, data transfer and connection termination phases.
- In order for two hosts to communicate using TCP they must first establish a connection by exchanging messages in what is known as the three-way handshake.
- Fig. 3.3.3 shows the process of the **three-way handshake.**

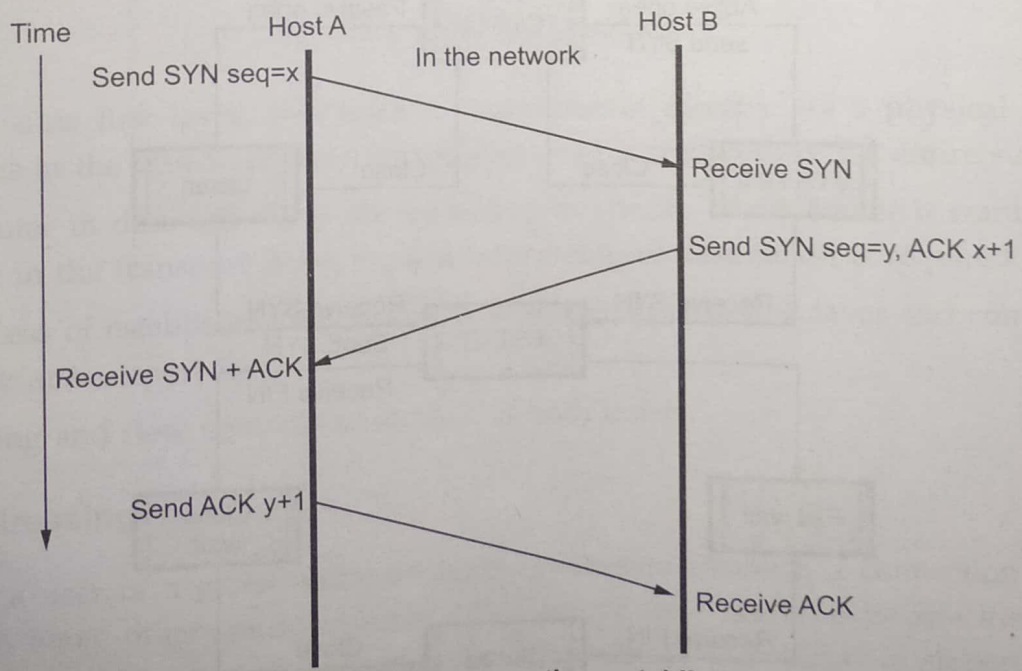


Fig. 3.3.3 TCP connection establishment

- From Fig. 3.3.3 it can be seen that there are three TCP segments exchanged between two hosts, host A and host B. To start, host A initiates the connection by sending a TCP segment with the SYN control bit set and an Initial Sequence Number (ISN) here it is represent as the variable x in the sequence number field.

Scanned with CamScanner

- At some moment later in time, host B receives this SYN segment, process it and responds with a TCP segment of its own. The response from host B contains the SYN control bit set and its own ISN represented as a variable y. Host B also sets the ACK control bit to indicate the next expected byte from host A should contain data starting with sequence number x+1.
- When host A receives host B's ISN and ACK, it finishes the connection establishment phase by sending a final acknowledgement segment to host B. In this case, host A sets the ACK control bit and indicates the next expected byte from host B by placing acknowledgement number y+1 in the acknowledgement field.
- Once ISNs have been exchanged, communicating applications can transmit data between each other.
- If the two hosts simultaneously attempt to establish a connection between the same sockets, the sequence of events is shown in the Fig. 3.3.4 Only one connection is established, not two because connections are identified by their end points.

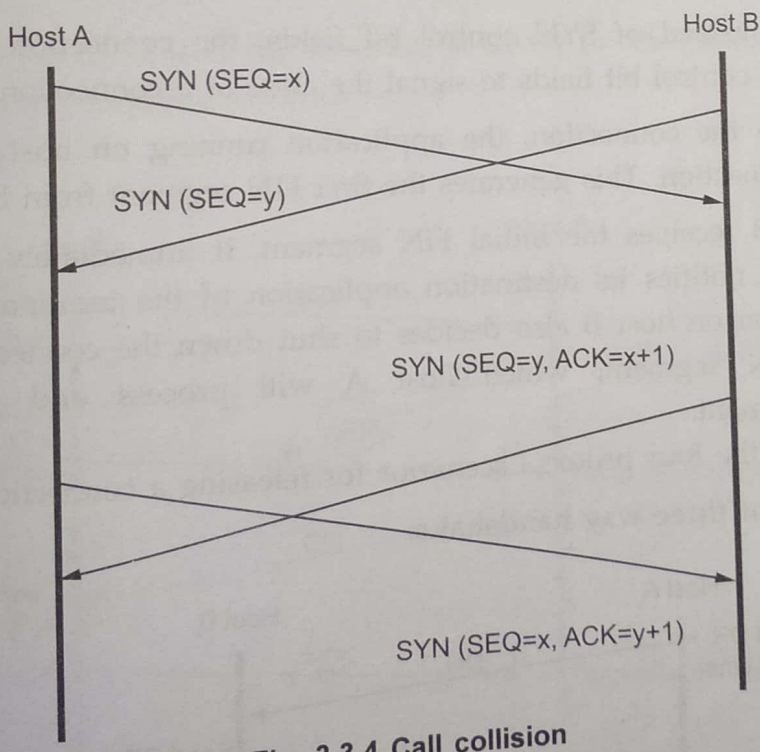


Fig. 3.3.4 Call collision

3.3.3 Connection Termination

- In order for a connection to be released, four segments are required to completely close a connection. Four segments are necessary due to the fact that TCP is a full-duplex protocol, meaning that each end must shut down independently. The connection termination phase is shown in Fig. 3.3.5.

Scanned with CamScanner

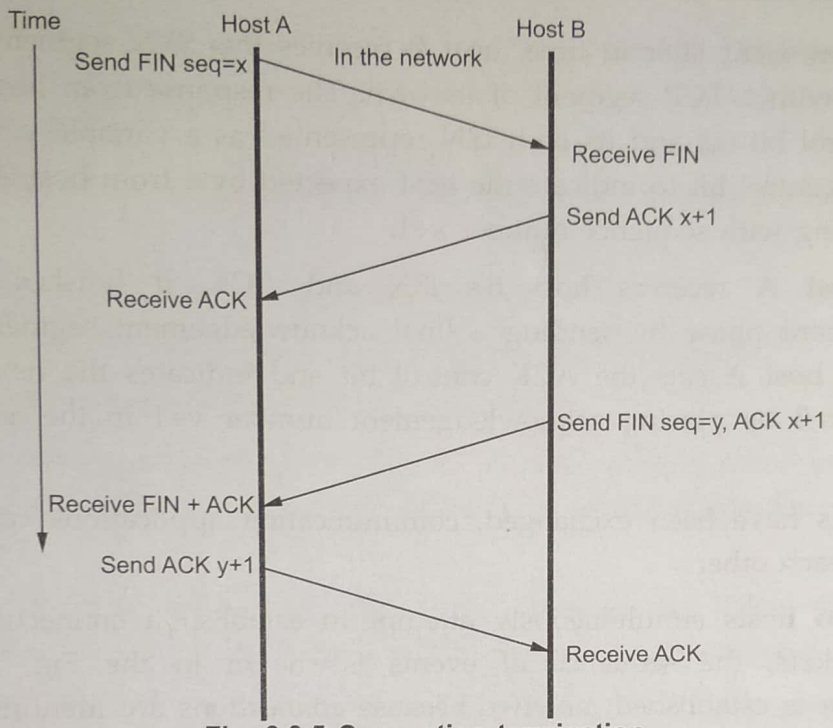


Fig. 3.3.5 Connection termination

- Notice that instead of SYN control bit fields, the connection termination phase uses the FIN control bit fields to signal the close of a connection.
- To terminate the connection, the application running on host A signals TCP to close the connection. This generates the first FIN segment from host A to host B.
- When host B receives the initial FIN segment, it immediately acknowledges the segment and notifies its destination application of the termination request. Once the application on host B also decides to shut down the connection, it then sends its own FIN segment, which host A will process and respond with an acknowledgement.

Fig. 3.3.6 shows the four protocol scenarios for releasing a connection.

a) Normal case of three way handshake.

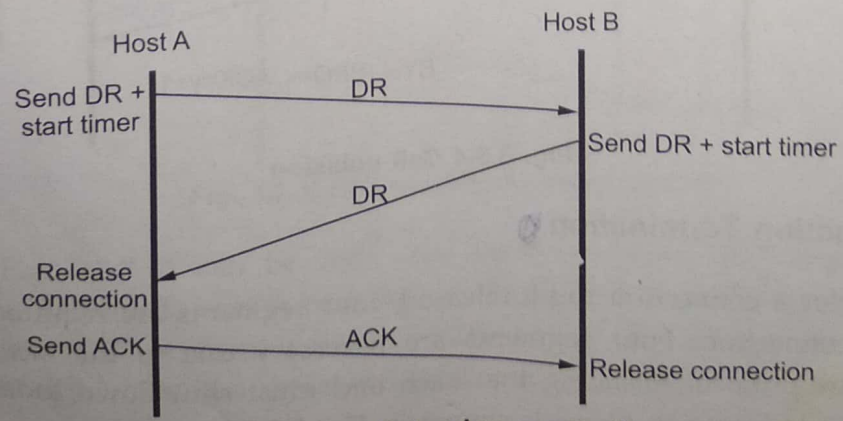


Fig. 3.3.6 (a) Releasing connection

Scanned with CamScanner

One of the user sends a DR (DISCONNECTION REQUEST) TPDU to initiate the connection release. When it arrives, the receipent sends back and DR TPDU and start a timer. When this DR arrives, the original sender sends back an ACK TPDU and releases the connection. Finally, when the ACK TPDU arrives, the receiver also releases the connection.

b) Final ACK lost

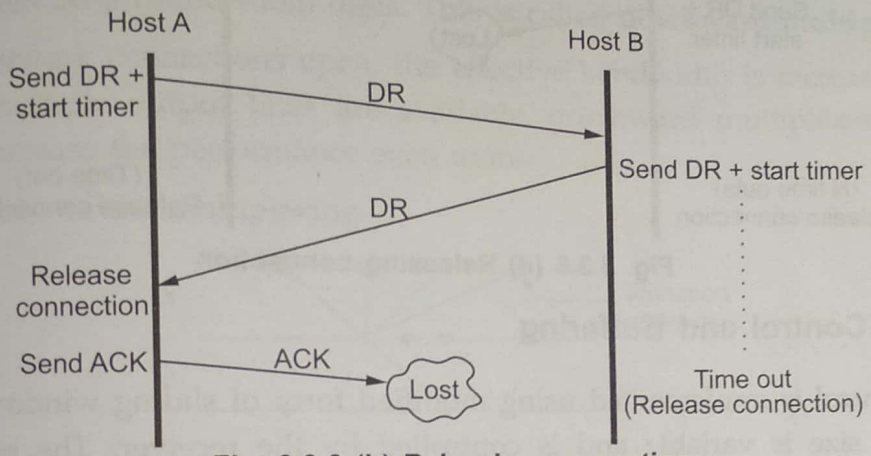


Fig. 3.3.6 (b) Releasing connection

If the final ACK TPDU is lost, the situation is saved by the timer. When the timer expires, the connection is released anyway.

c) Response lost

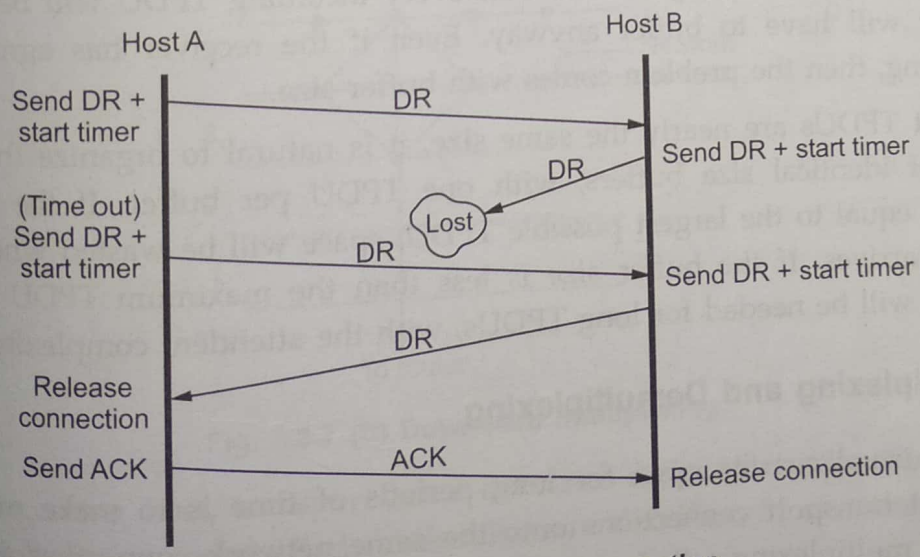


Fig. 3.3.6 (c) Releasing connection

If the second DR lost then the user initiating the disconnection will not receive the expected response, will time out. The second time no TPDU are lost and all TPDUs are delivered correctly and on time.

Scanned with CamScanner

d) Response lost and subsequent DR lost

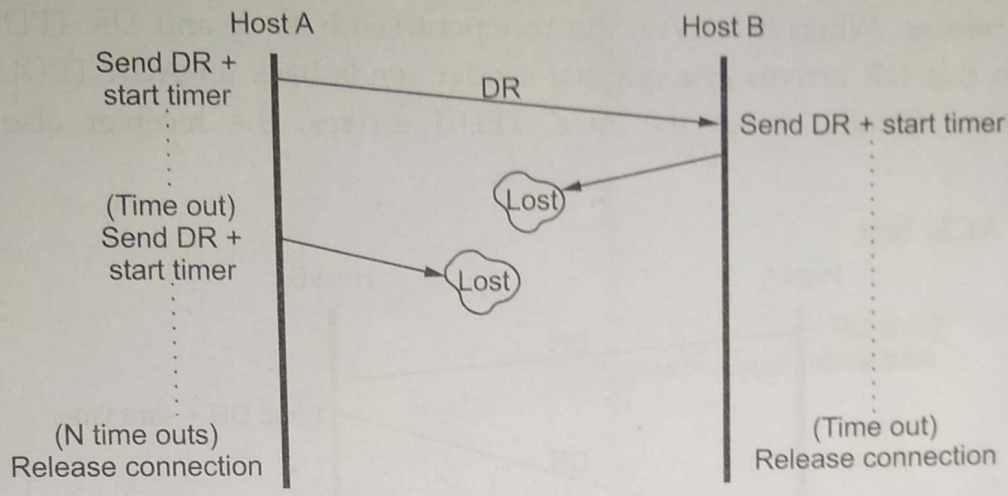


Fig. 3.3.6 (d) Releasing connection

3.3.4 Flow Control and Buffering

- Flow control is implemented using modified form of sliding window protocol. The window size is variable and is controlled by the receiver. The receiver sends a credit allocation to the sender. The credit allocation indicates how many TDPU the receiver is ready to receive if the network service is unreliable, the sender must buffer all TDPUs sent. With reliable network service, if the sender knows that the receiver always has buffer space, it need not retain copies of the TPDU it sends.
- If the receiver cannot guarantee that every incoming TPDU will be accepted, the sender will have to buffer anyway. Even if the receiver has agreed to do the buffering, then the problem comes with buffer size.
- If most TPDUs are nearly the same size, it is natural to organize the buffers as a pool of identical size buffers, with one TPDU per buffer. If the buffer size is chosen equal to the largest possible TPDU, space will be wasted whenever a short TPDU arrives. If the buffer size is less than the maximum TPDU size, multiple buffers will be needed for long TPDUs, with the attendent complexity.

3.3.5 Multiplexing and Demultiplexing

- Many virtual circuits open for long periods of time is to make multiplexing of different transport connections onto the same network connection attractive. This form of multiplexing called **upward multiplexing.**
- Four distinct transport connections all use the same network connection to the remote host. The transport layer forms the group of transport connection according to their destination and map each group onto the minimum number of network connections.

Scanned with CamScanner

- Many transport connections are mapped onto one network connection, the performance will be poor, because users will have to wait their turn to send one message. If too few transport connections are mapped onto one network connection, the service will be expensive.
- The transport layer opens multiple network connections and distributes the traffic among them on a round-robin basis. This is called **downward multiplexing.**
- With k network connections open, the effective bandwidth is increased by a factor of k. If multiple output lines are available, downward multiplexing can also be used to increase the performance even more.
- Fig. 3.3.7 shows the multiplexing.

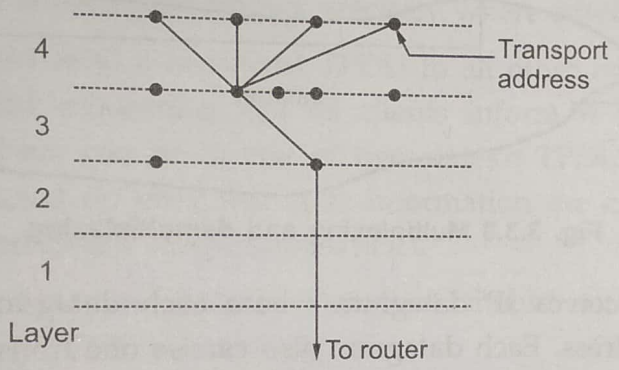


Fig. 3.3.7 (a) Upward multiplexing

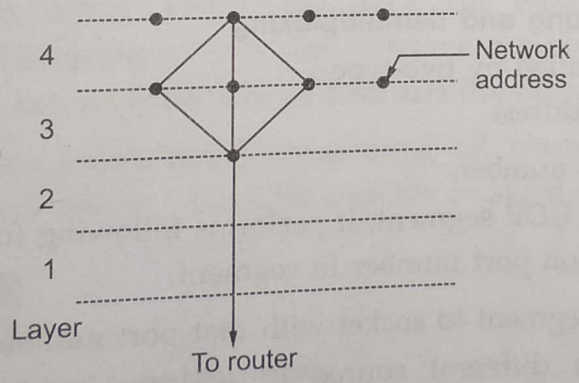


Fig. 3.3.7 (b) Downward multiplexing

- In the receiving host, transport layer does not deliver data directly to a process but it delivers to socket. Each socket has an identifier. More than one socket is available at any given time.
- Usually there are multiple application processes running on one host.
- **Multiplexing** means the dividing flows of data from the application into one or more short packets. **Demultiplexing** means by allocating each communication flow a unique identifier.
- Fig. 3.3.8 shows transport layer multiplexing and demultiplexing.

Scanned with CamScanner

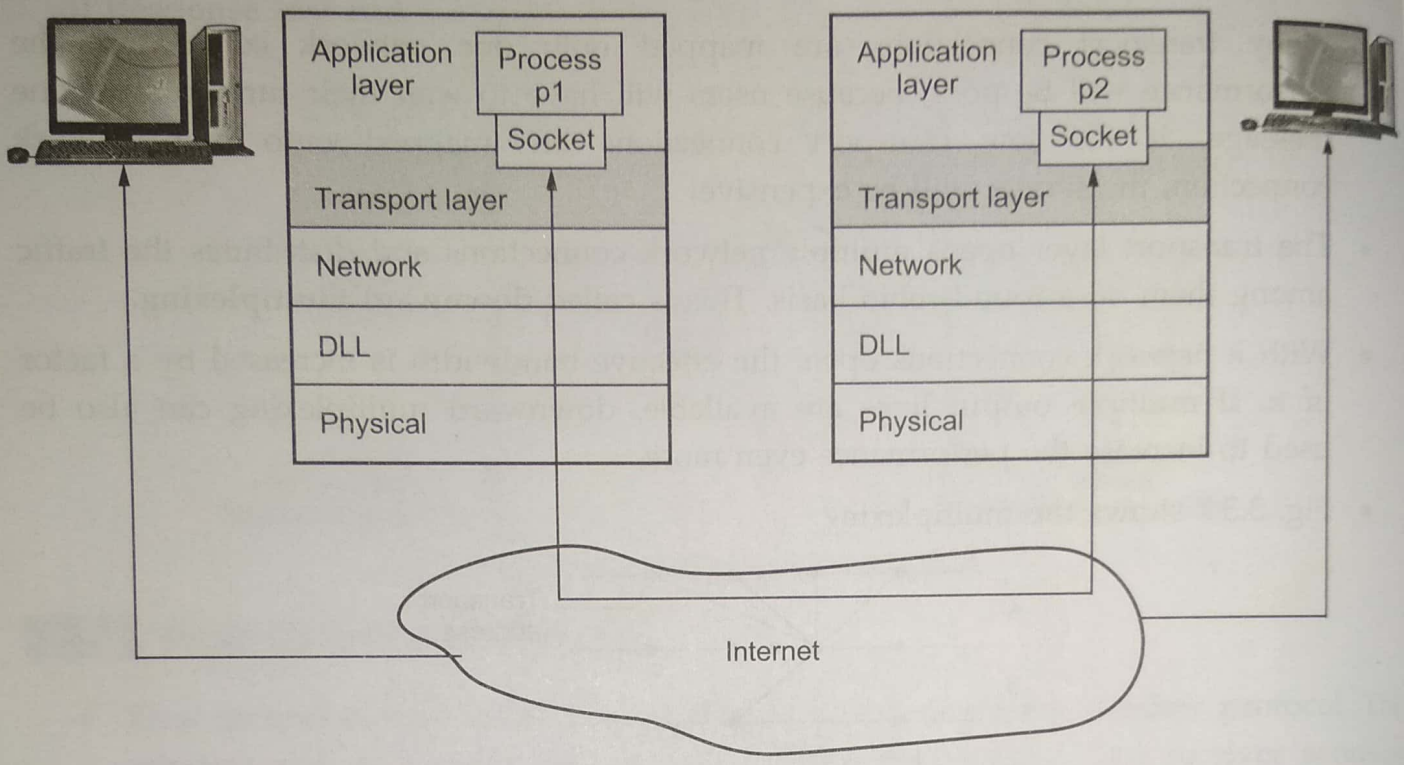


Fig. 3.3.8 Multiplexing and demultiplexing

- Receiving host receives IP datagram where each datagram contains source and destination IP address. Each datagram also carries one transport segment.
- Host uses IP addresses and port numbers to direct segment to appropriate socket.

Connectionless multiplexing and demultiplexing

- UDP socket is identified by two-type :
  1. Destination IP address
  2. Destination port number.
- When host receives UDP segment, it performs following functions :
  1. Checks destination port number in segment.
  2. It directs UDP segment to socket with that port number.
- IP datagrams with different source IP address and/or source port numbers directed to same socket.

Connection-Oriented multiplexing and demultiplexing

- TCP socket identified by four-tuple :
  1) Source IP address
  2) Source port number
  3) Destination IP address
  4) Destination port number.
- Receiving host uses all four values to direct segment to appropriate socket. Server host may support many simultaneous TCP sockets.

Scanned with CamScanner

- Web servers have different sockets for each connecting client.

3.3.6 Crash Recovery

- If the host computer (server) and routers are subject to crashes, the recovery from these crashes makes some problem. If the network layer provides connection-oriented service, then the lost of data or virtual circuit is handled by establishing a new connection. Then retransmit the TPDUs, which has not received. If the host server crashes, then the client must quickly reboot.
- When the server crash while receiving data from client, the outstanding TPDU is lost. To recover the data, when the server comes back up, its tables are reinitialized, so it no longer knows precisely where it was.
- The server might send a broadcast TPDU to all other host, announcing that it had just crashed and requesting that its clients inform it for the status of all open connections. Client can be in one of two states : TPDU outstanding or no TPDU outstanding. Based on only this state information the client must decide whether or not to retransmit the most recent TPDU.
- To avoid the duplicate of data the client should retransmit only if it has an unacknowledged TPDU outstanding.
- There are many situation for crash recovery. If the server send acknowledge and crash the server before writing the data. The writing (saving data) data and sending acknowledgement, both are different process. So the server and client programmed in any way, the lost of data and duplicate of data may occurs.
- If both sides are programmed considering all situation, upto some limit it is possible to recover the lost data and possible to avoid retransmitting.

University Questions

1. *Explain crash recovery of transport protocol.* **GTU : June-11, Marks 4**
2. *Explain connection establishment and connection release in transport protocols.* **GTU : May-12, Marks 7**
3. *Write about flow control and buffering mechanism in transport protocols.* **GTU : May-12, Marks 7**

3.4 User Datagram Protocol **GTU : Summer-13, Winter-13**

- UDP is a simple, datagram-oriented, transport layer protocol. This protocol is used in place of TCP. UDP is connectionless protocol provides no reliability or flow control mechanisms. It also has no error recovery procedures.

Scanned with CamScanner

- Several application layer protocols such as TFTP (Trivial File Transfer Protocol) and the RPC use UDP. UDP makes use of the port concept to direct the datagrams to the proper upper-layer applications. UDP serves as a simple application interface to the IP.
- Fig. 3.4.1 (a) shows the encapsulation of a UDP datagram as an IP datagram.

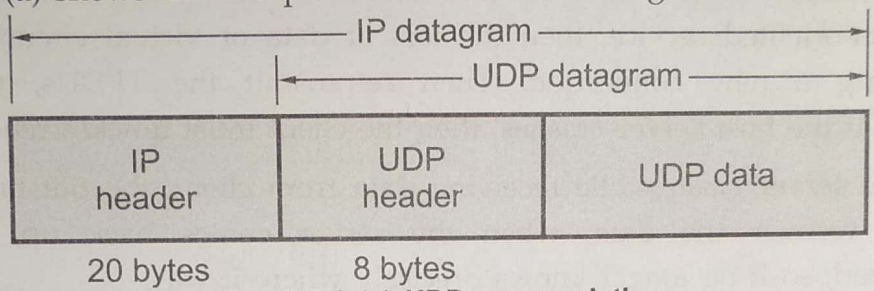


Fig. 3.4.1 (a) UDP encapsulation

- Fig. 3.4.1 (b) shows the format of the UDP header. The port number identify the sending process and the receiving process.

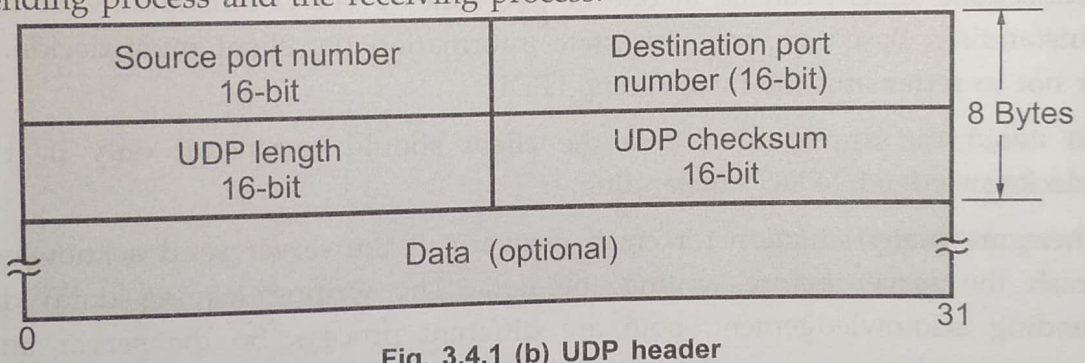


Fig. 3.4.1 (b) UDP header

- The UDP datagram contains a source port number and destination port number. Source port number identifies the port of the sending application process. The destination port number identifies the receiving process on the destination host machine.
- The UDP length field is the length of the UDP header and the UDP data in bytes. The minimum value for this field is 8 bytes.
- UDP checksum covers the UDP header and the UDP data. Both UDP and TCP include a 12 byte pseudo-header with the UDP datagram just for the checksum computation. This pseudo-header includes certain fields from the IP header. The purpose is to let UDP double check that the data has arrived at the correct destination.
- UDP checksum is end-to-end checksum. It is calculated by the sender, and then verified by receiver. It is designed to catch any modification of the UDP header or data anywhere between sender and receiver.
- UDP provides only error checking, it does not do anything to recover from error.

Scanned with CamScanner

UDP Checksum Example

Consider two 16-bit integers

```
  1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
+ 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
-----------------------------------
1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
```

The carry form MSB is added to result. Then

```
  1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
+                               1
-----------------------------------
1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 (Sum)
```

- Take 1's complement of the sum by converting all the 0's to 1's and converting all 1's to 0's.
- Checksum is the 1's compliment of the sum.

```
1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 (Sum)
0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 (1's complement)
```

Checksum

3.4.1 Port Numbers

- UDP uses port numbers as the addressing mechanism in the transport layer.
- Following is the list of well-known port number used by UDP.

Port No.	Protocol	Description
7	Echo	Echoes a received datagram back to the sender.
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns the quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol

Scanned with CamScanner

111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol

Applications of UDP

1. UDP is used for some route updating protocols such as RIP.
2. UDP is used for multicasting.
3. It is suitable for a process with internal flow and error control mechanisms.

3.4.2 Remote Procedure Calls (RPC)

- RPC is based on a client-server model that is an asymmetric type of communication. The ISO-OSI model and TCP/IP support the process of RPC. Client server model widely used in the local area networks in which dump terminals node access the server to obtain application software, files, etc.
- RPC is implemented in the client-server operation through a technique called **STUB**. Stub is a procedure such as read or write and can be defined for each server's clients. The read procedure becomes library procedure and client can obtain the services through a simple read statement. It is file to be read, number of byte to be read and a buffer to contain the result of the read. This then becomes a simple message transfer to the server, after which client waits for a reply from the server.
- Fig. 3.4.2 shows the remote operations with stub. If the server fails problem will occur in RPC. e.g. If client is sending request continuously to server and the request is not sent back to the client before the server fails.

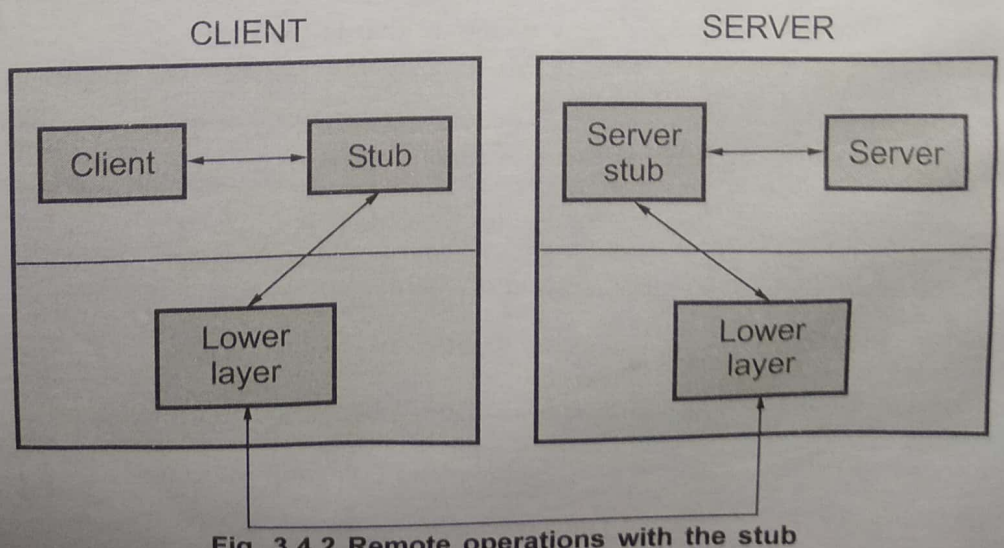


Fig. 3.4.2 Remote operations with the stub

Scanned with CamScanner

- If the client repeats the operation and resends the traffic (server is ON) then the reply is successfully executed and sent back. This type of operation is called idempotent.

OSI remote procedure operations :

- It is based on two operation sending request to server and receiving the result to the client. The result of the operation can report on various combinations of success or failure. ROSE also uses class number to describe the result of the operation, either for synchronous or asynchronous communication processes.

Class number	Definition
1. Synchronous	: Result or error
2. Asynchronous	: Result or error
3. Asynchronous	: Error only
4. Asynchronous	: Result only
5. Asynchronous	: Report nothing

Result = Report success Error = Failure

- For obtaining the services from remote server, Unix operating system programming syntax is used. Remote commands also allow the C programs to write data as input to the remote process and read from the local program what the remote process has output.

3.4.3 Real Time Transport Protocol

- Real time transport protocols run over user datagram protocol. Real Time Protocol (RTP) used in multimedia applications, videoconferencing, music-on-demand, video-on-demand. Audio, video and text are the content of the multimedia.
- Multimedia application also contains other types of data streams. All these data is stored into the RTP library in user space along with the application. This library then multiplexes the streams and encodes them in RTP packets, which then stuffs into a socket.
- Fig. 3.4.3 shows the RTP and packet nesting. Socket means communication end points.

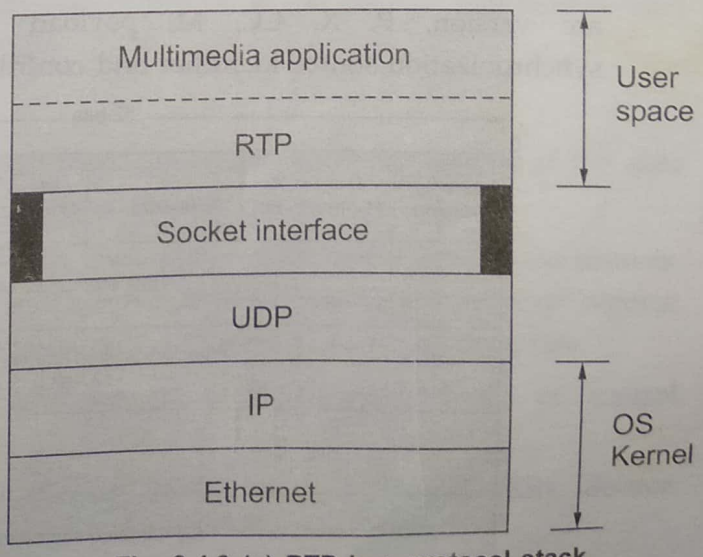


Fig. 3.4.3 (a) RTP in a protocol stack

Scanned with CamScanner

At the other sides of socket, UDP packets are generated and it is embedded in the IP packets. RTP uses user space and linked with the application program. So that it look like an application protocol. RTP is a generic and application independent protocol.

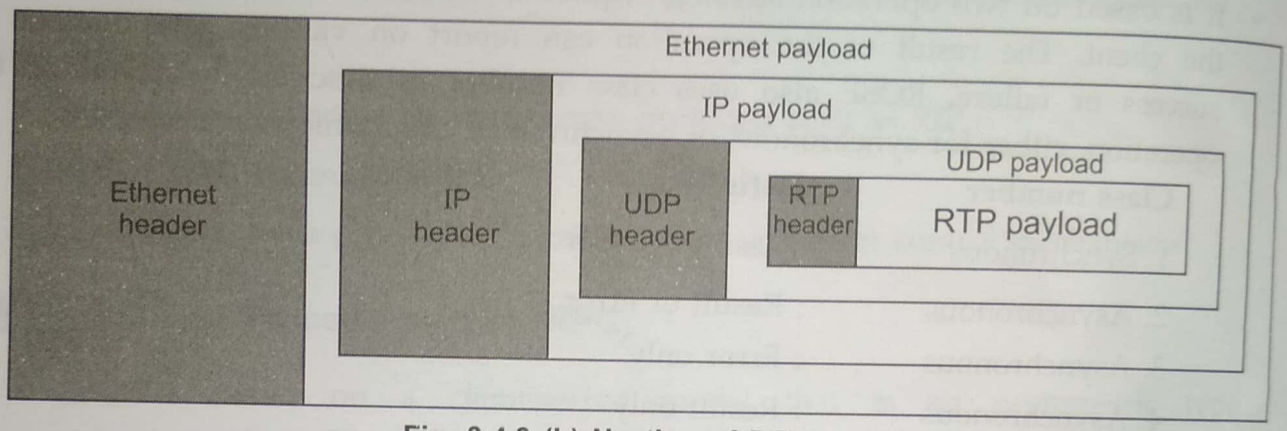


Fig. 3.4.3 (b) Nesting of RTP packet

- Fig. 3.4.3 shows RTP in a protocol stack and nesting.
  1) RTP handles real-time data streams onto a single stream of UDP packets.
  2) RTP has no flow control.
  3) RTP does not support error control.
  4) RTP has no acknowledgement and no mechanism to request retransmissions.
  5) Sequence number is given to each packet in an RTP stream.
  6) Sequence number is higher than its predecessor by one.
  7) Sequence numbering helps destination to determine the missing packets.
- Fig. 3.4.4 shows the RTP header. RTP header size is 32 bits. Fields in the headers are version, P, X, CC, M, payload type, sequence number, timestamp, synchronization source identifier and contributing source identifier.

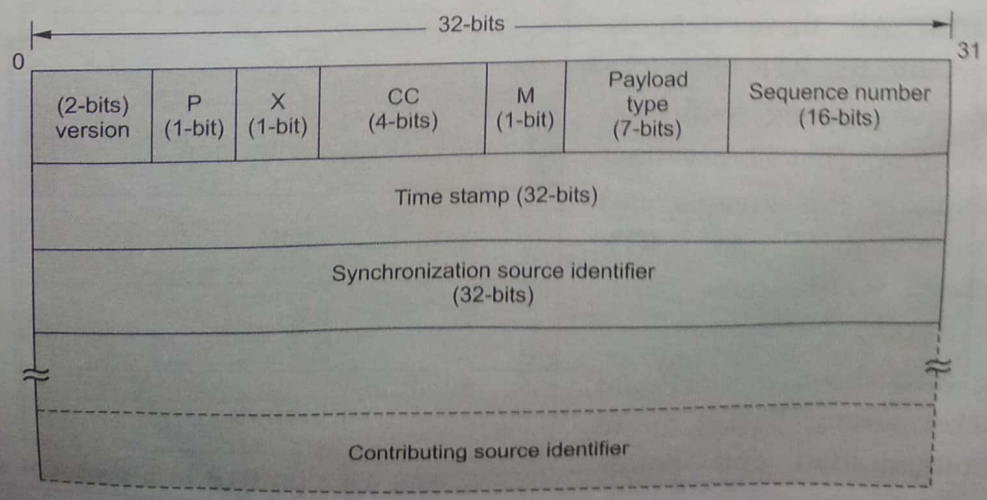


Fig. 3.4.4 RTP header

Scanned with CamScanner

1. **Version :** Size of version field is 2-bits. It indicates version number. The current version is 2.
2. **P bit :** Size is 1-bit. P bit indicates that the packet has been **padded** to a multiple of 4 bytes.
3. **X-bit :** Size is again 1-bit and it indicates that the **extension** header is present.
4. **CC field :** Size of CC field is 4-bits. CC field is used for indicating number of source present. The range is from 0 to 15.
5. **m bit :** Marker bit is of 1-bit size. This bit is used to indicate start of the frame. It may be video frame, start of a word in an audio channel.
6. **Payload type :** Size of the payload type field is 7-bits. This field is used for indicating encoding algorithm has been used. It determines its interpretation by the application.
7. **Sequence number :** This 16-bit field is incremented by one each time an RTP packet is sent. The number can be used by the receiver to detect packet loss and to recover packet sequence. The initial value is selected at random.
8. **Time stamp :** It is 32-bits number specifies the sampling instant of the first byte in the RTP data packet. This value can help to reduce jitter at the receiver by decoupling the playback from the packet arrival time. The initial value is selected at random.
9. **Synchronization source identifier :** This field tells which stream the packet belongs to. It is the method used to multiplex and demultiplex multiple data streams onto a single stream of UDP packets.
10. **Contributing source identifier :** This list of 0 to 15 thirty-two bit items specifies the contributing sources for the payload contained in the packet. This field is used when mixers are present in the studio.

RTP Control Protocol (RTCP)

- RTCP is the protocol of RTP. RTCP provides feedback on the quality of the data distribution. It does not support transport of data.
- RTCP provides the feedback on delay, jitter, congestion, bandwidth to the sources. This feedback information is sent in the form of RTCP sender and receiver reports. This information can be used by the encoding process to increases data rate.
- RTCP defines several types of packets to carry different types of control information.
- Types of packets are : Sender Report (SR), Receiver Report (RR), Source Description (SDES), BYE and APP.

Scanned with CamScanner

1. Sender report gives transmission and reception statistics from active senders.
2. Receiver report gives reception statistics from participants that are not active senders.
3. SDES provides source description items such as CNAME, email, name, phone number, location etc.
4. BYE indicates the end of participation by the sender.
5. APP provides application specific functions that are defined in profile specification.

University Questions

1. *Discuss the working principle of UDP.* **GTU : Summer-13, Marks 7**
2. *User datagram protocol.* **GTU : Winter-13, Marks 4**

3.5 Principle of Reliable Data Transfer

GTU : Dec.-10,11, June-11, May-12, Winter-12, Summer-13,14

- Problem of reliable data transfer occurs at application layer, application layer, transport layer and data link layer.
- Reliable data transfer means, data received without loss or corrupt. All the packets are delivered in the order in which they were sent.
- The complexity of the reliable data transfer protocol determined by characteristics of unreliable channel. Fig. 3.5.1 shows reliable data transfer service model.

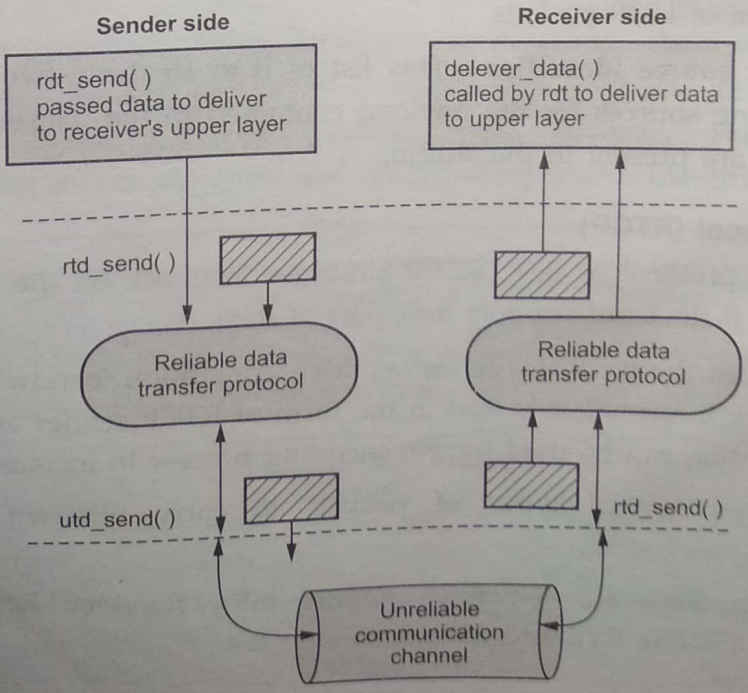


Fig. 3.5.1

Scanned with CamScanner

3.5.1 Building Reliable Data Transfer Protocol

- Develop sender side and receiver sides of reliable data transfer protocol. Here we consider only unidirectional data transfer. But the control information will flow in both directions.
- The finite state machine is used for specifying sender and receiver. Fig. 3.5.2 shows Finite State Machine (FSM) model.
- State : When in this state, next state is uniquely determined by next event.

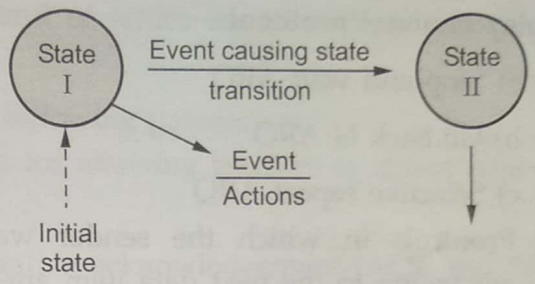


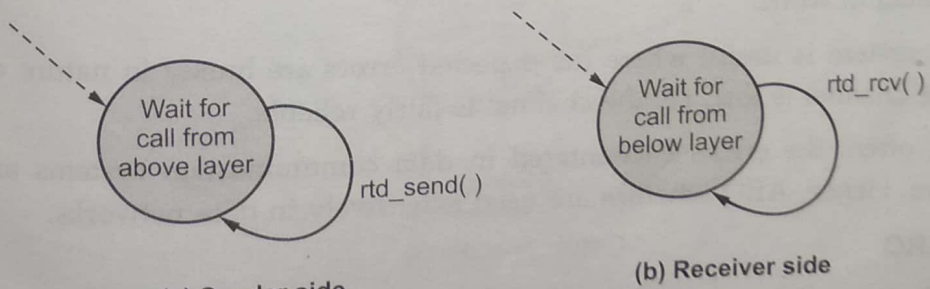
Fig. 3.5.2 FSM model

Sender side

- It takes data from upper layer by using system call rdt_send (data). Then it creates packet which contains user data and sends the packets into the communication channel.

Receiver side

- In this side, rdt receives a packet from the underlying communication channel via the rdt_rcv (packet) event. It removes the data from the packet and sends the passes the data up to the upper layer.
- Here, unit of data and packet is considered as same. All packet flow from sender to receiver.
- Fig. 3.5.3 shows sender and receiver reliable channel protocol.



(a) Sender side (b) Receiver side

Fig. 3.5.3 Protocol for reliable channel

Reliable Data Transfer over Channel with Bit Error

- Here we discuss the underlying channel may flip bits in packets. When packet is transmitted, bit errors occur.
- The protocols used for noiseless channel and noisy channels are as follows.

Scanned with CamScanner

1. Noiseless channel protocols are

a) Simplest

b) Stop and wait

2. Noisy channel protocol

a) Stop and wait ARQ

b) Go back N ARQ

c) Selective repeat ARQ

- Protocols in which the sender waits for a positive acknowledgement before advancing to the next data item are often called **Positive Acknowledgement with Retransmission** or **Automatic Repeat Request** (ARQ).
- In a real life network, the data link protocols are implemented as bidirectional; data flow in both directions.
- When data frame arrives, instead of immediately sending a separate control frame, the receiver restrains itself and waits until the network layer passes it the next, packet. The acknowledgement is attached to the outgoing data frame.
- In effect, the acknowledgement gets a free ride on the next outgoing data frame. The technique of temporarily delaying outgoing acknowledgements so that they can be hooked onto the next outgoing data frame known as **piggybacking.**
- In ARQ scheme, the information word is coded with adequate redundant bits so as to enable detection of errors at the receiving end.
- If an error is detected, the receiver asks the sender to retransmit the particular information word.
- ARQ system is useful where the expected errors are bursty in nature or error rate of the channel is low, i.e. the channel is fairly reliable.
- Most often, the errors encountered in data communication systems are bursty in nature. Hence, ARQ schemes are used extensively in data networks.

Steps in ARQ

An ARQ protocol is characterized by four functional steps.

1) Transmission of frames.

2) Error checking at the receiver end.

3) Acknowledgement.

a) Negative if error is detected (NAK).

b) Positive if no error is detected (ACK).

Scanned with CamScanner

4) Retransmission if acknowledgement is negative (NAK) or if no acknowledgement is received within a stipulated time.

- It may be noted that ARQ protocols require two way communication even if the information transfer is simplex, i.e. one way only. Information is exchanged in the form of frames, the beginning and the end of which are identified by means of flags of special characters.
- ARQ protocol handles bit error by using following parameters :
  1. **Error detection :** Create a mechanism for allowing receiver to detect when bit errors occured.
  2. **Receiver feedback :** Positive and negative acknowledgement (ACK and NAK) are used.
  3. **Retransmission :** A packet that is received in error at the receiver will be retransmitted by the sender.
- Fig. 3.5.4 shows reliable data transfer protocol with error.

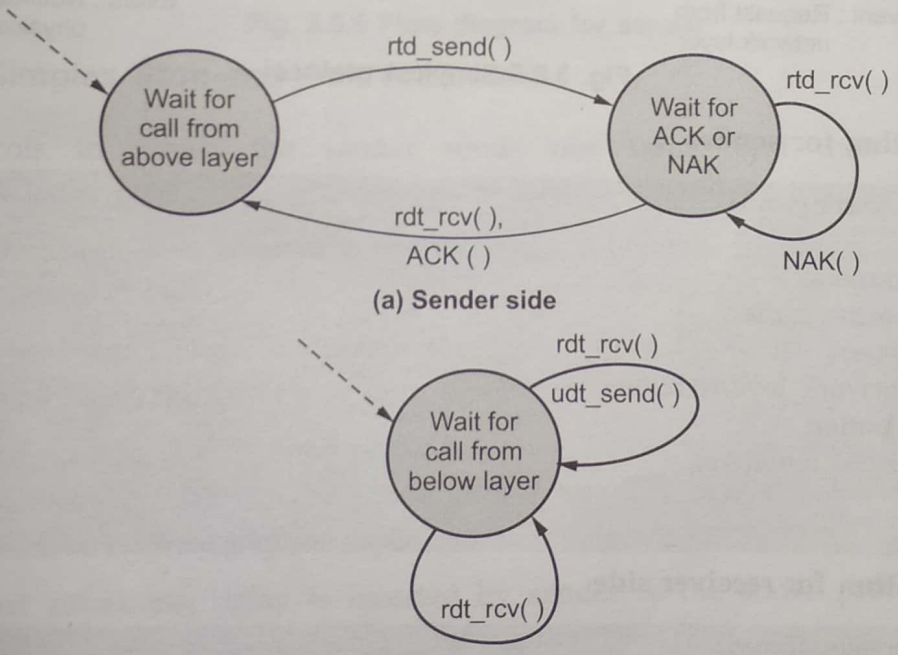


Fig. 3.5.4

3.5.2 Simplest Protocol

- In simplest protocol, there is no flow control and error control. It is a unidirectional protocol in which data frames are traveling in only one direction i.e. from the sender to receiver.
- We also assume that the receiver can immediately handle any frame it receives with a processing time that is small enough to be negligible.

Scanned with CamScanner

- The protocol consists of two distinct procedures a sender and a receiver. The sender runs in the data link layer of the source machine and the receiver runs in the data link layer of the destination machine. No sequence number or acknowledgements are used here.
- Fig. 3.5.5 shows the simplest protocol.

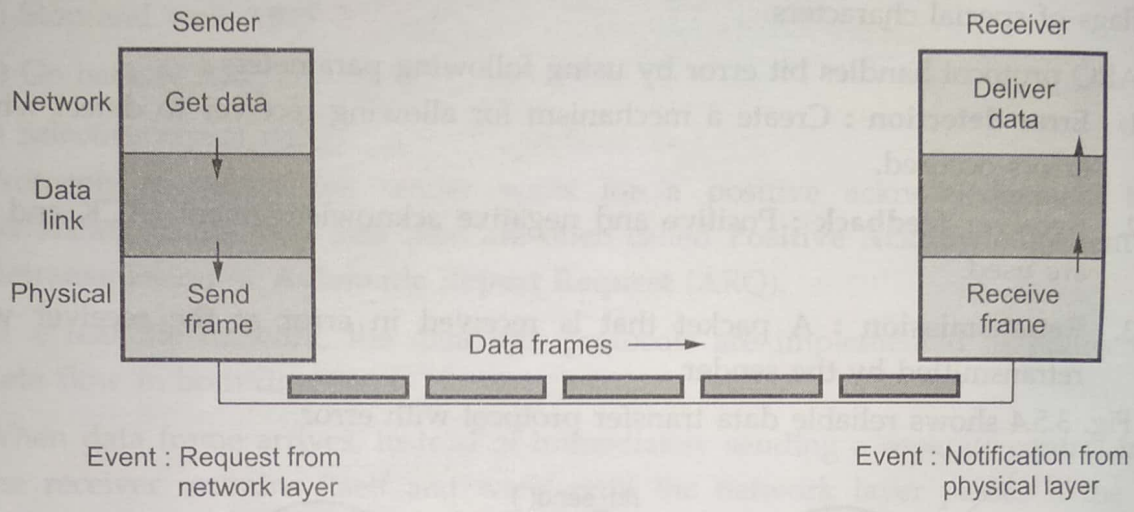


Fig. 3.5.5 Simplest protocol

- **Algorithm for sender**

```
void sender1 (void)
{
        frame s;
        packet buffer;
while(true){
from_network_layer(&bufer);
s.info=buffer;
to_physical_layer(&s);
}
}
```

- **Algorithm for receiver side**

```
void receiver1(void)
{
        frame r;
        event_type event;
        while(true){

 wait_for_event (&event);
 from_physical_layer(&r);

 to_network_layer(&r.info);
 }
 }
```


Scanned with CamScanner

- Fig. 3.5.6 shows the flow diagram for sender algorithm.

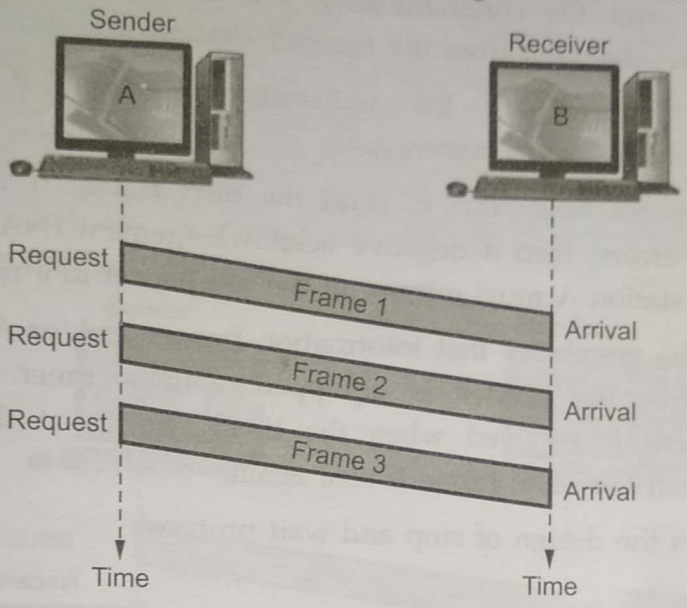


Fig. 3.5.6 Flow diagram for sender

3.5.3 A Simplex Stop-and-Wait Protocol

- Protocols in which the sender sends one frame and then waits for an acknowledgement before proceeding are called **stop-and-wait.**
- The communication channel is still assumed to be error free however and the data traffic is still complex.
- **Main problem :** How to prevent the sender from flooding the receiver with the data faster than the latter is able to process it.
- It is also assumed that there is no automatic buffering and queueing done within the receiver's hardware. The sender never transmit new frame until old one has been fetched by *from_physical_layer*.
- In some situations, delay is inserted by sender in the above protocol to slow it down sufficiently to keep from swamping the receiver.
- A more general solution to this dilemma is to have the receiver provide feedback (ACK) to the sender. After having passed a packet to its network layer, the receiver sends a little dummy frame back to the sender which, in effect, gives the sender permission to transmit the next frame.
- After having sent a frame, the sender is required by the protocol to bide its time until the little dummy (i.e., acknowledgement) frame arrives.
- Using feedback from the receiver to let the sender know when it may send more data is an example of the flow control.

Scanned with CamScanner

- The simplest retransmission protocol is stop-and-wait. Transmitter (station A) sends a frame over the communication line and then waits for a positive or negative acknowledgement from the receiver (station B).
- If no errors occurred in the transmission, station B sends a positive acknowledgement (ACK) to station A.
- The transmitter can now start to send the next frame. If frame is received at station B with errors, then a negative acknowledgement (NAK) is sent to station A. In this case station A must retransmit the old packet in a new frame.
- There is also the possibility that information frames and/or ACKs can be lost. To account for this, the sender is equipped with a timer. If no recognizable acknowledgement is received when the timer expires at the end of time out interval t_{out}, then the same frame is sent again.
- Fig. 3.5.7 shows the design of stop and wait protocol.

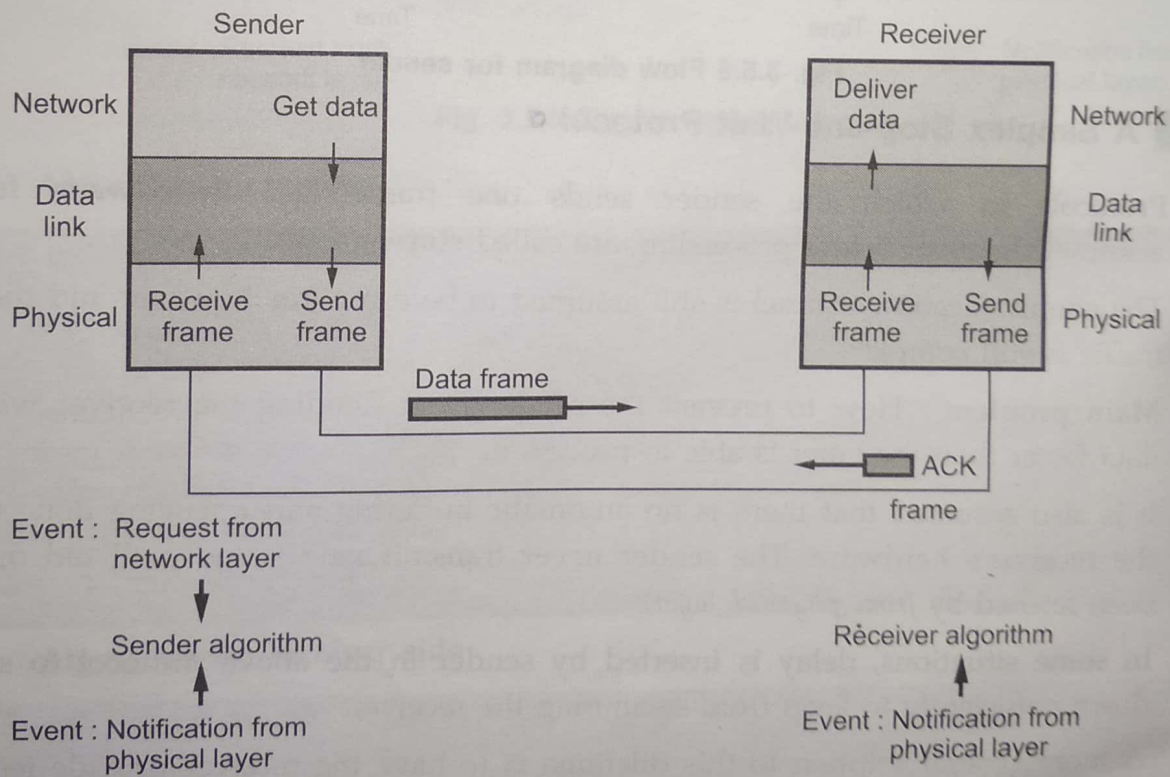


Fig. 3.5.7 Design of stop and wait protocol

- Protocols in which the sender sends one frame and then waits for an acknowledgement before process are called **stop and wait.**
- **Algorithm for sender**

```
void sender (void)
{
        frame s;
        packet buffer;
```

Scanned with CamScanner

```
        event_type event;
while(true){
        from_network_layer(&buffer);
        s.info=buffer;
        to_physical_layer(&s);
        wait_for_event(&event);
}
}
```

- Fig. 3.5.8 shows the flow diagram.

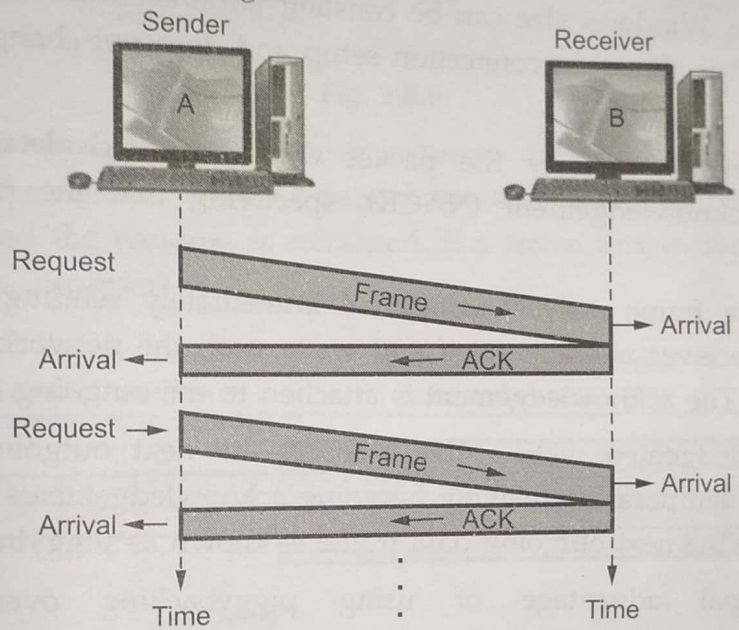


Fig. 3.5.8 Flow diagram for stop and wait

- **Algorithm for receiver side**

```
void receiver(void)
{
        frame r,s;
        event_type event;
 while(true){

wait_for_event (&event);
from_physical_layer(&r);
to_network_layer(&r.info);
to_physical_layer(&s);
}
}
```

- **Major drawback of stop-and-wait flow control :**
  1) Only one frame can be in transmission at a time.
  2) This leads to inefficiency if propagation delay is much longer than the transmission delay

Scanned with CamScanner

3.5.4 Pipelined Reliable Data Transfer Protocol Pipelined/Sliding Window Protocol

- Sliding windows is one of the methods of error correction. To increase the data rate, this method allows the sender to transmit a specific number of packets in continuous mode, i.e. at the maximum possible rate, without receiving positive acknowledgments for these packets.
- The number of packets that can be transmitted in such a way is known as the **window size.** Windows size can be constant parameters for this algorithm, which means that it is chosen at connection setup and does not change during the entire session.
- If the destination receives the packet with corrupted data, it might send a **Negative Acknowledgement (NACK),** specifying that the packet needs to be retransmitted.
- When a data frame arrives, instead of immediately sending a separate control frame, the receiver restrains itself and waits until the network layer passes it the next packet. The acknowledgement is attached to the outgoing data frame.
- The acknowledgement gets a free ride on the next outgoing data frame. The technique of temporarily delaying outgoing acknowledgements so that they can be hooked onto the next outgoing data frame is known as **piggybacking.**
- The principal advantage of using piggybacking over having distinct acknowledgement frames is a better use of the available channel bandwidth. The **ack** field in the frame header costs only a few bits, whereas a separate frame would need a header, the acknowledgement and a checksum.
- In a sliding window protocol each outbound frame contains a sequence number in the range 0 to some maximum (MaxSeq). If **n** bits are allocated in the header to store a sequence number then the number range would be from 0 to $2^n - 1$. **For example :** If a 3 bit number is used the sequence numbers would range from 0 to 7. The sender and receiver maintain a window.
- **Sending window :** It is a list of consecutive frame sequence numbers that can be sent by the sender or that have been sent and acknowledgments are waited for.
- When an ack arrives and all previous frames have already been acknowledged the window can be advanced and a new message obtained from the host to be transmitted with the next highest available sequence number. If ack arrives for a frame that is not within the 'window' it is discarded.

Scanned with CamScanner

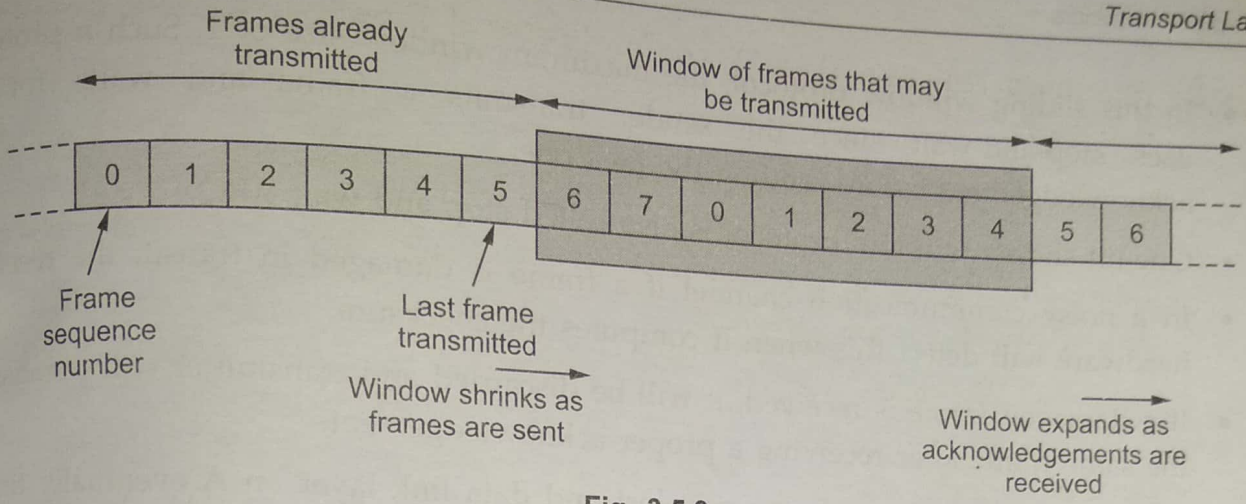


Fig. 3.5.9

- **Receiving window :** It is a list of sequence numbers for frames that can be accepted by the receiver. When a valid frame arrives and all previous frames have already arrived the window is advanced. If a frame arrives that is not within the 'window' it is discarded.

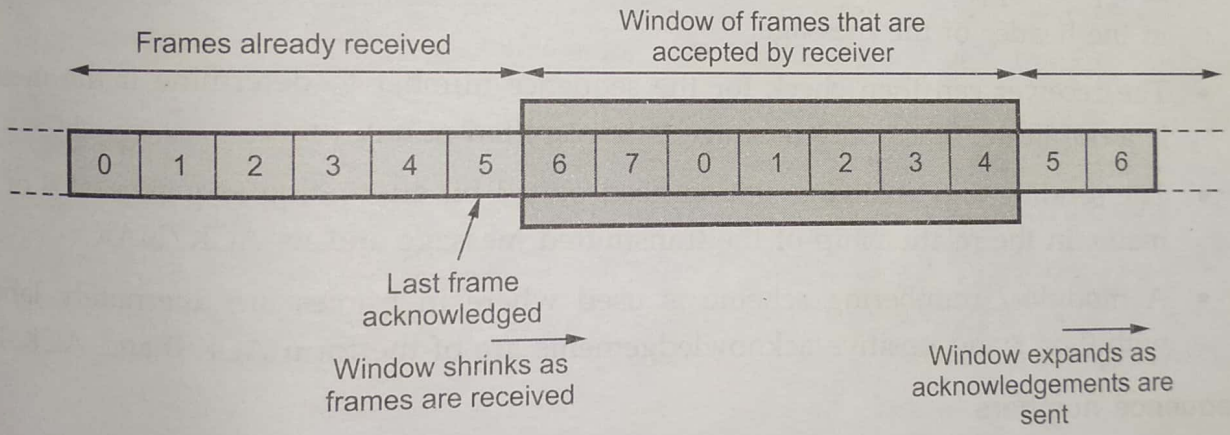


Fig. 3.5.10

Advantages of sliding windows protocol :

1) The sliding window is simpler, having only one set of parameters to manage.
2) Simultaneous communication in both directions is possible.
3) Better utilization of network bandwidth, especially if there are large transmission delays.
4) Traffic flow with reverse traffic data, known as piggybacking.

3.5.5 Stop and Wait ARQ Protocol

- This is the 'simplex protocol with sequence numbers and with the ack frame indicating the sequence number of the next frame expected'.

Scanned with CamScanner

- In this sliding window protocol, the maximum window size of 1. Such a protocol uses stop-and-wait since the sender transmits a frame and waits for its acknowledgement before sending the next one.
- One-bit sliding window protocol is also called stop and wait ARQ.
- In a noisy communication channel if a frame is damaged in transit, the receiver hardware will detect this when it computes the checksum.
- If a damaged frame is received, it will be discarded and transmitter will retransmit the same frame after receiving a proper acknowledgement.
- If the acknowledgement frame gets lost and data link layer on A eventually times out. Not having received an ACK, it assumes that its data frame was lost or damaged and sends the frame containing packet 1 again. This duplicate frame also arrives at data link layer on B, thus part of file will be duplicated and protocol is said to be failed.
- A typical approach to solve this problem is the provision for a sequence number in the header of the message.
- The receiver can then check for the sequence number to determine if the message is a duplicate. Since only message is transmitted at any time.
- The sending and receiving station need only 1-bit alternating sequence of 0 or 1 to maintain the relationship of the transmitted message and its ACK/NAK.
- A modulo-2 numbering scheme is used where in frames are alternately labelled with 0 or 1 and positive acknowledgements are of the form ACK 0 and ACK 1.

Sequence numbers

- The protocol specifies that frames need to be numbered. This is done by using sequence numbers. A field is added to the data frame to hold the sequence number of that frame.
- The sequence numbers are based on modulo-2 arithmetic.
- Fig. 3.5.11 shows the design of the stop and ARQ wait protocol.
(See Fig. 3.5.11 on next page)
- Stop-and-Wait ARQ is the simplest mechanism for error control and flow control.

Operation

- The sender transmits the frame, when frame arrives at receiver it checks for damage and acknowledges to the sender accordingly. While transmitting a frame there can be four situations.

Scanned with CamScanner

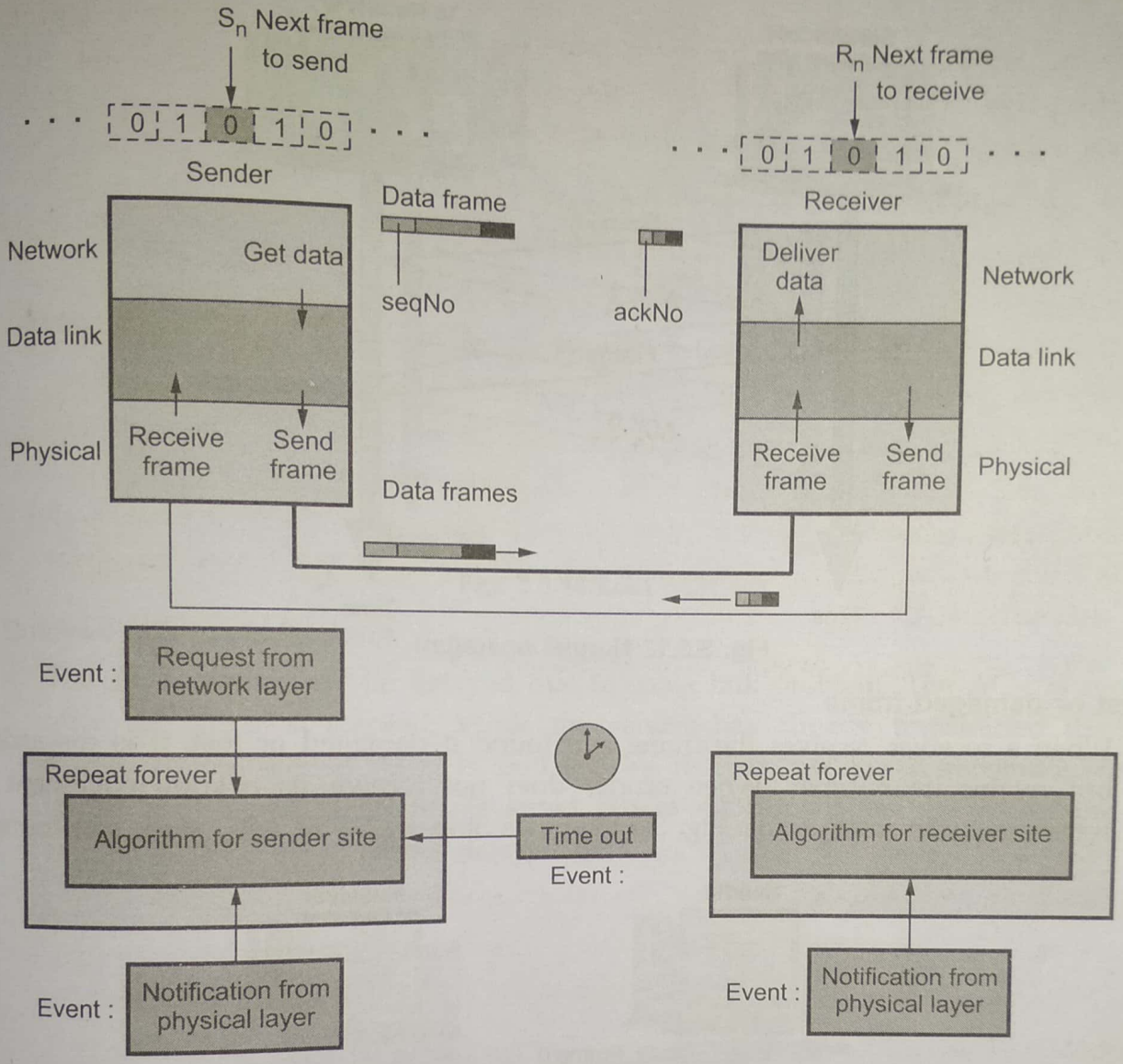


Fig. 3.5.11 Design of stop and wait ARQ protocol

a) Normal operation.

b) The frame is lost.

c) The acknowledgement is lost.

d) The acknowledgement is delayed.

a) Normal operation

- In normal operation the sender sends frame 0 and waits for acknowledgement ACK 1. After receiving ACK 1, sender sends next frame 1 and waits for its acknowledgement ACK 0. This operation is repeated. Fig. 3.5.12 shows this operation.
- Usually a timer is set by sender after each frame is transmitted, its acknowledgement must be received before timer expires.

Scanned with CamScanner

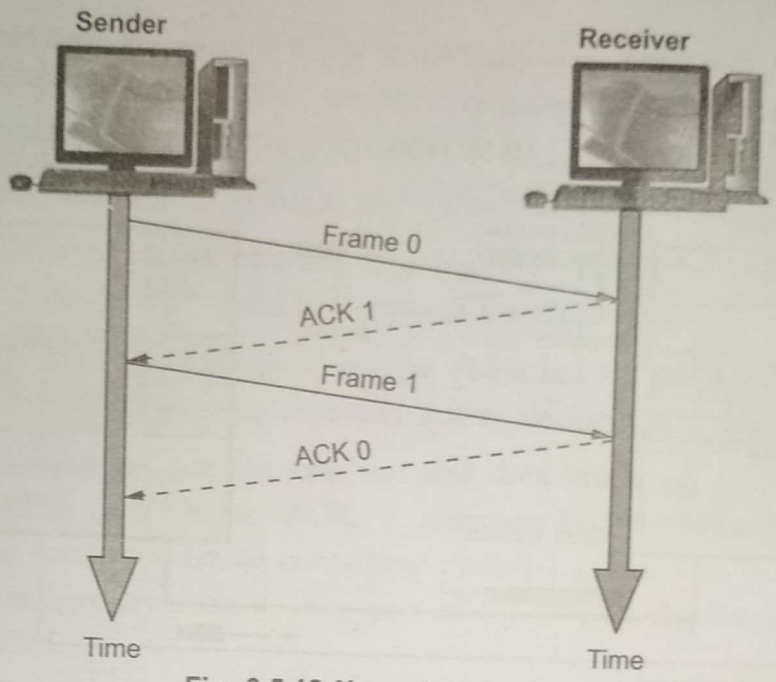


Fig. 3.5.12 Normal operation

b) Lost or damaged frame

- When a receiver receives the frame and found it damaged or lost, it is discarded but retains its number. When sender does not receive its acknowledgement it retransmits the same frame. Fig. 3.5.13 shows Stop-and-Wait ARQ with lost frame.

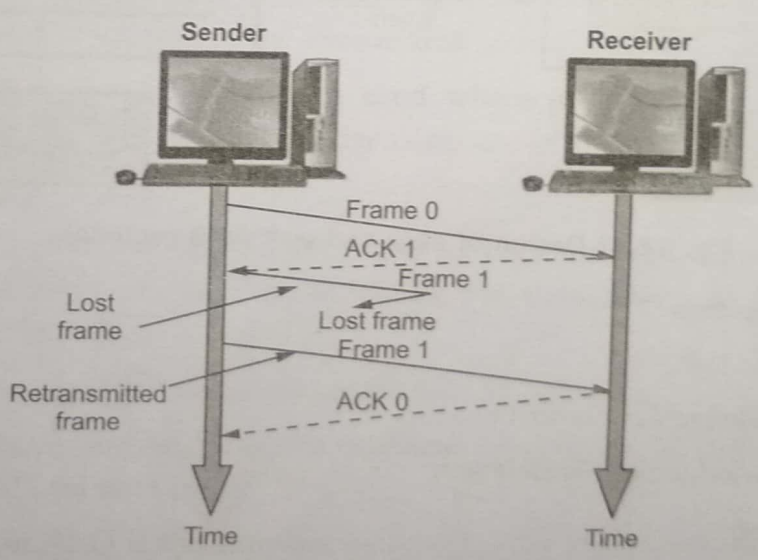


Fig. 3.5.13 Lost frame in Stop-and-Wait ARQ

c) Lost acknowledgement

- When an acknowledgement is lost, the sender does not know whether the frame is received by receiver. After the timer expires, the sender re-transmits the same frame. On the other hand, receiver has already received this frame earlier hence the second copy of the frame is discarded. Fig. 3.5.14 shows lost ACK.

Scanned with CamScanner

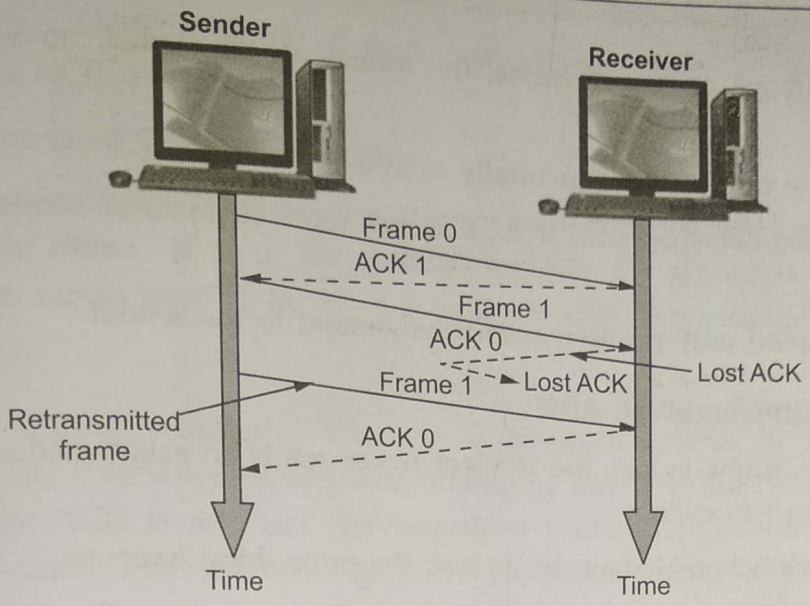


Fig. 3.5.14 Lost ACK

d) Delayed acknowledgement

- The ACK frame may be delayed due to some link problem. The ACK is received after the timer is elapsed. While the sender has already transmitted the same frame. Again second ACK is initiated by receiver for the retransmitted frame, hence the second ACK is discarded. To avoid duplication the ACKs must be numbered. Fig. 3.5.15 shows delayed ACK.

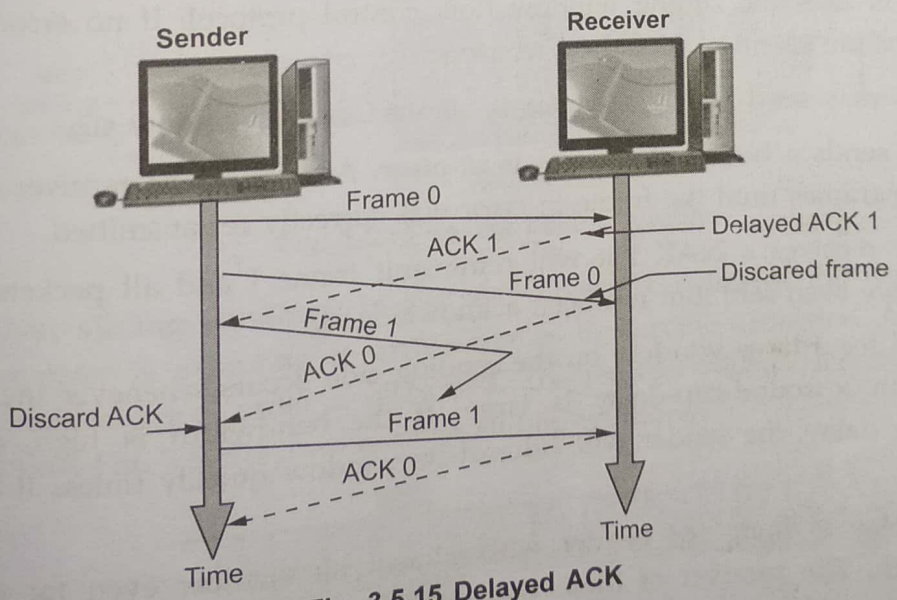


Fig. 3.5.15 Delayed ACK

3.5.5.1 Features of Stop-and-Wait ARQ

1) Sender keeps a copy of last transmitted frame until its ACK is received.
2) Both data frame and ACK frame are alternately numbered as 0 and 1 for identification of frame and to avoid duplication of frames.

Scanned with CamScanner

3) In case of damage or loss of frame, the frames are discarded, no acknowledgment is sent.

4) The frames are numbered sequentially to avoid duplication.

5) The sender maintains a timer, if ACK is not received in time, sender assumes it it lost.

6) The receiver send only positive acknowledgement to the sender.

Shortcomings of Stop-and-Wait ARQ

1) If the sender's frame is lost, the receiver never sends an acknowledgement, and the sender will wait forever.

2) If the receiver's acknowledgement is lost, the same thing happens.

3) If the acknowledgement is damaged, the sender may draw the wrong conclusion and make protocol fail.

4) Both sender and receiver do a lot of waiting, it is just like teacher is giving one assignment question at a time. The student takes the question at home, works on it, brings it back to school, gives it to the teacher and waits for the next question.

3.5.6 Go-Back-N ARQ

- Go-Back-N uses the sliding window flow control protocol. If no errors occur the operations are identical to sliding window.
- A station may send multiple frames as allowed by the window size.
- Receiver sends a NAK i if frame i is in error. After that, the receiver discards all incoming frames until the frame in error was correctly retransmitted.
- If sender receives a NAK i it will retransmit frame I and all packets i+1, i+2,... which have been sent, but not been acknowledged.
- The need for a large window on the sending side occurs whenever the product of bandwidth x round-trip-delay is large. If the bandwidth is high, even for a moderate delay, the sender will exhaust its window quickly unless it has a large window.
- If the delay is high, the sender will exhaust its window even for a moderate bandwidth. The product of these two factors basically tells what the capacity of the pipe is and the sender needs the ability to fill it without stopping in order to operate at peak efficiency. This technique is known as **pipelining.**
- As in Stop-and-Wait protocol senders has to wait for every ACK then next frame is transmitted. But in Go-Back-N ARQ W frames can be transmitted without

Scanned with CamScanner

waiting for ACK. A copy of each transmitted frame is maintained until the respective ACK is received.

Additional features of Go-Back-N ARQ

1) Sequence numbers : Sequence numbers of transmitted frame are maintained in the header of each frame. If k is the number of bits for sequence number, then the numbering can range from 0 to $2^k - 1$ e.g. for k = 3. Sequence numbers are 0 to 7 $(2^3 - 1)$.

2) Sender sliding window : Window is a set of frames in buffer waiting for acknowledgment. This window keeps on sliding in forward direction. The window size is fixed. As the ACK is received, the respective frame goes out of window and new frame to sent come into window. Fig. 3.5.16 illustrates sliding of window for window size = 7.

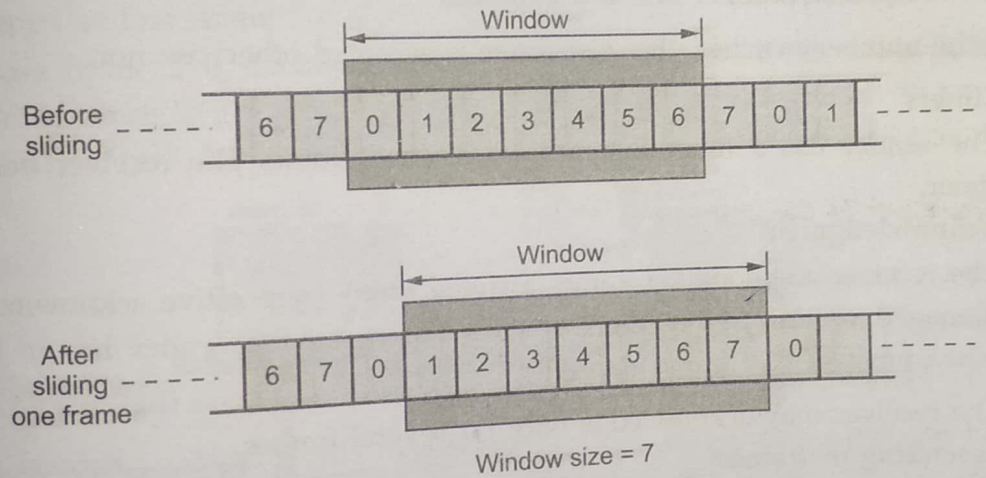


Fig. 3.5.16 Sender sliding window

3) Receiver sliding window : In the receiver side the size of the window is always one. The receiver is expecting to arrive frames in specific sequence. Any other frame received which is out of order is discarded. The receiver slides over after receiving the expected frame. Fig. 3.5.17 shows receiver sliding window.

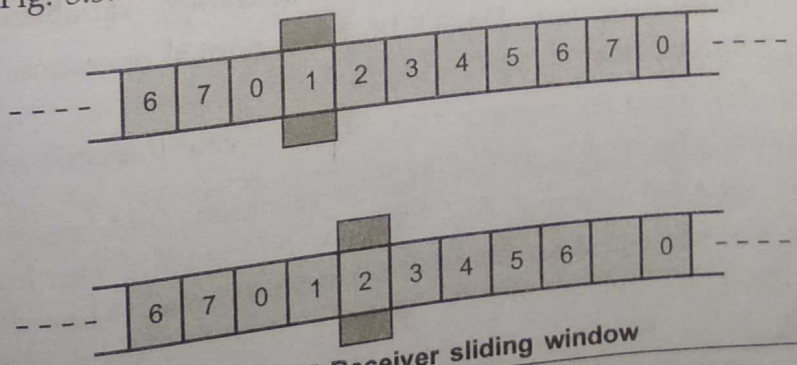


Fig. 3.5.17 Receiver sliding window

Scanned with CamScanner

4) Control variables :

a) Sender variables

- The sender deals with three different variables.

$S \rightarrow$ Sequence number of recently sent frame.

$S_F \rightarrow$ Sequence number of first frame in window.

$S_L \rightarrow$ Sequence number of last frame in window.

$\therefore$ Window size $W = S_L - S_F + 1$

e.g. in previous feature, $W = 7 - 0 + 1 = 8$

b) Receiver variable

- The receiver deals with one variable only.

$R \rightarrow$ Sequence number of frame expected

If the number matches, then the frame is accepted otherwise not.

5) Timers

- The sender has a timer for each transmitted frame. The receiver don't have any timer.

6) Acknowledgment

- The receiver responds for frames arriving safely by positive acknowledgments. For damaged or lost frames receiver does not reply, the sender has to retransmit it when timer of that frame elapsed.
- The receiver may acknowledge once for several frames.

7) Resending of frames

- If the timer for any frame expires, the sender has to resend that frame and the subsequent frames also, hence the protocol is called Go-Back-N ARQ.

Operation

a) Normal operation

- The sender sends frames and update the control variables i.e. S_F, S, S_L and receiver updates variable R. Fig. 3.5.18 shows normal operation.

(See Fig. 3.5.18 on next page)

Scanned with CamScanner

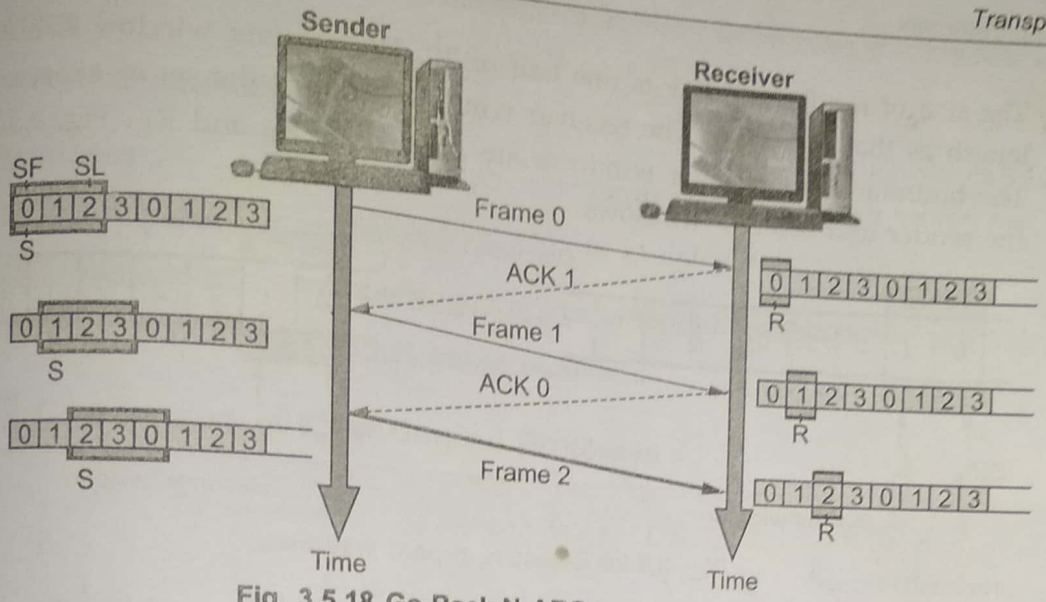


Fig. 3.5.18 Go-Back-N ARQ, normal operation

b) Damaged or lost frame

- Suppose frame 2 is damaged or lost and if receiver receives frame 3, it will be discarded since it is expecting frame 2. Sender retransmits frame 2 and frame 3. Fig. 3.5.19 shows this process.

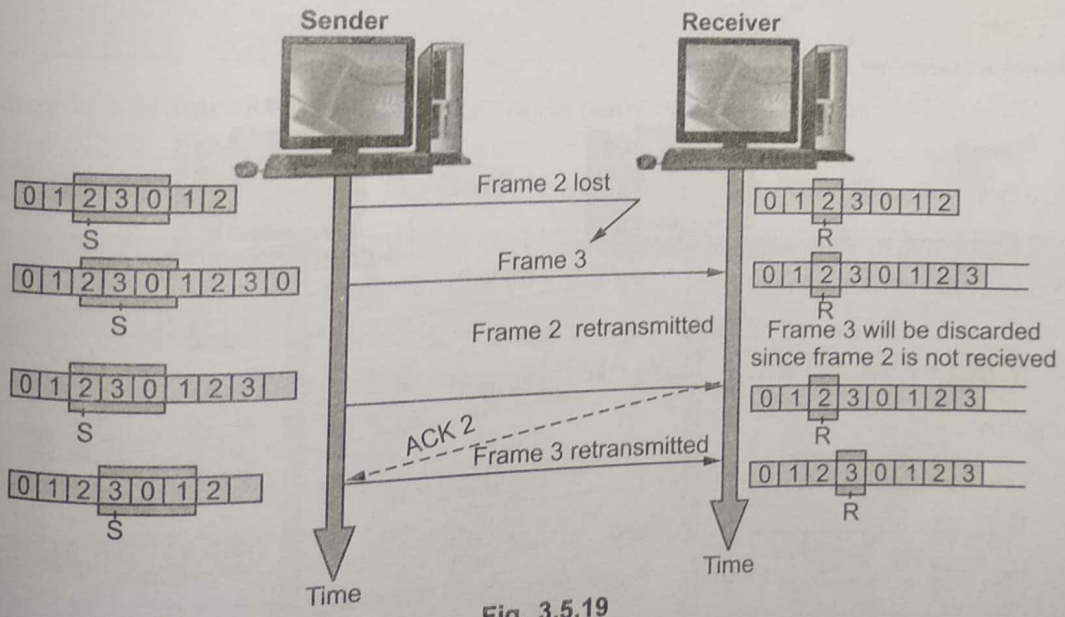


Fig. 3.5.19

3.5.7 Selective Repeat ARQ

- Selective repeat ARQ retransmitts only the damaged or lost frames instead of sending multiple frames. The selective retransmission increases the efficiency of transmission and is more suitable for noisy channel. The circuit complexities at the receiver side increases.

Scanned with CamScanner

- The size of sender window is one half of 2^k. The receiver window size is of same length as that of sender. The receiver window includes the set of expected frames. The boundaries of receiver windows are defined by R_F and R_L. Fig. 3.5.20 shows the sender and receiver windows.

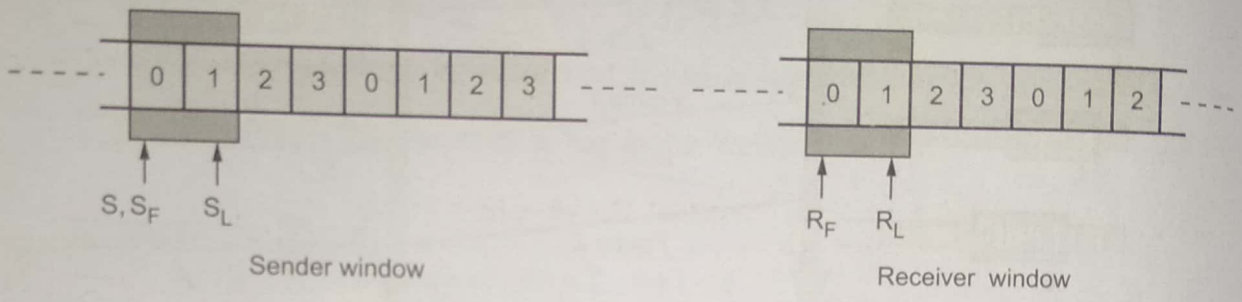


Fig. 3.5.20 Selective repeat windows

- Negative acknowledgement (NAK) is used for lost or damaged frames.

Operation

- In sequential transmission of frame 0, 1, 2, 3, suppose frame 2 is lost and the next frame 3 is already received then receiver sends NAK 2 frame to sender. Then sender retransmits frame 2 only. Fig. 3.5.21 shows operation of selective repeat ARQ.

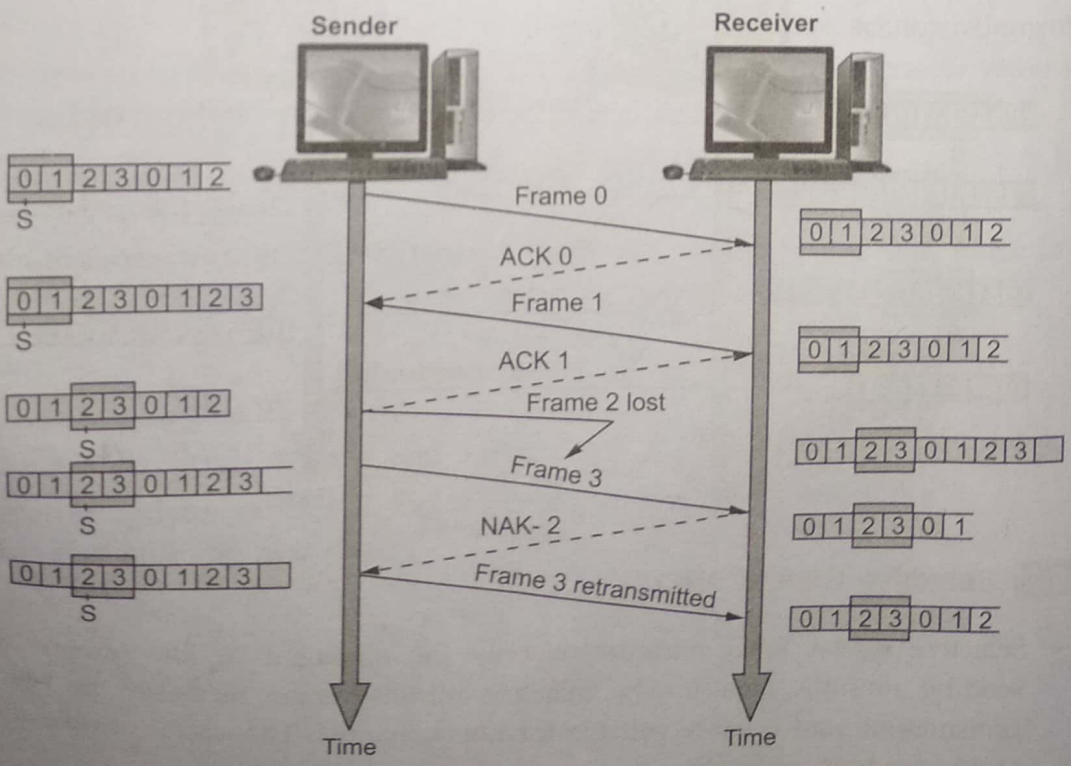


Fig. 3.5.21 Selective repeat ARQ

Scanned with CamScanner

Advantage :

1) Fewer retransmissions.

Disadvantages :

1) More complexity at sender and receiver.
2) Each frame must be acknowledged individually (no cumulative acknowledgements).
3) Receiver may receive frames out of sequence.

3.5.8 Comparison of Flow Control Protocols

Sr. No.	Parameters	Stop-and-Wait protocol	Go-Back-N protocol	Selective repeat protocol
1.	Sending window size	One frame	Less than 2^k	Less than or equal to 2^k minus receiving window size
2.	Receiving window size	One frame	One frame	Less than or equal to 2^k minus sending window size

where k = Number of bits used for frame number.

3.5.9 Difference between Go-Back-N and Selective Repeat

Sr. No.	Go-Back-N	Selective repeat
1.	Go-back-N requires all retransmission of the succeeding frame along with the lost or damaged frame.	In selective repeat, only the specific damaged or lost frame is retransmitted.
2.	Sender does not require any logic to select the specific frame for retransmission.	Extra logic is required for searching and retransmission of specific frame.
3.	Receiver do not required any sort of storage and sorting mechanism.	The complexity of sorting and storage mechanism is required by the receiver.
4.	It is not expensive.	It is expensive.

Scanned with CamScanner

Example 3.5.1 *Consider the use of 1000-bit frames on a 1 Mbps satellite channel with a 270 ms delay. What is the maximum link utilization for*

a) Stop-and-wait flow control ?

b) Sliding window flow control with a window size of 7 ?

c) Sliding window flow control with a window size of 127 ?

d) Sliding window flow control with a window size of 255 ?

Solution : Given data :

Frame = 1000 bits,

Channel data rate = 1 Mbps,

Propagation delay = 270 ms

a) Maximum link utilization with stop-and-wait flow control :

$$U = \frac{1}{1+2a}$$

$$\text{where } a = \frac{t_{prop}}{t_{frame}}$$

Since t_{prop} = 270 ms.

In order to find the value of U, we need to calculate t_{frame}

Frame = 1000 bits

Max bit rate = Channel bit rate = 1 Mbps then

$$t_{frame} = \frac{1000}{10^6}$$

$$U = \frac{1}{1+2a} = \frac{1}{1+2\times270} = 1.85 \times 10^{-3} = 0.185\ \%$$

b) Maximum link utilization with flow control of windows size 7 :

Maximum link utilization for window flow control is U = 1 for W 2a + 1

$$U = \frac{W}{2a+1} \qquad \text{for } W < 2a + 1$$

Since W = 7 and a = 270

Then (2a + 1) = (2 × 270 + 1) = 541 which means that W < 2a + 1

$$U = \frac{W}{2a+1} = \frac{7}{541} = 0.013 = \mathbf{1.3\ \%}$$

c) Maximum link utilization with flow control of windows size 127 :

Maximum link utilization for window flow control is U = 1 for W 2a + 1

Scanned with CamScanner

$$U = \frac{W}{2a+1} \qquad \text{for } W < 2a + 1$$

Since $W = 127$ and $a = 270$

Then $(2a + 1) = (2 \times 270 + 1) = 541$ which means that $W < 2a + 1$

$$U = \frac{W}{2a+1} = \frac{127}{541} = 0.235 = 23.5\ \%$$

d) Maximum link utilization with flow control of windows size 255 :

Maximum link utilization for window flow control is U = 1 for W 2a+1

$$U = \frac{W}{2a+1} \qquad \text{for } W < 2a + 1$$

Since $W = 255$ and $a = 270$

Then $(2a + 1) = (2 \times 270 + 1) = 541$ which means that $W < 2a + 1$

$$U = \frac{W}{2a+1} = \frac{W}{2a+1} = 0.471 = \mathbf{47.1\ \%}$$

Example 3.5.2 *A channel has a data rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50 % ?*

Solution : Data rate = 4 kbps

$$\text{Bit duration} = \frac{1}{4000} = 0.25 \text{ ms}$$

Time to transmit frame is (t_{frame}) :

$$t_{frame} = \frac{\text{Frame size}}{\text{Bit rate}} = \text{Frame size} \times \text{Bit duration}$$

For stop and wait flow control, efficiency $(U) = \frac{1}{(2a+1)}$

Where $a = t_{prop} / t_{frame}$

$t_{prop} = 20$ ms

Solving this equation with respect to a

$$\mathbf{a = 0.5[(1/U) - 1]}$$

For $U \geq 50\ \% = 0.5$, then $a \leq 0.5\left[(1/0.5) - 1\right] \Rightarrow a \leq 0.5$

Since $a = t_{prop}/t_{frame}$ then $t_{prop}/t_{frame} \leq 0.5 \Rightarrow t_{frame} \geq 2t_{prop}$

But frame_size $= t_{frame}$/bit_ duration =>

Frame_size $\geq 2t_{prop}$ / bit_ duration $= 2 \times 20$ ms / 0.25 ms $= 160$

Scanned with CamScanner

University Questions

1. *Explain the one bit sliding window protocol and go back n protocol. Write down the drawback of both the protocols.* **GTU : Dec.-10, Marks 7**
2. *Explain piggybacking, 1-bit sliding window protocol with go back n and selection repeat.* **GTU : June-11, Marks 7**
3. *Explain stop and wait ARQ in detail.* **GTU : June-11, Marks 7**
4. *Explain sliding window protocol using GO back to N.* **GTU : Dec.-11, Marks 7**
5. *Explain STOP and wait protocol.* **GTU : Dec.-11, Marks 7**
6. *Explain the term piggybacking, sending window and receiving window. Explain one-bit sliding window protocol.* **GTU : May-12, Marks 7**
7. *Explain a protocol using Go Back N strategy using pipelining and show the scenario in the case of 1) when receiver window size is 1 and 2) when receiver window size is large.* **GTU : Winter-12, Marks 7**
8. *Explain the working principle of stop and wait protocol.* **GTU : Summer-13, Marks 7**
9. *Explain sliding window protocol for sender and receiver.* **GTU : Winter-13, Marks 7**
10. *Explain stop and wait protocol for simplex communication.* **GTU : Winter-13, Marks 7**
11. *Explain sliding window protocol.* **GTU : Summer-14, Marks 7**
12. *Compare : Simplex and stop and wait protocol.* **GTU : Summer-14, Marks 7**

3.6 Connection Oriented Transport (TCP)

GTU : Dec.-11, Summer-13,14, Winter-12,14

- Transmission Control Protocol (TCP) is the connection oriented protocol whereas User Data Protocol (UDP) is connectionless protocol. Both are internet protocols used in the transport layer.
- TCP provides a connection-oriented, reliable, byte stream service. The term connection oriented means the two applications using TCP must establish a TCP connection with each other before they can exchange data.

3.6.1 TCP Services

- TCP and UDP use the same network layer (IP), TCP provides totally different services. TCP provides a connection-oriented, reliable, byte stream service. There are exactly two end points communicating with each other on a TCP connection.
- TCP does not support multicasting and broadcasting. The application data is broken into what TCP considers the best sized chunks to send. The unit of information passed by TCP to IP is called a segment.

Scanned with CamScanner

- When TCP sends a segment it maintains a timer, waiting for the other end to acknowledge reception of segment. If an acknowledgement isn't received in time, the segment is retransmitted.
- When TCP receives data from the other end of the connection, it sends an acknowledgement. TCP maintains a checksum on its header and data.
- TCP segments are transmitted as IP datagrams and since IP datagrams can arrive out of order, TCP segments can arrive out of order. Since IP datagrams can get duplicated, a receiving TCP must discard duplicate data.
- TCP also provides flow control. Each end of a TCP connection has a finite amount of buffer space. A receiving TCP only allows the other end to send as much data as the receiver has buffers for. This prevents a fast host from taking all the buffers on a slower host.
- A TCP connection is a byte stream, not a message stream. A stream of 8-bit bytes is exchanged across the TCP connection between the two applications. There are no record markers automatically inserted by TCP. This is called a **byte stream service**.
- If the application on one end writes 20 bytes followed by a write of 40 bytes, followed by a write of 80 bytes, the application at the other end of the connection cannot tell what size the individual writes there. The other end may read 140 bytes once at a time or 140 bytes in two reads of 70 bytes at a time.
- TCP does not interpret the contents of the bytes at all. TCP has no idea if the data bytes being exchanged are binary data, ASCII character or any other.

3.6.2 TCP Segment Format

- The TCP data is encapsulated in an IP datagram as shown in the Fig. 3.6.1 (a).

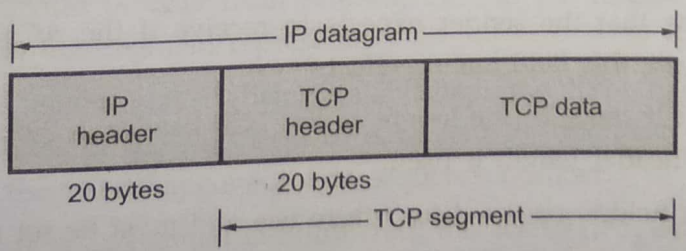


Fig. 3.6.1 (a) Encapsulation of TCP data

Scanned with CamScanner

- Fig. 3.6.1 (b) shows the format of the TCP header.

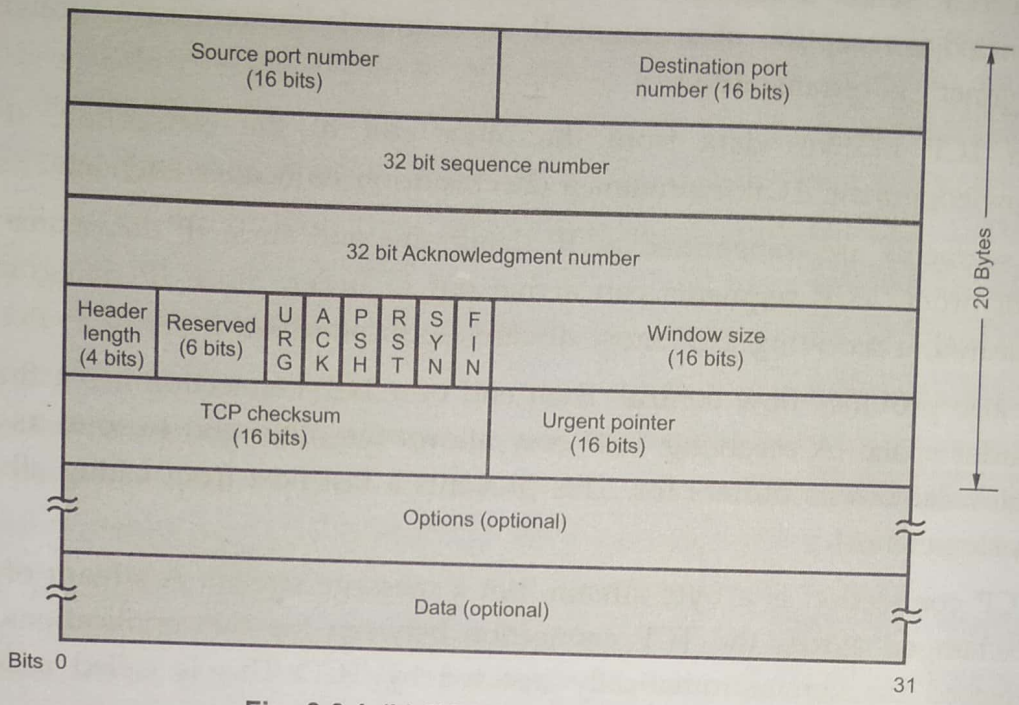


Fig. 3.6.1 (b) TCP header format

- Description of field in the TCP header as follows :

1. **Source port :** It specifies the application sending the segment. This is different from the IP address, which specifies an internet address.
2. **Destination port :** It identifies the receiving application port numbers below 256 called well-known ports and are assigned to commonly used applications. For examples, port 23 corresponds to a Telnet function. Port 53 for DNS name server and port 21 assigned for FTP.
3. **Sequence number :** Each byte in the stream that TCP sends is numbered. The sequence number wraps back to 0 after $2^{32} - 1$.
4. **Acknowledgement number :** This field identifies the sequence number of the next data by the that the sender expects to receive if the ACK bit is set. If the ACK bit is not set, this field has no effect.
5. **Header length :** It specifies the length of the TCP header in 32-bit words. Because of option field, header length is used.
6. **Reserved :** This field is reserved for future use and must be set to 0 (zero).
7. TCP header contains six flag bits. One or more than one can be turned on at the same time. The function of each flag is as follows.
   a. **URG :** The Urgent pointer is valid if it set to 1.
   b. **ACK :** ACK bit is set to 1 to indicate that the acknowledgement number is valid.

Scanned with CamScanner

c. **PSH :** The receiver should pass this data to the application as soon as possible.

d. **RST :** This flag is used to reset the connection. It is also used to reject an invalid segment.

e. **SYN :** Synchronize sequence number to initiate a connection. The connection request has SYN = 1 and ACK = 0 to indicate that the piggyback acknowledgement field is not in use.

f. **FIN :** The FIN bit is used to release a connection. It specifies that the sender is finished sending data.

8. **Window size :** It specifies the number of bytes the sender is willing to accept. This field can be used to control the flow of data and congestion.

9. **Checksum :** Used for transport layer error detection.

10. **Urgent pointer :** If the URG flag bit is set, the segment contains urgent data meaning the receiving TCP entity must deliver it to the higher layers immediately.

11. **Options :** Size of this field is variable options field may be used to provide other functions that are not covered by the header.

12. **Data :** Data field size is variable. It contains user data.

- TCP header normal size is 20 bytes, unless options are present. Each TCP segment contains the source and destination port number to identify the sending and receiving application.
- The port number along with the source and destination IP addresses in the IP header, uniquely identify each connection. The combination of an IP address and a port number is sometimes called a **socket**.
- Sequence number is a 32-bit unsigned number. Sequence number identifies the byte in the stream of data from sending TCP to the receiving TCP that the first byte of data in this segment represents.
- When a new connection is being established, the SYN flag is turned on. The sequence number of the first byte of data sent by this host will be the ISN plus one because, the SYN flag consumes a sequence number.
- Every byte that is exchanged is numbered, the acknowledgement number contains the next sequence number that the sender of the acknowledgement excepts to receive. Therefore the sequence number plus 1 of the last successfully received byte of data. This field is valid only if the ACK flag is on.
- TCP provides full duplex service. Therefore, each end of a connection must contain a sequence number of the data flowing in each direction.

Scanned with CamScanner

- TCP can be described as a sliding window protocol without selective or negative acknowledgements.
- The TCP header length tells how many 32-bit words are contained in the TCP header. This information is needed because the options field is of variable length with a 4-bit field, TCP is limited to a 60-byte header. Without options, the normal size is 20 bytes.
- TCP's flow control is handled using a variable size sliding window. This is the number of bytes, starting with the one specified by the acknowledgement number field, that the receiver is willing to accept.
- This is a 16-bit field, limiting the window to 65535 bytes.
- The checksum covers the TCP segment, the TCP header and the TCP data. Checksum field must be calculated and stored by the sender and then verified by the receiver.
- The urgent pointer is valid only if the URG flag is set. This pointer is a positive offset that must be added to the sequence number field of the segment to yield the sequence number of the last byte of urgent data. Option field is the maximum segment size option, called the Maximum Segment Size (MSS). MSS is the largest chunk of data that TCP will send to the other end.

3.6.3 TCP Protocol

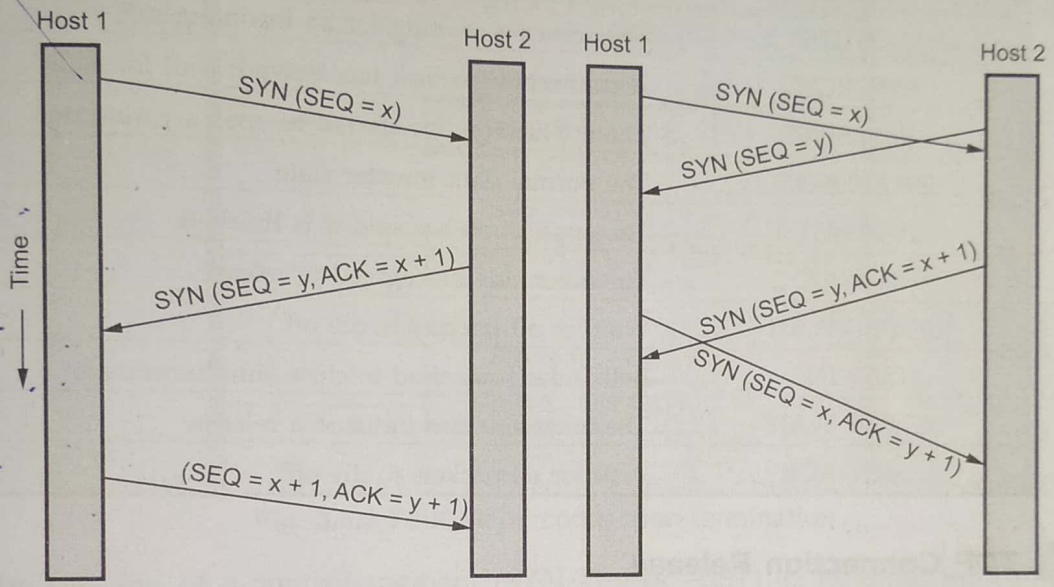
- Sending and receiving TCP entities exchange data in the form of segments. A TCP segment consists of a fixed 20-byte header followed by zero or more data bytes.
- TCP software decides how big segments should be. Two limits restrict the segment size.
  1. Each segment including the TCP header must fit in the 65515 bytes IP payload.
  2. Each network has a Maximum Transfer Unit (MTU) and each segment must fit in the MTU.
- The basic protocol used by TCP entities is the sliding window protocol.

3.6.4 TCP Connection Establishment

- Connection establishment in a TCP session is initialized through a three-way handshake. To establish the connection, one side (server) passively waits for an incoming connection by executing the LISTEN and ACCEPT primitives, either specifying a specific source.

Scanned with CamScanner

- Other side (client) executes a CONNECT primitive specifying the IP address and port to which it wants to connect, the maximum TCP segment size it is willing to accept, and optionally some user data.
- Fig. 3.6.2 shows the TCP connection establishment in the normal case and call collision.



(a) TCP connection establishment in the normal case (b) Call collision

Fig. 3.6.2 TCP connection establishment

- A connection is established using a three-way handshake.
- The transmitter sends Connection Request (seq = x) to start a connection with transmitter message id x.
- The receiver replies Connection Accepted (seq = y, ACK = x+1), to acknowledge x and establish for its messages the identity y.
- Finally the transmitter confirms the connection with Connection Accepted (seq = x+1,ACK = y+1) to confirm its own identifier x and accept the receiver's identifier y.
- If the receiver wanted to reject x, it would send Reject(ACK = x).
- If the transmitter wanted to reject y it would send Reject(ACK = y).
- As part of the handshake the transmitter and receiver specify their MSS (Maximum Segment Size) that is the maximum size of a segment they can accept. A typical value for MSS is 1460.
- TCP connections are full duplex. The steps required establishing and release connections can be represented in a finite state machine.

Scanned with CamScanner

- The states used in the TCP connection management finite state machine are as follows :

State	Description
CLOSED	No connection is active or pending.
LISTEN	The server is waiting for an incoming call.
SYN RCVD	A connection request has arrived; wait for ACK.
SYN SENT	The application has started to open a connection.
ESTABLISHED	The normal data transfer state.
FIN WAIT 1	The application has said it is finished.
FIN WAIT 2	The other side has agreed to release.
TIMED WAIT	Wait for all packets to die off.
CLOSING	Both sides have tried to close simultaneously.
CLOSE WAIT	The other side has initiated a release.
LAST ACK	Wait for all packets to die off.

3.6.5 TCP Connection Release

- Any of the two parties involved in exchanging data can close the connection. When connection in one direction is terminated, the other party can continue sending data in the other direction.
- Four steps are required to close the connection in both direction. Fig. 3.5.3 shows four step connection termination.
- Steps are as follows

1. The client TCP sends the first segment, a FIN segment.
2. The server TCP sends the second segment, an ACK segment, to confirm the receipt of the FIN segment from the client.
3. The server TCP can continue sending data in the server client direction. When it does not have any more data to send, it sends the third segment.
4. The client TCP sends the fourth segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. (See Fig. 3.6.3 on next page).

3.6.6 TCP Connection Management Modeling

- Fig. 3.6.4 shows connection management finite state machine.
- The lightface lines are unusual event sequences. Each line in Fig. 3.6.4 is marked by an event/action pair. The event can either be a user-initiated system call (CONNECT LISTEN, SEND or CLOSE), a segment arrival (SYN, FIN, ACK or RST) or in one case, a timeout of twice the maximum packet lifetime. The action is

Scanned with CamScanner

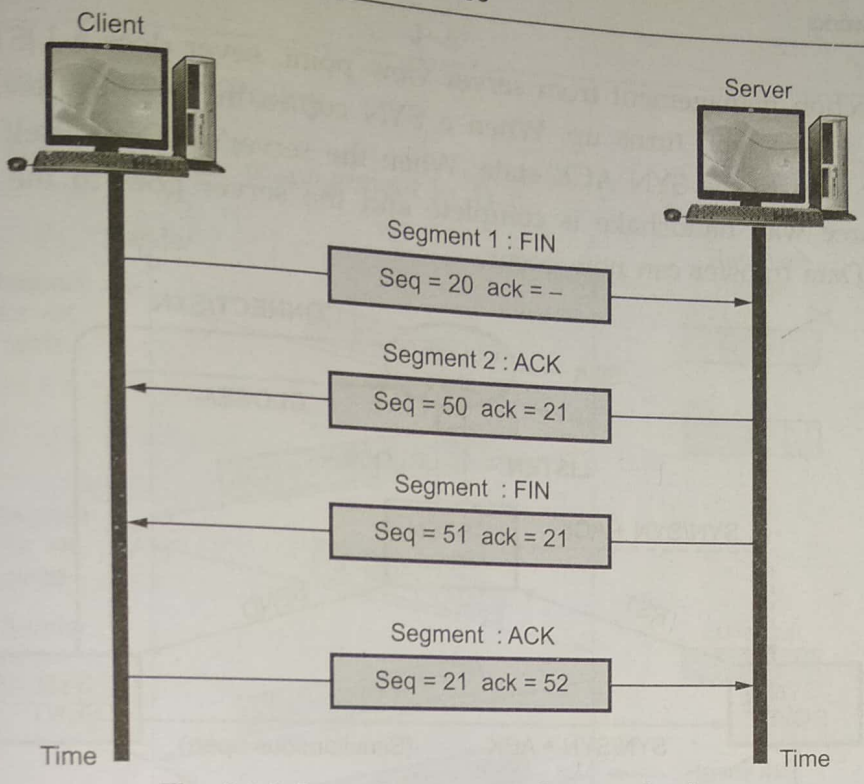


Fig. 3.6.3 Four steps connection termination

the sending of a control segment (SYN FIN or RST) or nothing, indicated by ---. Comments are shown in parentheses.

- The diagram can best be understood by first following the path of a client (the heavy solid line) then later the path of a server (the heavy dashed line). When an application on the client machine issues a CONNECT request, the local TCP entity creates a connection record, marks it as being in the SYN SENT state, and sends a SYN segment.
- Note that many connections may be open (or being opened) at the same time on behalf of multiple applications, so the state is per connection and recorded in the connection record. When the SYN + ACK arrives. TCP sends the final ACK of the three-way handshake and switches into the ESTABLISHED state data can now be sent and received.
- When an application is finished, it executes a CLOSE primitive, which causes the local TCP entity to send a FIN segment and wait for the corresponding ACK (dashed box marked active close).
- When the ACK arrives, a transition is made to state FIN WAIT 2 and one direction of the connection is now closed. When the other side closes, too, a FIN comes in, which is acknowledged. Now both sides are closed, but TCP waits a time equal to the maximum packet lifetime to guarantee that all packets from the connection have died off, just in case the acknowledgement was lost. When the timer goes off, TCP deletes the connection record.

Scanned with CamScanner

- Connection management from server view point, sever does a LISTEN and settles down to see who turns up. When a SYN comes in, it is acknowledged and the server goes to the SYN ACK state. When the server's SYN is itself acknowledged, the three way handshake is complete and the server goes to the ESTABLISHED state. Data transfer can now occur.

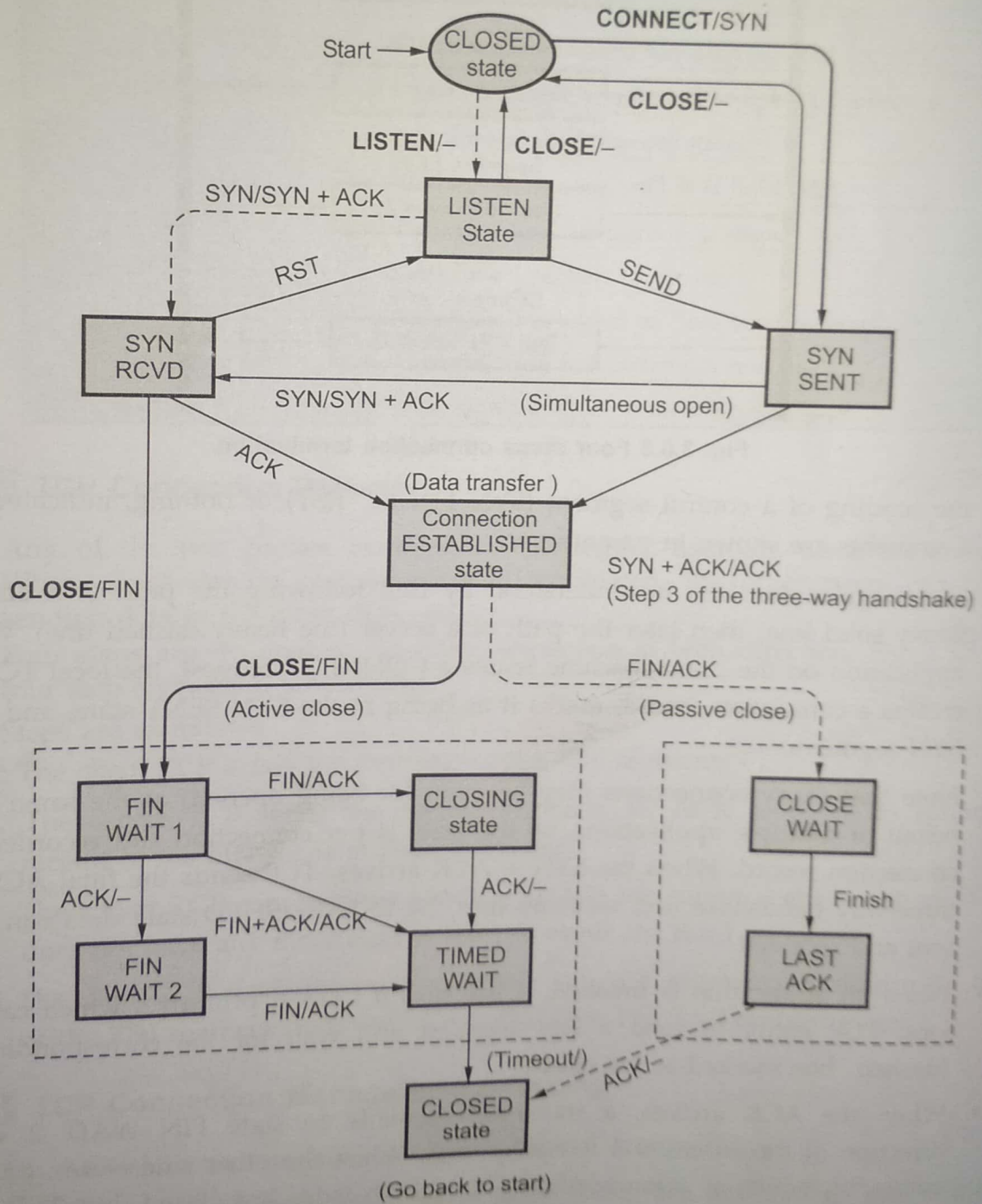


Fig. 3.6.4 Finite state machine for TCP connection

Scanned with CamScanner

3.6.7 TCP Transmission Policy

- Fig. 3.6.5 shows window management in TCP.

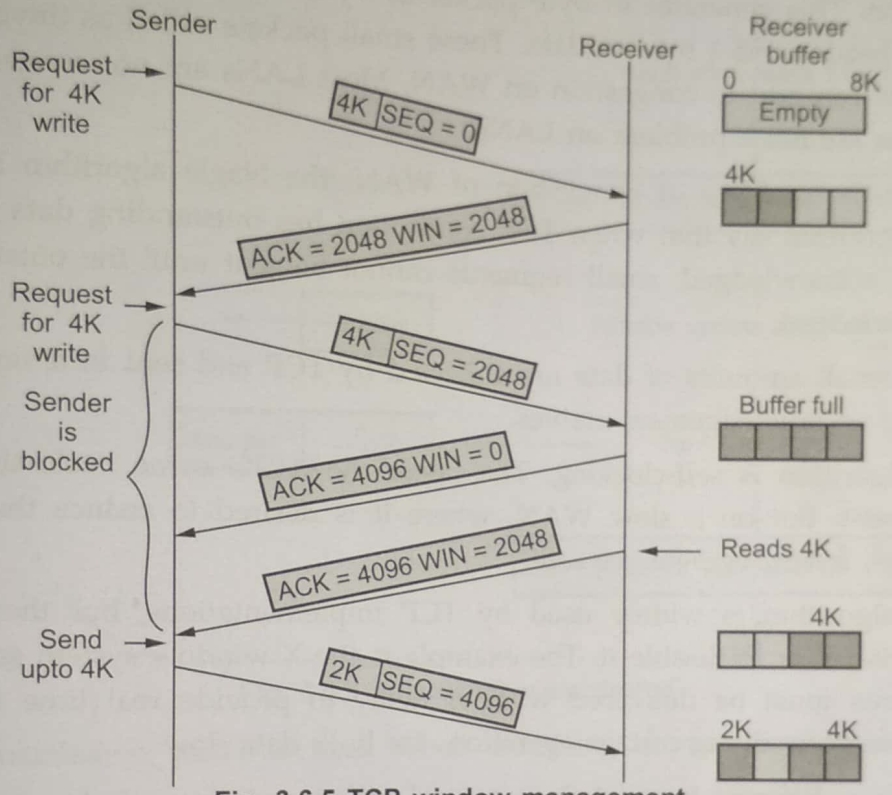


Fig. 3.6.5 TCP window management

- Let us assume that receiver buffer size is 4096-byte.
- If the sender transmits a 2048-byte segment that is correctly received, the receiver will acknowledge the segment.
- 2048 bytes of buffer space is only available and it will advertise a window of 2048 starting at the next byte expected.
- Again sender transmit one more 2048 bytes, which are acknowledged, but the advertised window size 0 (zero).
- Sender must stop until the application process has removed some data from the buffer.
- When the window is 0, the sender may not normally send segments because of two reasons :
  1. Urgent data may be sent.
  2. Sender may send a 1-byte segment to make the receiver reannounce the next byte excepted and window size.

Scanned with CamScanner

3.6.7.1 NAGLE Algorithm

- One byte at a time normally flows from the client to the server across an Rlogin connection. This generates 41-byte packet 20 bytes for the IP header and 20 bytes for TCP header and 1 byte of data. These small packets called as tinygrams. These tinygrams can add to congestion on WAN. Most LANs are not congested because tinygrams are not a problem on LANs.
- To solve the problem of congestion of WAN, the Nagle algorithm is used. The Nagle algorithm say that when TCP connection has outstanding data that has not yet been acknowledged, small segments cannot be sent until the outstanding data is acknowledged.
- Instead, small amounts of data are collected by TCP and sent in a single segment when the acknowledgement arrives.
- Nagle algorithm is self-clocking. The faster the ACKs come back, the faster the data is sent. But on a slow WAN, where it is desired to reduce the number of tinygrams, fewer segment are sent.
- Nagles algorithm is widely used by TCP implementations, but there are times when it is better to disable it. The example is the X window system server. Mouse movements must be delivered without delay to provide real time feedback for interactive users doing certain operations for bulk data flow.
- TCP uses a different form of flow control called a sliding window protocol. This sliding window protocol working is same as Data link layer sliding window protocol.

3.6.7.2 Silly Window Syndrome

- When large block of data is passed from sender but the receiver reads data one byte at a time. Receiving side, the TCP buffer is full and the sender know the condition. The interactive application reads one character from the TCP stream.
- Receiving TCP tells to the sender to send the only 1 byte. Sender send 1 byte. Now buffer is full and receiver send acknowledgement the 1-byte segment and set the window 0. This operation is continuous. Fig. 3.6.6 shows these steps.
- Nagle algorithm and Clark's solution to the silly window syndrome are complementary. Clark solution is to prevent the receiver from sending a window update for 1 byte. Instead it is forced to wait until it has a decent amount of space available. (See Fig. 3.6.6 on next page).

3.6.8 TCP Timer Management

- TCP manages four different timers for each connection.

Scanned with CamScanner

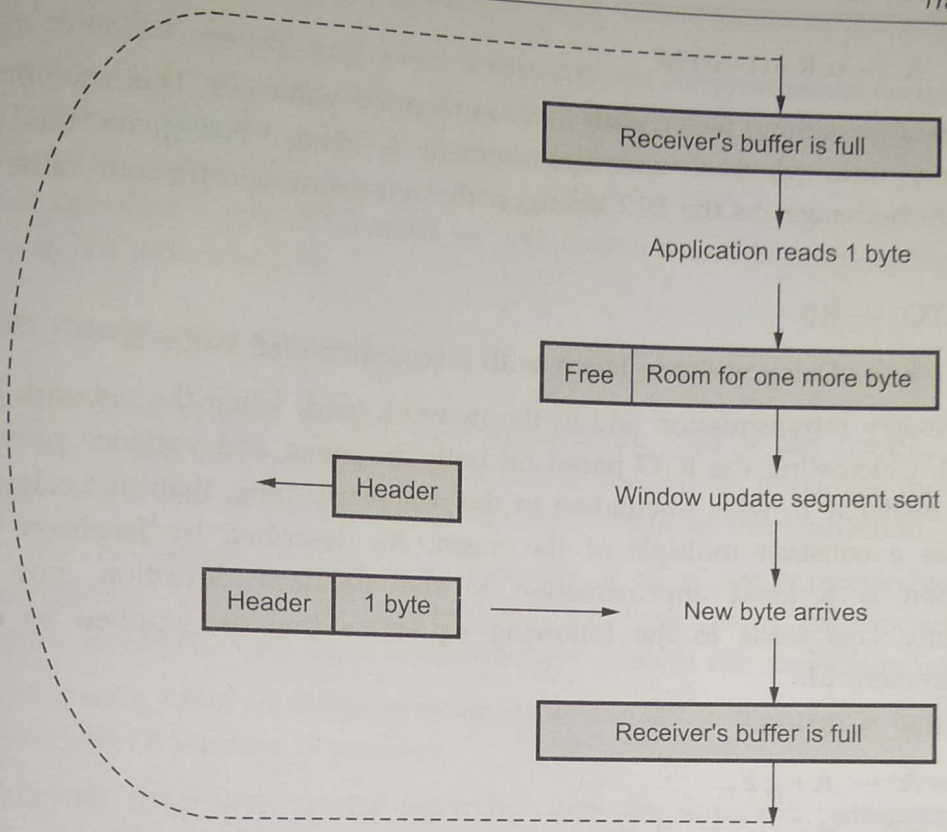


Fig. 3.6.6 Silly window syndrome

a) A **retransmission timer** is used when excepting an acknowledgement from the other end.

b) A **persist timer** keeps window size information flowing even if the other end closes its receiver window.

c) A **keep alive timer** detects when the other end on an otherwise idle connection crashes.

d) A 2 **maximum segment lifetime** (2 MSL) timer measures the time a connection has been in the TIME_WAIT state.

- Fundamental to TCP timeout and retransmission is the measurement of the Round-Trip Time (RTT) experienced on a given connection. The TCP must measure the RTT between sending byte with a particular sequence number and receiving an acknowledgement that covers that sequence number. For each connection, TCP maintains a variable RTT, that is the best current estimate of the round-trip time to the destination. When a segment is sent, a timer is started, both to see how long the acknowledgement takes and to trigger a retransmission if it takes too long.
- If the acknowledgement get back before the timer expires, TCP measures how long the acknowledgement took i.e. M. The original TCP specification had TCP update a smoothed RTT estimator (R) using low-pass filter.

Scanned with CamScanner

$$R \leftarrow \alpha R + (1-\alpha) M$$

where α is a smoothing factor with a recommended value 0.9. This smoothed RTT is updated every time when a new measurement is made. For given this smoothed estimator, which changes as the RTT changes, the retransmission timeout value (RTO) be set to

$$RTO = R\beta$$

where β = Delay variance factor with a recommended value 2.

- Unnecessary retransmission add to the network load, when the network is already loaded. Calculating the RTO based on both the mean and variance provide much better response to wide fluctuation in the round-trip time, than just calculating the RTO as a constant multiple of the mean. As described by Jacobson the mean deviation is a good approximation to the standard deviation, but easier to compute. This leads to the following equations that are applied to each RTT measurement M.

$$E_{rr} = M - A$$

$$A \leftarrow A + g E_{rr}$$

$$D \leftarrow D + h(|E_{rr}| - D)$$

$$RTO = A + 4D$$

where A = Smoothed RTT (estimator of average)

D = Smoothed mean deviation

E_{rr} = Difference between the measured value just obtained and the current RTT and estimator.

g = Gain

h = Gain of deviation

Both A and D are used to calculate the next Retransmission Time Out (RTO). The gain (g) is for the average and is set to 0.125 and h is set to 0.25. The larger gain for the deviation makes the RTO go up faster when the RTT changes.

a) Karn's algorithm :

- A problem occurs when a packet is retransmitted. If the packet is retransmitted, a timeout occurs, the RTO is backed off. The packet is retransmitted with the longer RTO and an acknowledgement is received. The received acknowledgement is whether the first transmission or the second. This is called the retransmission ambiguity problem.

Scanned with CamScanner

- Karn's algorithm specify that when a timeout and retransmission occur, we cannot update the RTT estimator when the acknowledgement for the retransmitted data finally arrives. Since the data was retransmitted, and the exponential back off has been applied to the RTO, we reuse this backed off RTO for the next transmission. Do not calculate a new RTO until an acknowledgement is received for a segment that was not retransmitted.

3.6.9 TCP Congestion Control

- When the load offered to any network is more than it can handle, congestion builds up. When a connection is established, the sender initializes the congestion window to the size of the maximum segment in use on the connection.
- When the congestion window is 'n' segments, if all 'n' are acknowledged on time, the congestion window is increased by the byte count corresponding to 'n' segments. In effects, each burst acknowledged doubles the congestion window.
- The congestion window keeps growing exponentially until either a timeout occurs or the receiver's window is reached.
- The Internet congestion control algorithm uses the threshold parameter which is initially 64 kB, in addition to the receiver and congestion windows. When a timeout occurs, the threshold is set to half of the current congestion window and the congestion window is reset to one maximum segment.

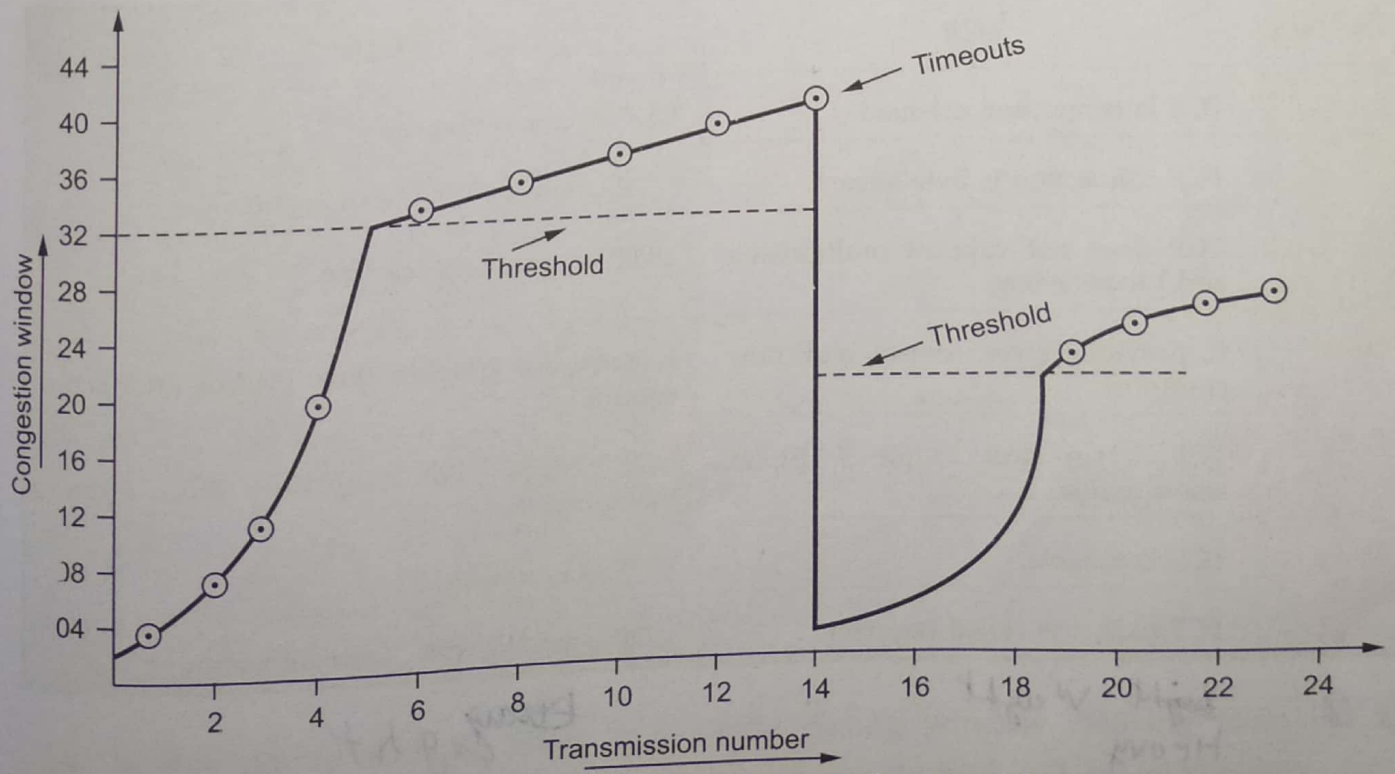


Fig. 3.6.7 Example of Internet congestion algorithm

Scanned with CamScanner

- Slow start is then used to determine what the network can handle, except that exponential growth stops when the threshold is hit. From that point on, successful transmissions grow the congestion window linearly instead of one per segment.
- Fig. 3.6.7 shows an example of the Internet congestion algorithm. (See Fig. 3.6.7 on previous page).
- The maximum segment size is 1024 bytes. Initially the congestion window was 64 kB, but a timeout occurred, so the threshold is set to 32 kB and the congestion window to 1 kB for transmission 0 (zero) here.
- The congestion window then grows exponentially until it hits the threshold (32 kB). Starting then, it grows linearly.
- Transmission 13 is unlucky and a timeout occurs. The threshold is set to half the current window, and slow start is initiated all over again.
- When the acknowledgements from transmission 14 start coming in, the first four each double the congestion window, but after that, growth becomes linear again.
- If no more timeouts occur, the congestion window will continue to grow up to the size of the receiver's window. At that point, it will stop growing and remain constant as long as there are no more timeouts and the receiver's window does not change size.

3.6.10 Comparison between TCP and UDP

Sr. No.	TCP	UDP
1	TCP is connection oriented.	UDP is connectionless.
2	TCP connection is byte stream.	UDP connection is message stream.
3	TCP does not support multicasting and broadcasting.	UDP support broadcasting.
4	It provides error control and flow control.	It does not provide flow control and error control.
5	TCP supports full duplex transmission.	UDP does not support full duplex transmission.
6	TCP is reliable.	UDP is unreliable.
7	TCP packet is called segment.	UDP packet is called user datagram.
8	~~Hight~~ Weight Heavy	~~Heavy~~ Light

Scanned with CamScanner

3.6.11 Sliding Window and Flow Control

- Flow control is a technique whose primary purpose is to properly match the transmission rate of sender to that of the receiver and the network. It is important for the transmission to be at a high enough rates to ensure good performance, but also to protect against overwhelming the network or receiving host.
- Flow control is not the same as congestion control. Congestion control is primarily concerned with a sustained overload of network intermediate devices such as IP routers.
- TCP uses the window field, as the primary means for flow control. During the data transfer phase, the window field is used to adjust the rate of flow of the byte stream between communicating TCP's.
- Fig. 3.6.8 below illustrates the concept of the sliding window.

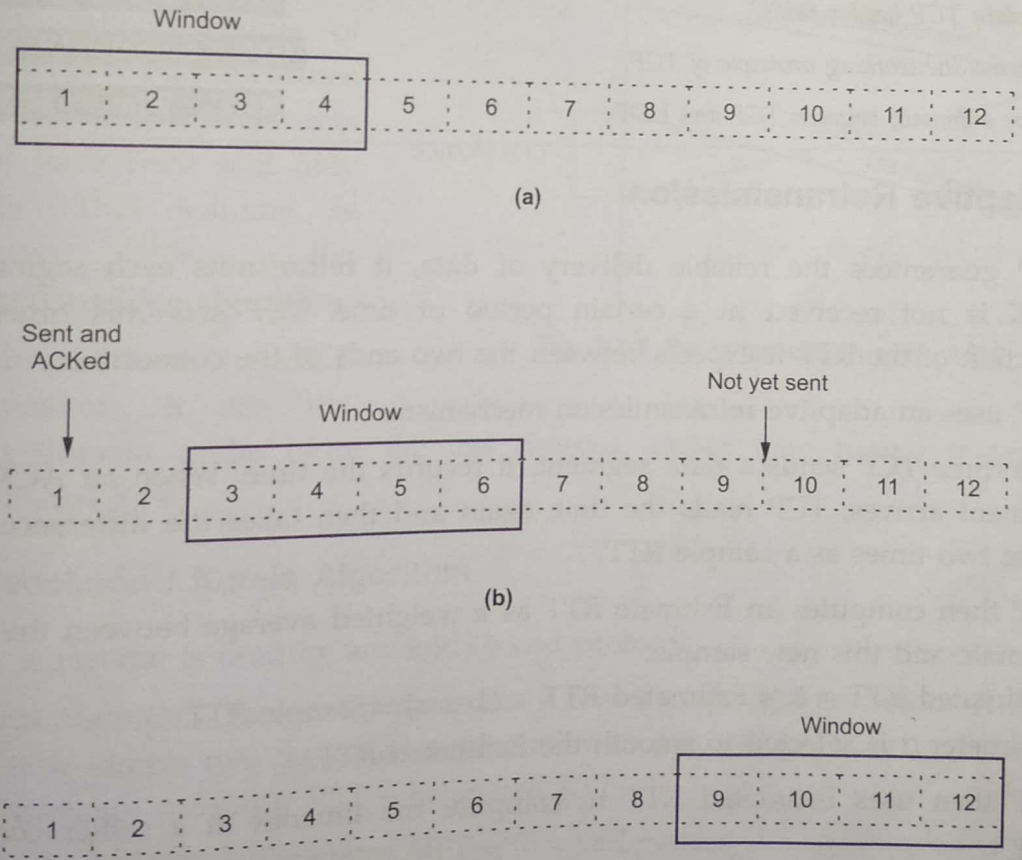


Fig. 3.6.8 Sliding window

- In this simple example, there is a 4-byte sliding window. Moving from left to right, the window "slides" as bytes in the stream are sent and acknowledged.

Scanned with CamScanner

- A TCP sliding window provides more efficient use of network bandwidth than Positive Acknowledgement and Retransmission (PAR) because it enables hosts to send multiple bytes or packets before waiting for an acknowledgement.
- In TCP, the receiver specifies the current window size in every packet. Because TCP provides a byte stream connection, window sizes are expressed in bytes. This means that a window is the number of data bytes that the sender is allowed to send before waiting for an acknowledgement. Initial window sizes are indicated at connection setup, but might vary through out the data transfer to provide flow control. A window size of zero, for instance means, send no data.

University Questions

1. *Compare : TCP and UDP.*	**GTU : Dec.-11, Summer-14, Marks 7**
2. *Explain TCP header fields.*	**GTU : Winter-12, Marks 7**
3. *Discuss the working principle of TCP.*	**GTU : Summer-13, Marks 7**
4. *Give difference between TCP and UDP.*	**GTU : Winter-14, Marks 7**

3.7 Adaptive Retransmission

- TCP guarantees the reliable delivery of data, it retransmits each segment if an ACK is not received in a certain period of time. TCP sets this timeout as a function of the RTT it expects between the two ends of the connection.
- TCP uses an adaptive retransmission mechanism.
- Everytime TCP sends a data segment, it records the time. When an ACK for that segment arrives, TCP reads the time again and then takes the difference between these two times as a sample RTT.
- TCP then computes an Estimate RTT as a weighted average between the previous estimate and this new sample.

  $\text{Estimated RTT} = \alpha \times \text{Estimated RTT} + (1 - \alpha) \times \text{Sample RTT}$
- Parameter α is selected to smooth the Estimated RTT.
- TCP then uses Estimated RTT to compute the timeout in a rather conservative way :

  $\text{Timeout} = 2 \times \text{Estimated RTT}$

3.7.1 Karn / Partridge Algorithm

- The problem of the above algorithm is that, an ACK does not really acknowledge a transmission; it actually acknowledges the receipt of data. Fig. 3.7.1 shows associating the ACK with retransmission.

Scanned with CamScanner

- If you assume that the ACK is for the original transmission but it was really for the second, then the **sample RTT** is too large, which is shown in Fig. 3.7.1.
- If you assume that the ACK is for the second transmission but it was actually for the first, then the sample RTT is too small.

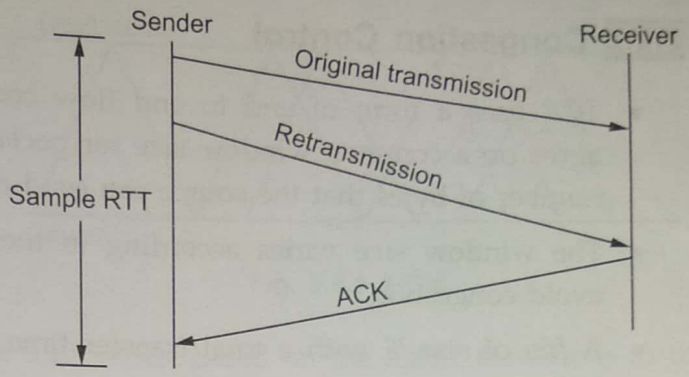


Fig. 3.7.1 ACK with original

- **Solution to the above problem**
- Whenever TCP retransmits a segment, it stops taking sample of the RTT; it only measures sample RTT for segments that have been sent only once. This solution is known as the **Karn/Partridge algorithm.**

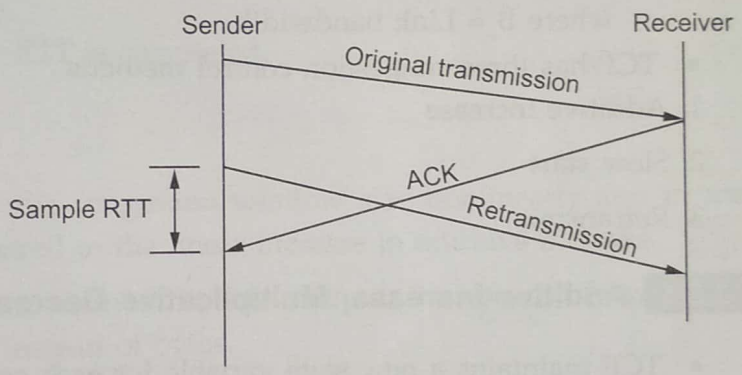


Fig. 3.7.2 Retransmission with ACK

- Each time TCP retransmits, it sets the next timeout to be twice the last timeout, rather than basing it on the last Estimated RTT.

3.7.2 Jacobson / Karels Algorithm

- This algorithm is used by any end to end protocol.
- In this algorithm, the sender measures a new sample RTT as before. It then folds this new sample into the timeout calculation as follows :

  Difference = Sample RTT – Estimated RTT

  Estimated RTT = Estimated RTT + ($\delta \times$ Difference)

  Deviation = Deviation + (δ | Difference | – Deviation)

  where δ is fraction between 0 and 1.
- TCP then computes the timeout value as a function and both Estimated RTT and deviation as follows :
- Timeout = $\mu \times$ Estimated RTT + $\phi \times$ Deviation

Scanned with CamScanner

3.8 Congestion Control

GTU : Dec.-10, Winter-12,14

- TCP uses a form of end to end flow control. Both the sender and the receiver agree on a common window size for packet flow. The window size represents the number of bytes that the source can send at a time.
- The window size varies according to the condition of traffic in the network to avoid congestion.
- A file of size 'f' with a total transfer time of 'Δ' on a TCP connection results in a TCP transfer throughput (r). i.e.

$$r = \frac{f}{\Delta}$$

- Bandwidth utilization $(\rho_u) = \frac{r}{B}$

  where B = Link bandwidth
- TCP has three congestion control methods

1. Additive increase
2. Slow start
3. Retransmit

3.8.1 Additive Increase, Multiplicative Decrease Control (AIMD)

- TCP maintains a new state variable for each connection, called congestion window, which is used by the source to limit how much data it is allowed to have in transit at a given time. The congestion window represents the amount of data, in bytes.
- AIMD performs a slow increase in the congestion window size when the congestion in the network decreases and a fast drop in the window size when congestion increases.
- Let W_m be the maximum window size, in bytes, representing the maximum amount of unacknowledged data that a sender is allowed to send.
- Let W_a be the advertised window sent by the receiver, based on its buffer size.
- TCP's effective window is revised as follows :

  Max window = MIN (Congestion window, Advertised window)

  Effective window = Max window – (Last byte sent – Last byte ACKed)
- **Max window** replaces **Advertised window** in the calculation of **Effective window.**

Scanned with CamScanner

- Fig. 3.8.1 shows the additive increase control for TCP congestion control.

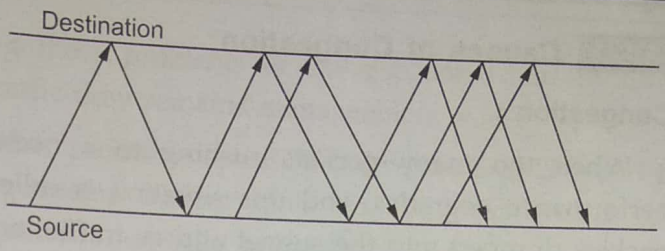


Fig. 3.8.1 AIDM

- The challenge in TCP congestion control is for the source node to find a right value for the congestion window. The congestion window size varies, based on the traffic conditions in the network. TCP watches for timeout as a sign of congestion.
- TCP technique requires that the timeout values be set properly. Two important factors in setting timeouts follow.

1. Average Round Trip Times (RTTs) and RTT standard deviation based to set timeouts.
2. RTTs are sampled once every RTT is completed.

3.8.2 Slow Start Method

- Slow start method increases the congestion window size nonlinearly and in most cases exponentially, as compared to the linear increase in additive increase.
- Fig. 3.8.2 shows the slow start method. In this method, the congestion window is again interpreted in packets instead of bytes.

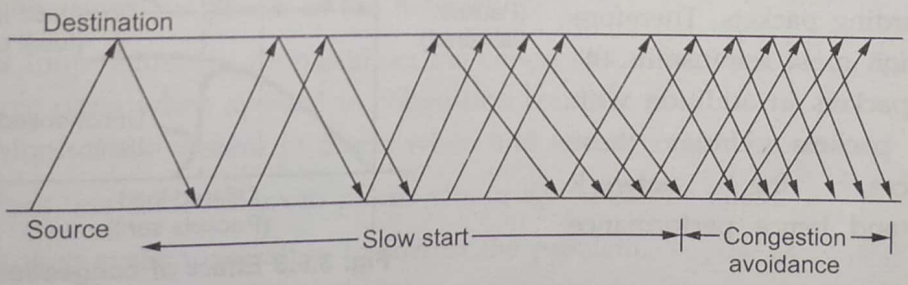


Fig. 3.8.2 Slow start

- Source initially sets the congestion window to one packet. When its corresponding acknowledgement arrives, the source sets the congestion window to two packets. Now, the source sends two packets. On receiving the two corresponding acknowledgements, TCP sets the congestion window size to 4. Thus, the number of packets in transit doubles for each round-trip time.
- The slow start method is normally used

1. Just after a TCP connection is set up.
2. When a source is blocked, waiting for a timeout.

Scanned with CamScanner

3.8.3 Causes of Congestion

Congestion :

When too many packets rushing to a node or a part of network, the network performance degrades, and this situation is called as **congestion**. When the number of packets dumped into the subnet and as traffic increases the network is no longer able to cope and design losing packets at very high traffic, performance collapses completely and almost no packets are delivered.

Congestion control :

Congestion control is a process of maintaining the number of packets in a network below a certain level at which performance falls off. Congestion control makes sure that subnet is able to carry the offered traffic. So congestion control is different process than flow control.

Effects on congestion :

The effect on congestion on throughput of a network is shown in Fig. 3.8.3.

Ideally as offered load increases, throughput also increases. Practically throughput drops with increase in offered load because the buffers at each node's are full and starts discarding packets. Therefore source station must retransmits the discarded packets in addition with the new packets. Under these circumstances, the network utilization and hence performance falls off.

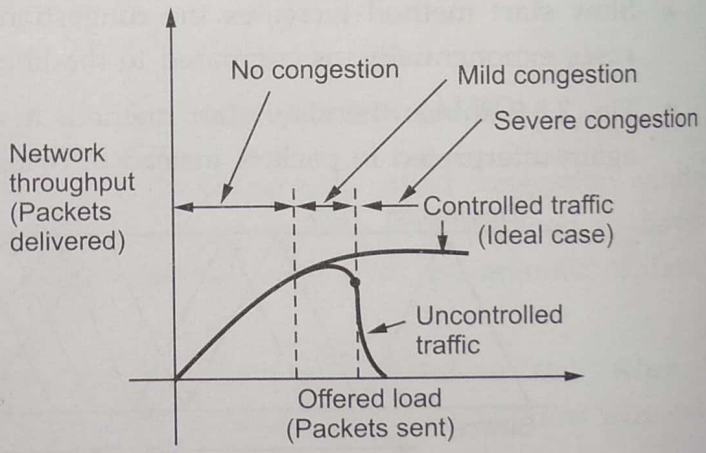


Fig. 3.8.3 Effect of congestion

Causes of congestion :

1) At any node if all of a sudden streams of packets begin arriving on three or more links and all needed the same output link, there is a queue of packet for outgoing channel. If the rate at which packets arrive and queue up exceeds the rate at which packets can be transmitted, the queue size grows, the delay experienced by a packet goes to infinity. If there is insufficient memory to hold all the packets, some packets will be lost. When such a saturation point is reached, one of the two general strategies can be adopted. The first such strategy is to simply discard any incoming packet for which there is no available space in buffer. The alternative is

Scanned with CamScanner

for the node that is experiencing these problems to exercise some sort of flow control over its adjacent so that traffic flow remains manageable.

2) The other cause of congestion is slow processor speed. If the router's CPU speed is slow and performing tasks like queueing buffers, table updating etc. queues are built up, even though the line capacity is not fully utilized.

3) The bandwidth of the links are also important in congestion. The links to be used must be of high bandwidth to avoid the congestion.

4) Any mismatch between parts of the system also cause congestion, upgrading the processor's speed but not changing the links or viceversa will cause the congestion.

5) When the arrival of the packets are not uniform i.e. when the traffic is bursty.

3.8.4 General Principles of Congestion Control

- General principles of congestion control can be divided into two groups i.e. open loop and closed loop.
- **Open loop** solutions attempt to solve the problem by good design, in essence, to make sure it does not occur in the first place. Once the system is up and running, midcourse corrections are not made.
- Tools for doing open loop control include deciding when to accept new traffic, deciding when to discard packets and which ones, and making scheduling decisions at various points in the network.
- **Closed loop** solutions are based on the concept of a feedback loop. This approach has three parts when applied to congestion control :
  1. Monitor the system to detect when and where congestion occurs.
  2. Pass this information to places where action can be taken.
  3. Adjust system operation to correct the problem.
- Metrics to monitor the subnet for congestion are : the average queue lengths, percentage of all packets discarded for lack of buffer space, average packet delay, the number of packets that time out and are retransmitted etc.

3.8.5 Congestion Prevention Policies

Data link layer, Network layer and Transport layer have some congestion prevention policies. These are given below :

1) Data link layer

a) Flow control policy
b) Acknowledgement policy
c) Retransmission policy
d) Out of order caching policy

Scanned with CamScanner

2) Network layer policies

a) Routing algorithm

b) Packet queueing and service policy

c) Packet lifetime management policy

d) Packet discard policy

e) Virtual circuit versus datagram inside the subnet.

3) Transport layer policies

a) Flow control policy

b) Acknowledgment policy

c) Retransmission policy

d) Out of order caching policy

e) Timeout determination.

3.8.6 Differences between Flow Control and Congestion Control

1) Flow control is done by server machine or sender machine whereas congestio control is done by router.

2) Buffering is used in flow control. Buffering is not possible in congestion control.

3) Flow control cannot block the bandwidth of medium. Congestion control block th bandwidth of medium.

4) In flow control, packet is lost in between one sender and receiver only. I congestion control, other users packet is also lost.

5) Flow control affect less on network performace. Congestion control affects the network performance.

University Questions

1. *Give difference between : Flow control versus congestion control.* **GTU : Winter-14, Marks 14**
2. *What are the various congestion prevention policies at datalink, network and transport layer of the OSI ?* **GTU : Dec.-10, Marks 4**
3. *Discuss and list the congestion prevention policies at data link, network and transport layers that can affect the congestion.* **GTU : Winter-12, Marks 7**

Scanned with CamScanner

3.9 Congestion Avoidance

GTU : Winter-12,14, May-12

- A congestion avoidance scheme allows a network to operate in the region of low delay and high throughput. It is a prevention mechanism while congestion control is a recovery mechanism.

3.9.1 DECbit Scheme

- DECbit means destination experiencing congestion bit.
- DECbit method is developed on the Digital Network Architecture (DNA). It split the responsibility between routers and end hosts. It is router-based congestion avoidance method.
- Uses a *congestion-indication bit* in packet header to provide feedback about congestion. Upon packet arrival, the average queue length is calculated for last (busy + idle) period plus current busy period. When the average queue length exceeds one, the router sets the congestion-indicator bit in arriving packet's header.
- If at least half of packets in source's last window have the bit set, decrease the congestion window exponentially.
- Queue length is counted over last busy period + idle + current busy period.
- Source machine adjust the packet flow rate. Source machine maintains a congestion window. It observes how many packets have the congestion bit set to 1 in the last window worth of packets.
- If less than 50 % of the ACKs have the DECbit set, then increase the window by 1 packet, otherwise, set the window to 0.875 times the original value.

3.9.2 RED

- RED stands for Random Early Detection. The main idea is to provide congestion control at the router for TCP flows. RED is based on DECbit, and was designed to work well with TCP.
- RED implicitly notifies sender by dropping packets. Packet dropping probability is increased as the average

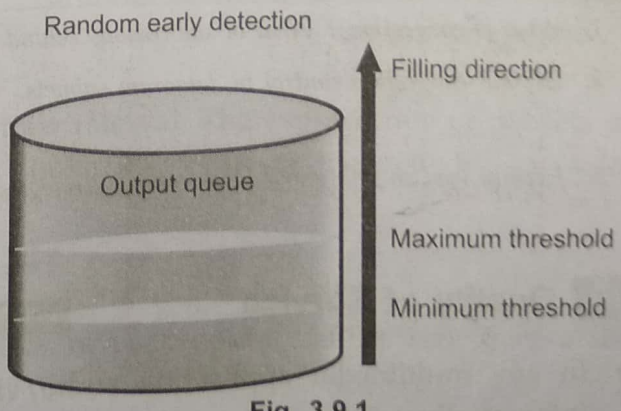


Fig. 3.9.1

Scanned with CamScanner

queue length increases. The moving average of the queue length is used so as to detect long term congestion, yet allow short term bursts to arrive.

- Properties of RED :
  1. Drops packets before queue is full, in the hope of reducing the rates of some flows.
  2. Drops packet for each flow roughly in proportion to its rate.
  3. Drops are spaced out in time.
  4. Because it uses average queue length, RED is tolerant of bursts.
  5. Random drops hopefully desynchronize TCP sources.
- RED calculates the average queue length using a weighted running average. Following formula is used for calculating average queue length

Average length = (1 – Weight) × Average length + Weight × Sample Length

Where sample length is queue length each time a packet arrives. The weight parameter is in between 0 and 1 i.e. **0 < Weight < 1.**

- RED uses a packet drop profile to handle packet discarding. This profile defines a set of dropping probabilities according to the level of queue occupancy. A minimum and a maximum threshold are defined as shown in the Fig. 3.9.1.
  1. If the queue occupancy lies beneath the minimum threshold, then packet drop does not occur.
  2. If the queue occupancy is somewhere between minimum and maximum thresholds then packets are dropped according to the configured drop probability.
  3. When the queue occupancy crosses maximum threshold, then all new packets attempting to enter the queue are discarded.

University Questions

1. *What is congestion ? What or the reasons behind congestion ?* **GTU : Winter-14, Marks 4**
2. *Explain congestion control in datagram subnets.* **GTU : May-12, Marks 4**
3. *Explain load shedding and jitter control strategies to handle the congestion.* **GTU : Winter-12, Marks 7**

3.10 Quality of Service

GTU : Dec.-10, June-11, May-12, Winter-13, 14

- In any multimedia application audio/video packets are delay sensitive but by internet all packets are treated equally i.e. QoS offered is same for all applications. This causes congestion in traffic followed by delay and loss of packets.

Scanned with CamScanner

- Analyzing varying network scenarios principles of Quality of Services (QoS) needed for multimedia applications are derived.

Principle 1 :

Packet marking allows a router to distinguish among packets belonging to different classes of traffic.

Modified principle 1 :

Packet classification allows a router to distinguish among packets belonging to different classes of traffic.

Principle 2 :

A degree of isolation is desirable among traffic flows, so that one flow is not adversely affected by another misbehaving flow.

Principle 3 :

For isolating flows, it is desired to use resources like BW and buffers as efficiently as possible.

Principle 4 :

A call admission process is needed where flows declare their QoS requirement.

3.10.1 Policing

- Policing is the regulation of the rate at which packet flow is injected into the network.

Criteria for policing

- Three important policing criteria are identified, these are :
  1. Average rate 2. Peak rate 3. Burst size

1. Average rate

Average rate is defined as packets per time interval. The average rate of packets in a network can be limited as a policy. This limits the traffic in the network for a long period of time.

2. Peak rate

Peak rate is defined as maximum number of packets that can be sent over a short period of time over a network.

Scanned with CamScanner

3. Burst size

Burst size is the maximum number of packets that can be sent into the network over a extremely short interval of time.

3.10.2 Integrated Services

- Integrated service is a framework to provide guaranteed Quality of Service (QoS) to individual application sessions.
- A call step process involves following steps :
  1. Traffic characterization and specification of desired QoS.
  2. Signalling for call setup.
  3. Pre element call admission.
- The intserv architecture defines two major classes of service.
  a) Guaranteed service.
  b) Controlled load service.
- The guaranteed QoS specification specifies queuing delays that a packet experience in a router.
- The actual delay is subject to the peak rate limitation of the input link and variations in the packet is transmission time.
- A session with controlled-load service will receive same QoS as from unloaded network, i.e. all the packets will successfully pass through router without loss and will experience a zero queuing delay.
- The controlled-load network service is developed mainly for real-time multimedia application over Internet.

3.10.2.1 Traffic Shaping

- Traffic shaping is about regulating the average rate of data transmission.
- Traffic shaping smooths out the traffic on the server, rather than on the client side.
- Monitoring a traffic flow is called traffic policing. Agreeing to a traffic shape and policing it afterward are easier with virtual circuit subnets than with datagram subnets.
- Traffic shaping is an open loop method of congestion control.
- Two types of algorithm are used for traffic shaping.
  1. Leaky bucket algorithm 2. Token bucket algorithm

1. Leaky bucket algorithm

- Leaky bucket i.e. a bucket with a small hole in the bottom is used to store the water. The outflow from hole is at constant rate and irrespective of rate of

Scanned with CamScanner

entering water. Once the bucket is full, any additional water entering it spills over the sides and is lost.

- The same idea can be applied to packets. This is similar to a single server queueing system with constant service time.
- Each host is connected to network with a finite internal queue. The host is allowed to put one packet per second on to the network. If a packet arrives at the queue when it is full, the packet is discarded. This mechanism turn an unregulated traffic of the host regulated traffic on the network. Thus bursty traffic is smoothen and chances of congestion is reduced. Fig. 3.10.1 illustrates this algorithm.

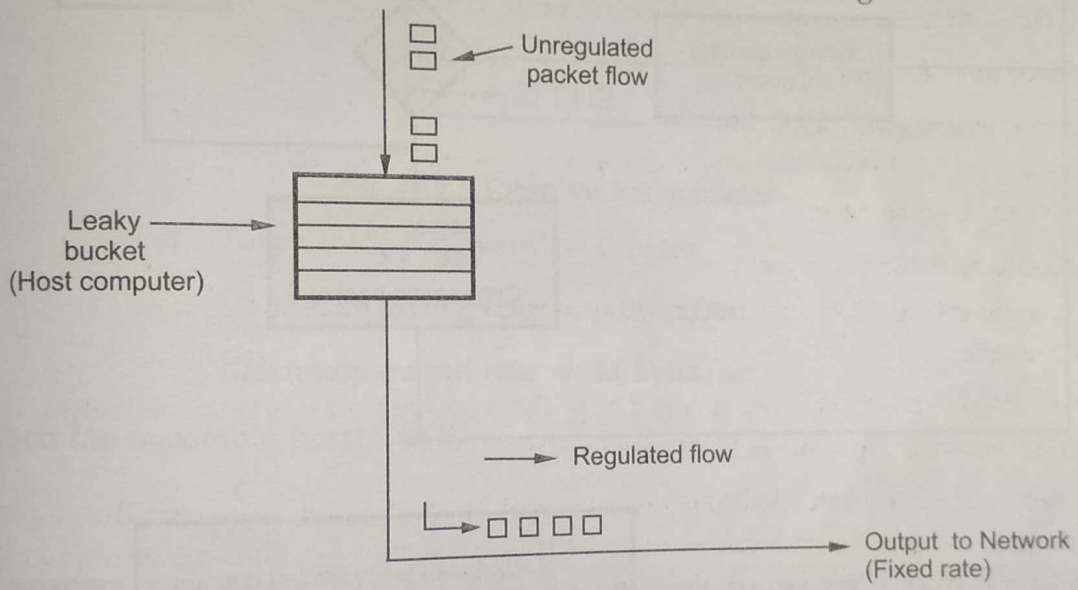


Fig. 3.10.1 Leaky bucket regulator

- A leaky bucket regulator allows to control the average rate, largest burst from a source. A leaky bucket regulator has both a packet bucket and a data buffer
- The main drawback of leaky bucket algorithm is that its output pattern cannot be modified i.e. if the bursty traffic arrives the output should speed up, so that no packets will be lost.
- Fig. 3.10.2 shows the leaky bucket algorithm that can be used to police the traffic flow. (See Fig. 3.10.2 on next page).
- At the arrival of the first packet, the content of the bucket X is set to zero and the last conforming time is set to the arrival time of the first packet. The depth of the bucket is L+I where L typically depends on the traffic burstiness.

2. Token bucket algorithm

- Token bucket algorithm eliminates drawback of leaky bucket algorithm. In this, the leaky bucket holds tokens. These tokens are generated by a clock at the rate of

Scanned with CamScanner

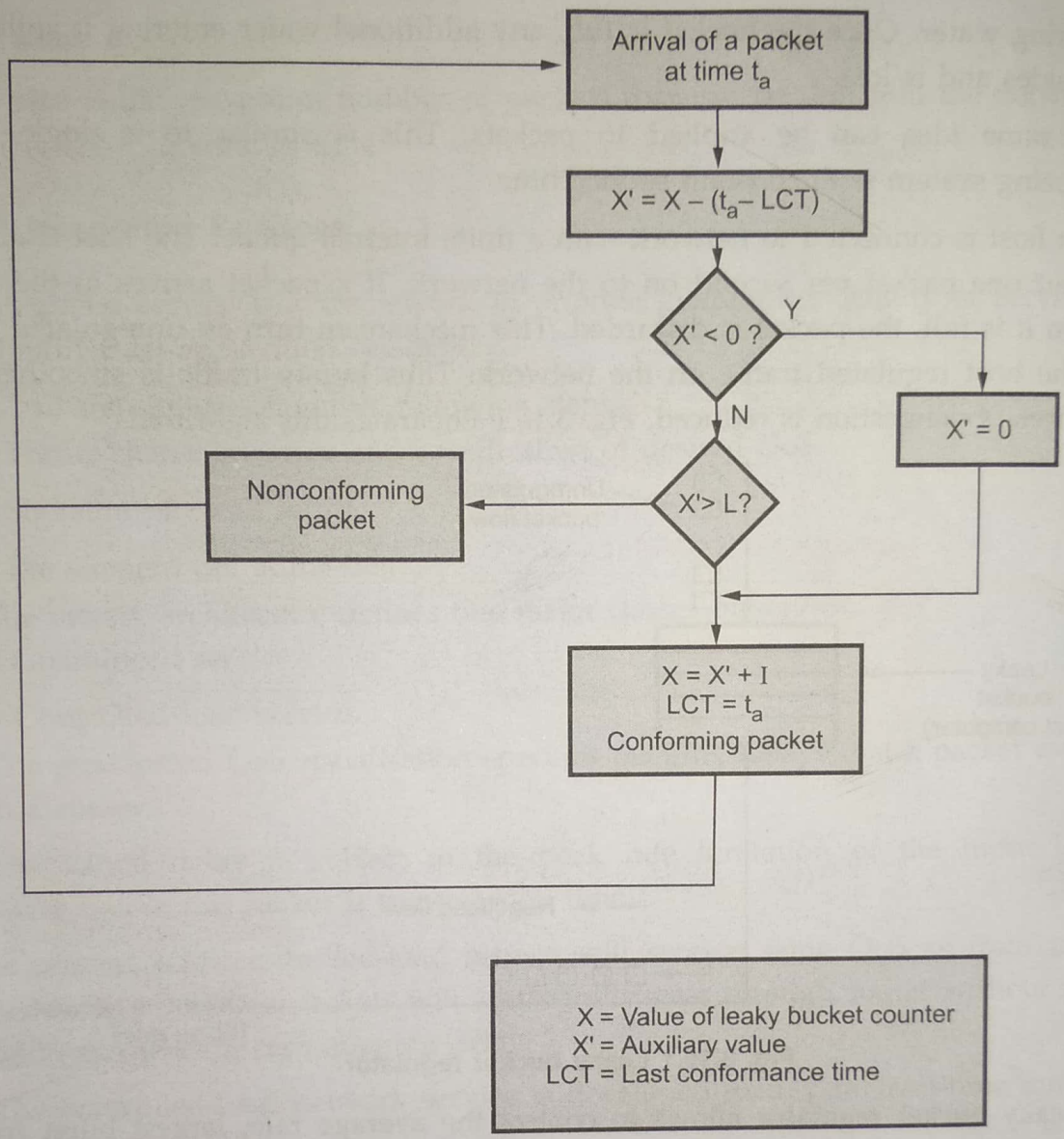


Fig. 3.10.2 Leaky bucket algorithm for policing

one token for every ΔT sec. In token bucket bursts of upto n packets can be sent at once, which gives faster response to sudden bursts of input.

- The regulator collects tokens in a bucket, which fills-up at steady drip rate by packets. When a packet arrives at the regulator, the regulator sends the packet if the bucket has enough tokens. Otherwise, the packet waits either until the buckets has enough tokens. If the bucket is already fill of tokens, incoming tokens overflow and are not available to future packets. Thus, at anytime, the largest burst a source can send into the network is roughly proportional to the size of the leaky bucket.
- The regulator delays a packet if does not have sufficient number of tokens for transmission. A counter keeps track of tokens, the counter is incremented by one every ΔT and decremented by one whenever a packet is sent. When the counter

Scanned with CamScanner

hits zero, no packets may be sent. Smoother traffic can be obtained by putting a leaky bucket after the token bucket.

- Fig. 3.10.3 illustrates token bucket regulator.

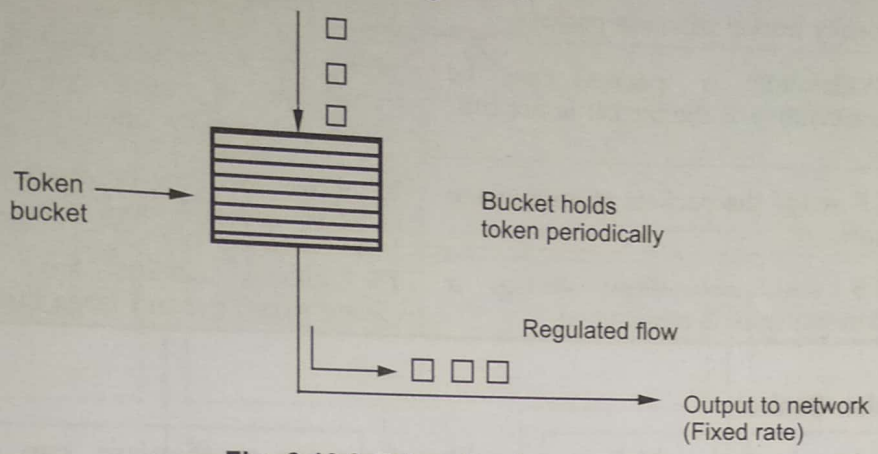


Fig. 3.10.3 Token bucket regulator

Let Token bucket capacity = C bytes

Token arrival rate = ρ bytes/sec

Maximum output rate = M bytes/sec

Then the maximum burst rate S,

$$S = \frac{C}{M-\rho}$$

Example 3.10.1 *A computer on 6 Mbps network is regulated by token bucket. The token bucket is filled at the rate of 1 Mbps. It is initially filled to capacity with 8 megabits. How long can the computer transmit at the fill 6 Mbps ?*

Solution : $\rho = 1$ Mbps

$C = 8$ Mb

$M = 6$ Mbps

$$S = \frac{C}{M-\rho}$$

$$S = \frac{8}{6-1}$$

$S = 1.6$ sec

$S \approx 2$ sec

Scanned with CamScanner

Comparison between leaky bucket and token bucket

Sr. No.	Leaky Bucket (LB)	Token Bucket (TB)
1.	Leaky bucket discards packets.	Token bucket discards tokens.
2.	With LB, a packet can be transmitted if the bucket is not full.	With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
3.	LB sends the packets at an average rate.	TB allows for large bursts to be sent faster by speeding up the output.
4.	LB does not allow saving, a constant rate is maintained.	TB allows saving up tokens (permissions) to send large bursts.

3.10.2.2 Admission Control

- An admission control, which is a quality of service mechanism, can also prevent congestion in virutal circuit networks. Admission control in ATM operates at the connection level and is therefore called connection admission control.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network. A router can deny establishing a virtual circuit connection if there is congestion in the network. A source initiating a new flow must first obtain permission from an admission control entity that decides whether the flow should be accepted or rejected.
- The QoS may be expressed interms of maximum delay, loss probability, delay variance, or other performance parameters. If the quality of service of the new flow can be satisfied without violating QoS of existing flows, the flow is accepted; otherwise, the flow is rejected.

3.10.2.3 RSVP (ReSource reserVation Protocol)

- RSVP is a signalling protocol used to reserve resources in the Internet. RSVP is a bandwidth reservation protocol.
- RSVP protocol allows applications to reserve bandwidth for their data flows.

Characteristics of RSVP

1. It provides reservations for bandwidth in multicast trees.
2. RSVP is receiver-oriented i.e. receiver intitiates this protocol for resource reservation.

- To get better reception and eliminate congestion any of the receivers in a group can send a reservation message up the tree to the sender. The message is propagated using the reverse path forwarding algorithm. Example of such a reservation is shown in the Fig. 3.10.4.

Scanned with CamScanner

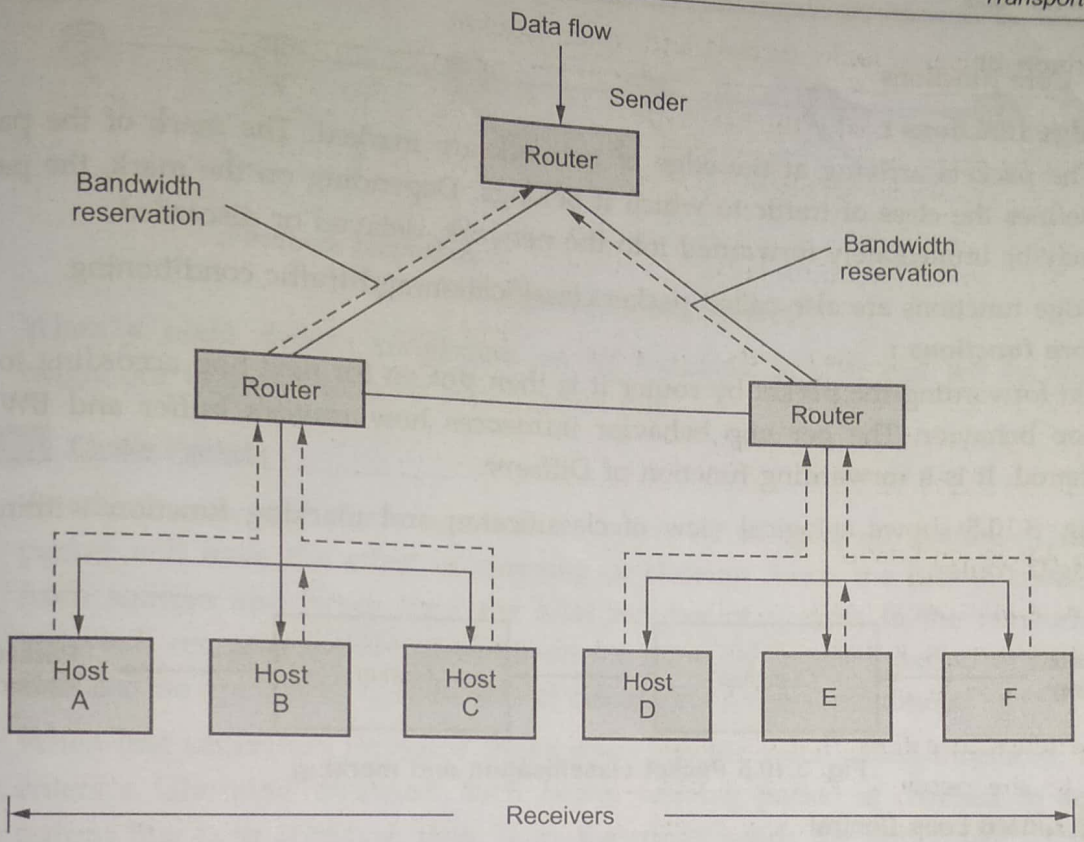


Fig. 3.10.4 RSVP

- Fig. 3.10.4 shows multicast spanning tree with data flowing from the top of the tree to hosts at the bottom of tree. The data originates from the sender and reservation message originates from the receiver. The upstream reservation messages from host can be merged with other reservation message.

3.10.3 Differentiated Services/QoS

- Some difficulties associated with RSVP and Intserv model are :

1. **Scalability :** Reservation requests cause significatnt overhead in large networks.
2. **Flexible service models :** The pre-specified service classes make them vulnerable to router crashes.

- The Differentiated Services (DiffServ) group has developed an architecture for providing scalable and flexible service differentiation. This architecture has the ability to handle different classes of traffic in different way within the internet. This approach is known as class-based QoS.

3.10.3.1 Functional Elements of Differentiated Service

- The Differentiated Services (DiffServes) architecture consists of two sets of functional elements :

1. Edge functions

Scanned with CamScanner

2. Core functions

1. Edge functions :

- The packets arriving at the edge of network are marked. The mark of the packet defines the class of traffic to which it belongs. Depending on the mark, the packet may be immediately forwarded into the network, delayed or discarded.
- Edge functions are also called packet classification and traffic conditioning.

2. Core functions :

- On forwarding the packet by router it is then put on for next hop according to per hop behavior. The per hop behavior influences how router's buffer and BW are shared. It is a forwarding function of Diffserv.
- Fig. 3.10.5 shows a logical view of classificaton and marking function within the edge7 router.

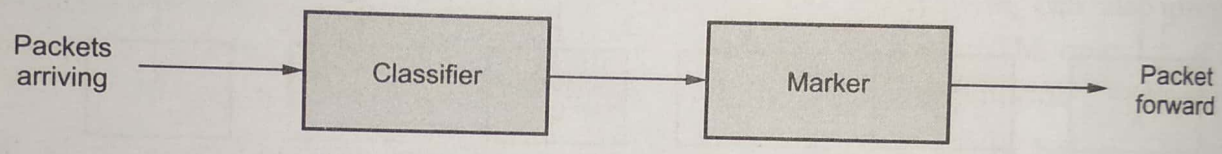


Fig. 3.10.5 Packet classification and marking

3.10.3.2 Closed Loop Control

- Closed loop control try to alleviate congestion after it happens.

End-to-End versus Hop-by-Hop

- In **end-to-end closed loop control**, the feedback information about state of the network is propagated back to the source that can regulate the packet flow rate. Fig. 3.10.6 shows end-to-end closed loop control.

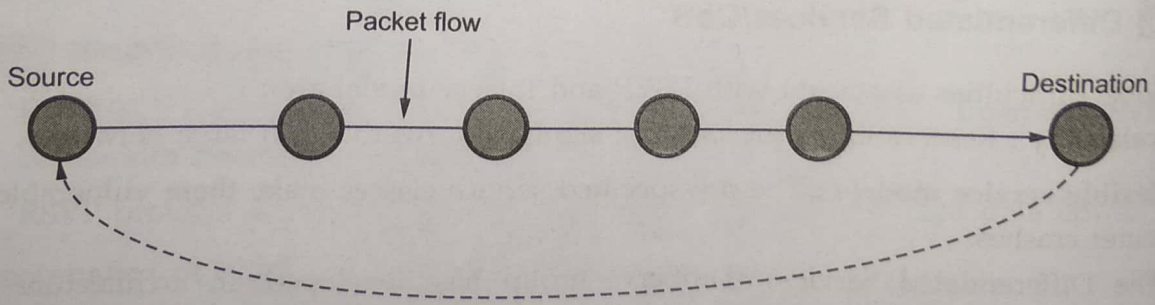


Fig. 3.10.6 End-to-end closed loop control

- The feedback information may be forwarded directly by a node that detects congestion, or it may be forwarded to the destination first which then relays the information to the source.
- With hop-by-hop closed loop control, the state of the network is propagated to the upstream node. Fig. 3.10.7 shows hop-by-hop control.

Scanned with CamScanner

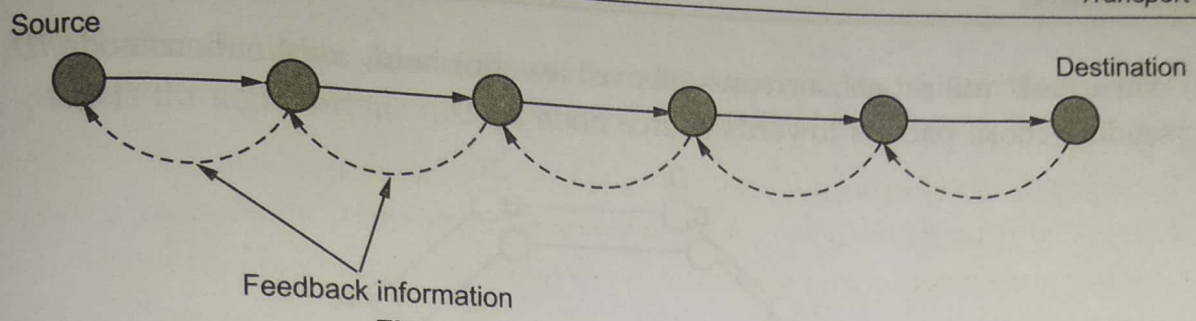


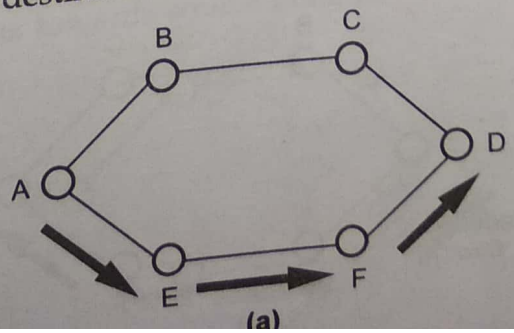
Fig. 3.10.7 Hop-by-hop loop control

- When a node detects congestion on its outgoing link, it can tell its upstream neighbour to slow down its transmission rate.

3.10.3.3 Choke Packets

- Another mechanism for congestion control is by using choke packets. This choke packet will have the effect of stopping or slowing down the rate of transmission from sources and hence limit the total number of packets in the networks. This approach requires additional traffic on the network during a period of congestion. This can be applicable to both virtual circuit and datagram subnets.
- When line utilization increases above some specific value called threshold, the line enters a 'alarming' situation. Each newly arriving packet is checked to see if its output line is in alarming state. If so, the router sends the said choke packet back to the source. This choke packet contains the destination address, so the source will not generate any more packets along the path.
- The traffic is reduced by adjusting parameters window size or leaky bucket output rate. Typically, the first choke packet causes the data rate to 50 % of its previous value the choke packet reduces the traffic to 25 % and so on.
- Congestion control using choke packets can be done by two ways. In first type the choke packet affects only source and in the second type the choke packet affects each hop it passes through. Fig. 3.10.8 shows choke packet that affects only the source.
- Fig. 3.10.8 shows a choke packet that affects each hop it passes through.

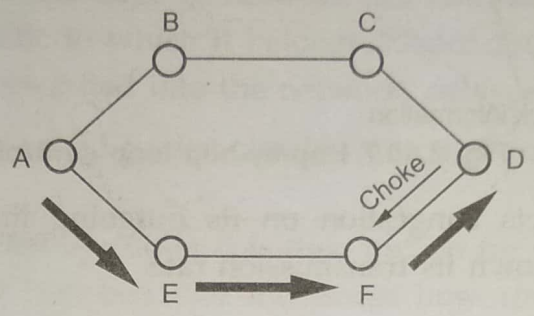
i) A subnet with six nodes A, B, C, D, E and F is shown in Fig. 3.10.8 (a). Here the source node is A and destination node is D.



(a)

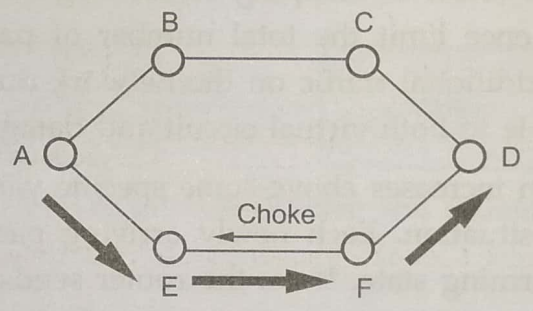
Scanned with CamScanner

ii) When link utilization increases above its threshold, destination node D starts sending choke packets towards source node A.



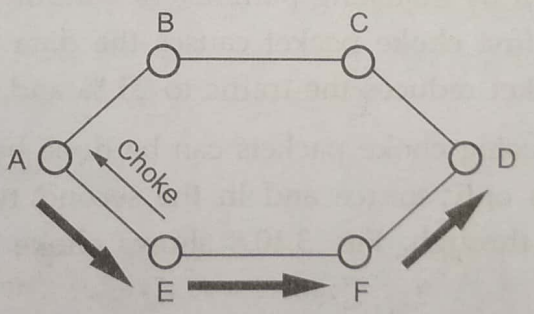
(b)

iii) The choke packets travels through the shortest or same path as that of packets.



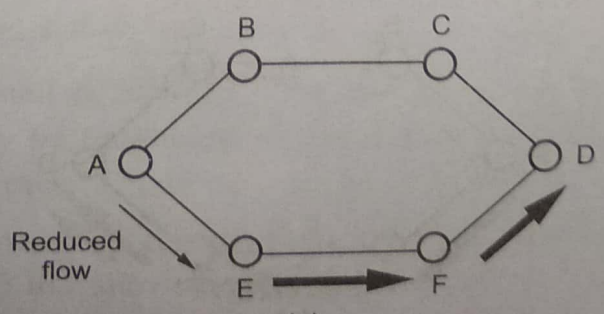
(c)

iv) The choke packet reaches to the source node A.



(d)

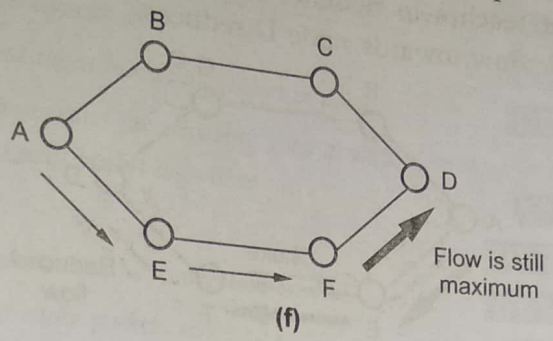
v) After receiving first chock packet source node A reduces its flow towards destination.



(e)

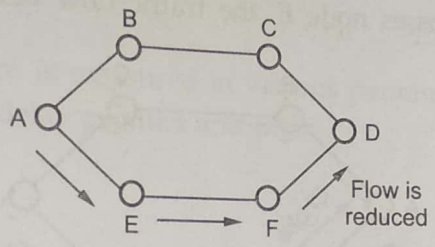
Scanned with CamScanner

vi) The reduced packet flow follows the same reversed path i.e. the path of choke packets through various nodes.



(f)

vii) The reduced flow reaches to destination node D.

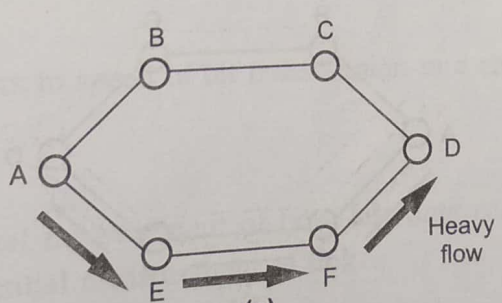


(g)

Fig. 3.10.8 A choke packet that affects only the source

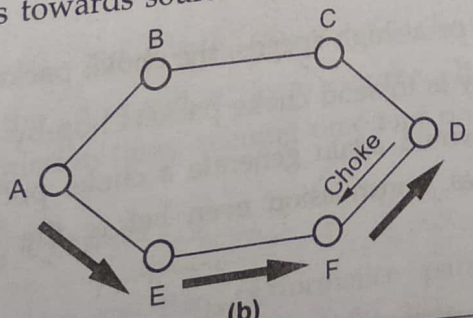
Fig. 3.10.9 shows a choke packet that affects each hop it passes through.

i) For the same subnet having nodes A, B, C, D, E and F source node and destination node D.



(a)

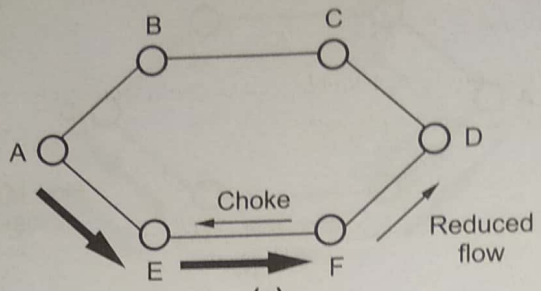
ii) When link utilization increased above its threshold destination node D starts sending choke packets towards source node A.



(b)

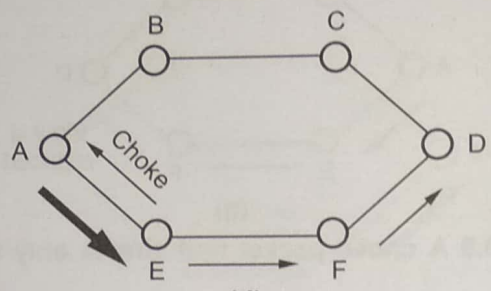
Scanned with CamScanner

iii) The choke packets follows the exactly reversed path of traffic flowing packets. Here the choke packet reaches to node F. Immediately after reaching choke packet at node F, the traffic flow towards node D reduces.



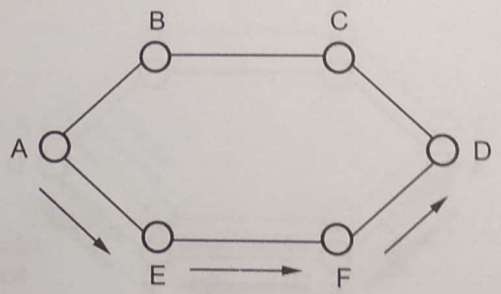
(c)

iv) As choke packet crosses node E, the traffic flow between nodes E, F and F, D is reduced.



(d)

v) After reaching choke packet to source node A, the traffic flow between node A and node E and hence upto the destination node D is reduced.



(e)

Fig. 3.10.9 A choke packet that affects each hop it passes

Hop-by-hop choke packets

- Over long distances or at high speeds, the choke packets are not very effective.
- A more efficient way is to send choke packets hop-by-hop.
- A congested node would again generate a choke pocket, but each hop would be needed to reduce its transmission even before the choke packet arrives at the source.

Scanned with CamScanner

University Questions

1. *Explain the leaky bucket algorithm.* GTU : Dec.-10, Marks 3
2. *Explain leaky bucket and token bucket algorithm in detail.* GTU : June-11, Marks 7
3. *Explain the following techniques for achieving good quality of service.*
   i) Traffic shaping ii) Leaky bucket algorithm
4. *Leaky bucket algorithm.* GTU : May-12, Marks 7
5. *Explain choke packet mechanism for congestion control.* GTU : Winter-13, Marks 4
6. *Write about hop by hop choke packets scheme.* GTU : Winter-14, Marks 3
   GTU : May-12, Marks 3

3.11 Performance

GTU : Dec.-10, June-11

- The network performance is measured in various parameters such as : bandwidth, throughput, bandwidth delay product and jitter.

3.11.1 Bandwidth

- Bandwidth is a characteristic of network. Bandwidth can be measured in hertz and in bits per seconds.

Bandwidth in Hertz

- Bandwidth in hertz, refers to the range of frequencies in a composite signal or the range of frequencies that a transmission channel can pass.

Bandwidth in bps

- Bandwidth in bps refers to speed of bit transmission in a channel or link.

3.11.2 Throughput

- Throughput is an actual measurement of how fast data can be transmitted where as bandwidth is a potential measurement of link.
- Throughput is usually less than bandwidth.

3.11.3 Latency

- Latency is also termed as delay. Latency is time required for a message to completely arrive at the destination from source. It has four components propagation time, transmission time, queuing time and processing delay.

3.11.4 Bandwidth - Delay Product

- The bandwidth and delay are two performance parameters of a link. The bandwidth-delay product defines the number of bits that can fill the link.

Scanned with CamScanner

3.11.5 Jitter

Jitter is a parameter related to delay. Jitter is introduced since different packets of data encounter different delays. The data packets reaching at receiver at different times causing jitter.

Formulas

$$\text{Latency} = \text{Propagation time} \times \text{Transmit time} \times \text{Queue size}$$

$$\text{Propagation time} = \frac{\text{Distance}}{\text{Speed of light}}$$

$$\text{Transmit time} = \frac{\text{Packet size}}{\text{Bandwidth}}$$

$$\text{Throughput} = \frac{\text{Packet transfer size}}{\text{Packet transfer time}}$$

$$\text{Transfer time} = \text{Round trip time} + \frac{1}{\text{Bandwidth} + \text{Packet transfer size}}$$

University Questions

1. *Explain various quality of service parameters for transport layer.* **GTU : June-11, Marks 3**
2. *Explain the following terms : Jitter, bandwidth and throughput.* **GTU : Dec.-10, Marks 3**

❑❑❑

4

Syllabus

Introduction
Internet, Ro

Contents

4.1 Functi
4.2 Netwo
4.3 Forwa
4.4 Routi
4.5 Unica
4.6 Distar
4.7 Link S
4.8 Hierar
4.9 Flood
4.10 Broad
4.11 Borde
4.12 Comp
4.13 Multic
4.14 Routi
4.15 IPv4
4.16 IPv6
4.17 Mobil
4.18 Study

Scanned with CamScanner

4 Network Layer

Syllabus

Introduction, Virtual and datagram networks, Study of router, IP protocol and addressing in the Internet, Routing algorithms, Broadcast and multicast routing.

Contents

4.1 Function of Network Layer . . . . . **Dec.-11,** . . . . . Marks 5

4.2 Network Layer Design Issue. **Dec.-10, June-11,**
. . . . . **Winter-13,** . . . . . Marks 4

4.3 Forwarding . . . . . **Summer-14,** . . . . . Marks 7

4.4 Routing

4.5 Unicast Routing Protocol . . . . . **Dec.-11, Winter-13, 14,** . . . Marks 7

4.6 Distance Vector Routing. **Dec.-10, 11, June-11, Winter-12,**
. . . . . **Summer-13, 14** . . . . . Marks 7

4.7 Link State Routing **Dec.-10, June-11, May-12,**
. . . . . **Winter-12, Summer-13,** . . Marks 7

4.8 Hierarchical Routing . . . . . **Summer-14,** . . . . . Marks 4

4.9 Flooding

4.10 Broadcast Routing

4.11 Border Gateway Protocol (BGP)

4.12 Comparison of RIP and OSPF

4.13 Multicast Routing . . . . . **Summer-14,** . . . . . Marks 3

4.14 Routing for Mobile Hosts

4.15 IPv4 Addresses **Dec.-10,11, Winter-12, 14,**
. . . . . **Summer-14,** . . . . . Marks 7

4.16 IPv6 . . . . . **Dec.-11, Winter-12,**
. . . . . **Summer-13,** . . . . . Marks 7
. . . . . **Winter-12,13,** . . . . . Marks 7

4.17 Mobile IP

4.18 Study of Router

Scanned with CamScanner

4.1 Function of Network Layer

GTU : Dec.-11

- The basic function of network layer is to provide an end-to-end communications capability to the transport layer which lies above it. Network layer is the lowest layer that deals with end-to-end transmission.
- To achieve the goal, the network layer must know about the topology of the communication subnet i.e. set of all routers and choose appropriate path through it. Network layer also takes care of loading of the chosen root.
- The network layer protocols are concerned with the exchange of packets of information between transport layer entities.
- A packet is a group of bits that includes data bits plus source and destination addresses.
- The service provided by the network layer to the transport layer is called **network service**.
- The functions carried out by a layer are different from its services. Functions are those activities which are carried out by a layer in order to provide the services.
- The network layer functions are carried out by adding a header to every Network Service Data Unit (NSDU) forming Network Protocol Data Unit (NPDU). The header contains all the information necessary for carrying out functions.
  1) It keeps track which MAC (Media Access Control), the unique number that each network card has address to send i.e. decides which system receives the information.
  2) It makes routing of data through network from source to destination.
  3) Virtual circuits are established in this layer.
  4) It translates logical network address into physical machine address.
  5) It breaks large packets into smaller so that it will be accepted by the frame of data link layer.
  6) Flow control of packetized information and congestion avoidance is concern of protocol.
  7) It determine the Quality Of Service (QOS) parameter.

University Question

1. *Design issues of network layer.* GTU : Dec.-11, Marks 5

4.2 Network Layer Design Issue

GTU : Dec.-10, June-11, Winter-13

Design issue of the network layer is discussed in this section

Scanned with CamScanner

4.2.1 Store and Forward Packet Switching

- Fig. 4.2.1 shows the network layer protocol environment.
- Host H1 is connected to one of the carrier's routers (A) by leased line.
- Host H2 is on a LAN with a router (F), owned and operated by the customer. This router is also connected by leased line to carrier's equipment.
- A host with a packet to send transmits it to the nearest router, either on its own LAN or a point to point link to the carrier.

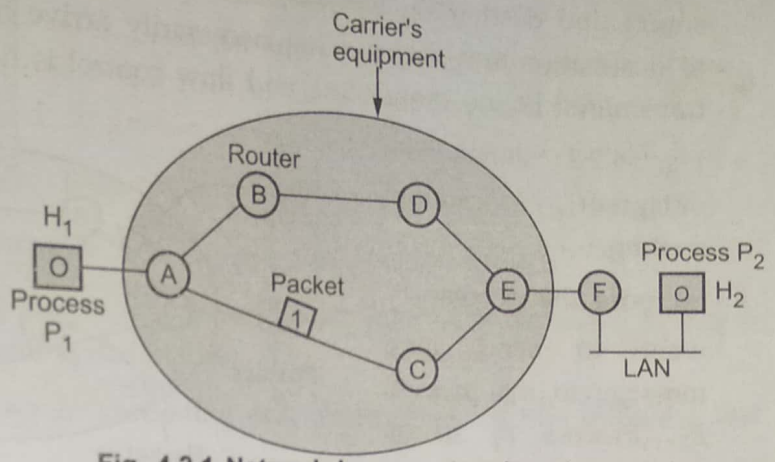


Fig. 4.2.1 Network layer protocol environment

- The packet is stored there until it has fully arrived so the checksum can be verified. Then the packets are forwarded to the next router along the path until it reaches the destination host.
- The above process or mechanism is called store and forward packet switching.

4.2.2 Services Provided to the Transport Layer

- Network layer provides the services to the transport layer at the network layer/transport layer interface.
- The network layer must fulfill following requirements.
  1. The service should be independent of network topology.
  2. The network addresses should be made available to the transport with a uniform numbering plan.
  3. The transport layer should be shielded from the number, type and topology of the routers present.

4.2.3 Implementation of Connectionless Service

- Connectionless network services is also known as **datagrams.** A datagram is a self-contained message unit which contains sufficient information to allow it to be routed from source DTE to destination DTE.

Scanned with CamScanner

- Each packet is treated independently. The network layer simply accepts messages from the transport layer and sends out each one as an isolated unit.
- Datagram system is analogous to postal system. Each packet contains the full source and destination address. Each packet follows a different route from source to destination and they do not necessarily arrive in the same order as they were transmitted i.e. no sequencing and flow control is done at receiver.
- Fig. 4.2.2 shows the datagram subnet routing.
- Suppose the process P_1 want to send long message to the process P_2. Process P_1 hands the message to the transport layer with instructions to deliver it to the process P_2 on host H_2.

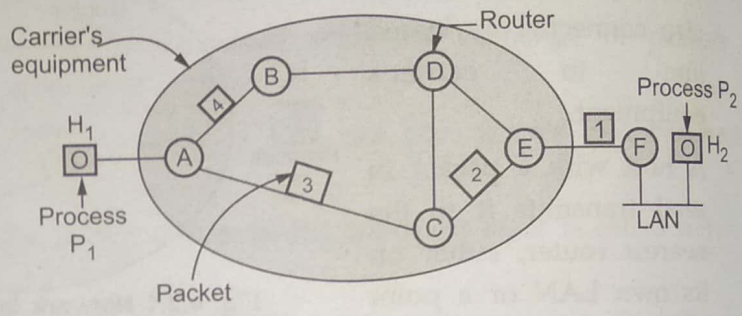


Fig. 4.2.2 Datagram subnet routing

- The transport layer code runs on H_1 within the OS. It prepends a transport header to the front of the message and hands the result to the network layer.
- Message size is four times longer than the maximum packet size. So the network layer breaks the packet into four packets i.e. packet 1, 2, 3 and 4.

A) Routing table of A

i) Initially

A	–
B	B
C	C
D	B
E	C
F	C

ii) Later

A	–
B	B
C	C
D	B
E	B
F	B

Scanned with CamScanner

B) Routing table of C

A	A
B	A
C	–
D	D
E	E
F	E

C) Routing table of E

A	C
B	D
C	C
D	D
E	–
F	F

- All four packets are send to router A using point to point protocol.
- Upto this, carrier takes over and every router has an internal table telling it where to send packets for each possible destination.
- Each routing table entry is a pair consisting of a destination and the outgoing line to use for that destination.
- Router A is connected with two outgoing line i.e. B and C. So every incoming packet must be sent to one of these routers, even if the destination is some other router.
- Initial routing table for router A is shown in figure. At router A, packets 1, 2 and 3 were stored briefly for verifying the checksums.
- After verifying checksum, each packet was forwarded to router C according to the router A routing table.
- Packet 1 was then forwarded to E and then to router F.
- When packet reach at F, it was encapsulated in a data link layer frame and sent to H_2 over the LAN. The packets 2 and 3 follows the same route.
- When packet 4 comes at router A, it forwards to router B even if the destination is router F. This is because of the traffic jam or any other reason.
- Routing table at router A is shown in figure with **later** case. Packet 4 uses this routing table.
- The algorithm that manages the tables and makes the routing decisions is called the **routing algorithm.**

4.2.4 Implementation of Connection-oriented Service

- Connection-oriented network is also known as **virtual circuit.** Virtual circuit is similar to telephone system. A route, which consists of a logical connection is first established between two users. The connection that is established is not a

Scanned with CamScanner

dedicated path between stations. The path is generally shared by many other virtual connections.

- The process is completed in three main phases -
  i) Establishment phase.
  ii) Data transfer phase.
  iii) Connection release phase.

i) Establishment phase :

During setting up of logical connection, the two users not only agree to setup a connection between them but also decide upon the quality of service associated with the connection. After this the sequences of packetized information are transmitted bidirectionally between the nodes. The information is delivered to the receiver in the same order as transmitted by sender.

ii) Data transfer phase :

During this phase it performs flow control and error control services.

The error control service ensures correct sequencing of packets and correct arrival of packets.

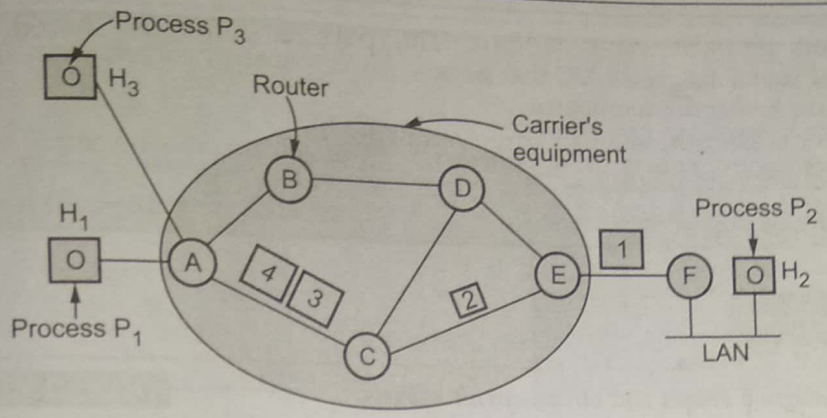
Flow control service ensures a slow receiver from being overwhelming with data from a faster transmitter.

iii) Connection release :

When the stations wish to close down the virtual circuit, one station can terminate the connection with a clear-request packet.

- Fig. 4.2.3 shows the routing within a virtual circuit subnet.
  (See Fig. 4.2.3 on next page)
- Host H_1 has established connection 1 with Host H_2. It is remembered as the first entry in each of the routing table. The first line of A's routing table says that if a packet bearing a connection identifier 1 comes in from H_1, it is to be sent to router C and given connection identifier 1.
- If Host H_3 also wants to establish a connection to H_2, it select connection identifier 1 and tells the subnet to establish the virtual circuit. This leads to the second row in the tables.
- Here is the conflict because although router A can easily distinguish connection 1 packets from H_1 from connection 1 packets from H_3 but router C cannot do this. For this reason, router A assigns a different.

Scanned with CamScanner



Routing table of A

H_1	1
H_3	1

in

C	1
C	2

out

Routing table of C

A	1
A	2

E	1
E	2

Routing table of E

C	1
C	2

F	1
F	2

Fig. 4.2.3 VC subnet routing

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.

4.2.5 Comparison between Virtual Circuit and Datagram Subnet

S.N.	Virtual circuit	Datagram
1.	Circuit setup is required.	Circuit setup is not required.
2.	Each packet contains a short VC number as address.	Each packet contains the full source and destination address.
3.	Route chosen when VC is setup and all packets follow this route.	Each packet is routed independently.

Scanned with CamScanner

4.	In case of router failure all VC that passed through the router are terminated.	Only crashed packets lost.
5.	Congestion control is easy using buffers.	Difficult congestion control.
6.	Complexity in the network layer	Complexity in transport layer.

University Questions

1. *Compare datagram subnet and virtual-circuit subnets.* GTU : Dec.-10, Marks 4
2. *Differentiate : Datagram subnet v/s virtual circuit subnet.* GTU : June-11, Marks 4
3. *Compare virtual circuit v/s datagram subnet.* GTU : Winter-13, Marks 4
4. *Explain different types of switching methods with examples.* GTU : Summer-14, Marks 7

4.3 Forwarding

GTU : Summer-14

- Forwarding refers to the way a packet is delivered to the next node. It requires a host or router to have a routing table.
- Forwarding refers to the router local action of transferring a datagram from an input link interface to the appropriate output link interface.
- When host has a packet to send, it looks at routing table to find the route to the final destination.

Types of forwarding techniques

1. Next hop versus route method.
2. Network specific versus host specific method.
3. Default method.

1) Next hop versus route method

- Fig. 4.3.1 shows network with routing table for this method. This method reduce the content of rounting table. Routing table stores only the address of the next hop.

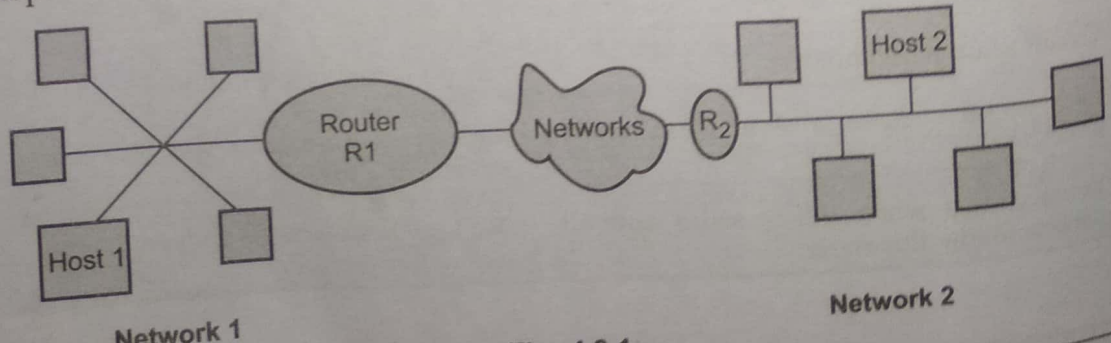


Fig. 4.3.1

Scanned with CamScanner

Routing table for next hop

For host 1

Destination address	Next hop
Host 2	R1

For router R2

Destination Address	Next hop
Host 2	R2

For Router R2

Destination Address	Next hop
Host 2	-

2) Network specific versus host specific method

- It simplify the searching process and also reduce the routing table size.
- Routing table contains only the address of the destination network.
- It provides good security.

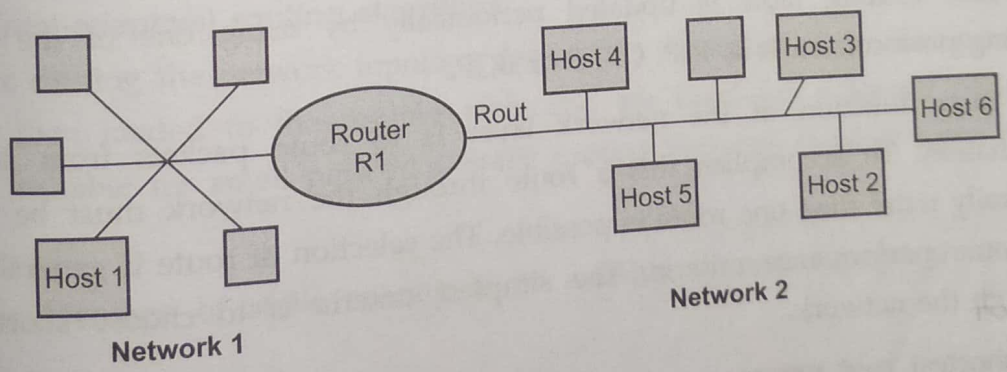


Fig. 4.3.2

Routing table for host 1

Destination address	Next hop
Network 2	R1

Scanned with CamScanner

3) Default method

- Host is connected with more than one routers.
- A router is assigned to receive all packets with no match in the routing table.
- Default router is used for communication with outside world.

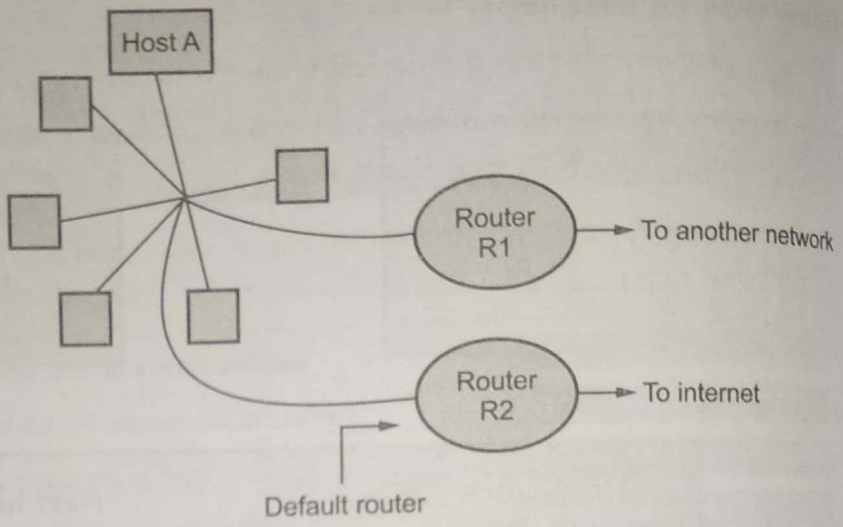


Fig. 4.3.3

4.4 Routing

- A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets. Routing table can be either static or dynamic.
- A static routing table contains information entered manually. The administrator enters the route for each destination into the table.
- Dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPE or BGP.
- The main function of the network layer is to route packets from source to destination. To accomplish this a route through the network must be selected, generally more than one route is possible. The selection of route is generally based on some performance criteria. The simplest criteria is to choose shortest root through the network.
- The shortest root means a route that passes through the least number of nodes. This shortest root selection results in least number of hops per packet. A routing algorithm is designed to perform this task. The routing algorithm is a part of network layer software.

Scanned with CamScanner

Properties of routing algorithm

Certain properties which are desirable in a routing algorithm are -

Correctness, simplicity, robustness, stability, fairness, optimality and efficiency.

1. Correctness and simplicity are self-explanatory.
2. Robustness means the ability to cope with changes in the topology and traffic without requiring all jobs in hosts to be aborted and network to be rebooted everytime.
3. Stability refers to equilibrium state of algorithm. It is the technique that react to changing conditions such as congestions. Under any conditions the network must not react too slow or experience unstable swings from one extreme to another.
4. Some performance criteria may favour the exchange of data packets between nearby stations and discourage the exchange between distant stations. Some compromise is needed between fairness and optimality.

Routing algorithm classification

Routing algorithm can be classified in several ways. Based on their responsiveness it can be classified into two types -

1. Static (non-adaptive) Routing Algorithms.
2. Dynamic (adaptive) Routing Algorithms.

1. Static (non-adaptive) routing algorithms

In static routing the network topology determines the initial paths. The precalculated paths are then loaded to the routing table and are fixed for a longer period. Static routing is suitable for small networks. Static routing becomes **cumber some** for bigger networks.

The disadvantage of static routing is its inability to respond quickly to network failure.

2. Dynamic (Adaptive) routing algorithms

Dynamic routing algorithms change their routing decision if there is change in topology, traffic. Each router continuously checks the network status by communicating with neighbours. Thus a change in network topology is eventually propagated to all the routers. Based on this information gathered, each router computes the suitable path to the destination.

The disadvantage of dynamic routing is its complexity in the router.

Scanned with CamScanner

Routing tables

Once the routing decision is made, this information is to be stored in routing table so that the router knows how to forward a packet. In virtual circuit packet switching, the routing table contains each incoming packet number and outgoing packet number and output port to which the packet is to forward. In datagram networks, routing table contains the next hop to which to forward the packet, based on the destination address.

4.4.1 Advantages and Disadvantages of Static Routing

Advantages

1. Minimal CPU/Memory overhead.
2. Granular control on how traffic is routed.
3. Simple to configure and maintain.
4. Secure as only defined routes can be accessed.
5. Bandwidth is not used for sending routing updates.

Disadvantages

1. Manual update of routes after changes
2. Explicit addition of routes for all networks
3. Impractical on large network.

4.4.2 Advantages and Disadvantages of Dynamic Routing

Advantages

1. Simpler to configure on larger networks.
2. Will dynamically choose a different (or better) route if a link goes down.
3. Ability to load balance between multiple links.

Disadvantages

1. Updates are shared between routers, thus consuming bandwidth.
2. Routing protocols put additional load on router CPU/RAM.
3. The choice of the "best route" is in the hands of the routing protocol, and not the network administrator

Scanned with CamScanner

4.4.3 Difference between Static and Dynamic Routing

Sr. No.	Static routing (Non adaptive)	Dynamic routing (Adaptive)
1.	Static routing manually sets up the optimal paths between the source and the destination computers.	Dynamic routing uses dynamic protocols to update the routing table and to find the optimal path between the source and the destination computers.
2.	The routers that use the static routing algorithm do not have any controlling mechanism if any faults in the routing paths.	The dynamic routing algorithms are used in the dynamic routers and these routers can sense a faulty router in the network.
3.	These routers do not sense the faulty computers encountered while finding the path between two computers or routers in a network.	the dynamic router eliminates the faulty router and finds out another possible optimal path from the source to the destination.
4.	The static routing is suitable for very small networks and they cannot be used in large networks	Dynamic routing is used for larger networks.
5.	The static routing is the simplest way of routing the data packets from a source to a destination in a network.	The dynamic routing uses complex algorithms for routing the data packets.
6.	The static routing has the advantage that it requires minimal memory.	Dynamic routers have quite a few memory overheads, depending on the routing algorithms used.
7.	The network administrator finds out the optimal path and makes the changes in the routing table in the case of static routing.	In the dynamic routing algorithm, the algorithm and the protocol is responsible for routing the packets and making the changes accordingly in the routing table.

4.4.4 Design Goals

Routing algorithms often have one or more of the following design goals :

1. Optimality
2. Simplicity and low overhead
3. Robustness and stability
4. Rapid convergence
5. Flexibility.

1. Optimality

Optimality refers to the ability of the routing algorithm to select the best route. The best route depends on the metrics and metric weightings used to make the calculation. For example, one routing algorithm might use number of hops and delay, but might

Scanned with CamScanner

weight delay more heavily in the calculation. Naturally, routing protocols must strictly define their metric calculation algorithms.

2. Simplicity

Routing algorithms are also designed to be as simple as possible. In other words, the routing algorithm must offer its functionality efficiently, with a minimum of software and utilization overhead. Efficiency is particularly important when the software implementing the routing algorithm must run on a computer with limited physical resources.

3. Robustness

Routing algorithms must be robust. In other words, they should perform correctly in the face of unusual or unforeseen circumstances such as hardware failures, high load conditions and incorrect implementations. Because routers are located at network junction points, they can cause considerable problems when they fail. The best routing algorithms are often those that have withstood the test of time and proven stable under a variety of network conditions.

4. Rapid convergence

Routing algorithms must converge rapidly. Convergence is the process of agreement, by all routers, on optimal routes. When a network event causes routes to either go down or become available, routers distribute routing update messages. Routing update messages permeate networks, simulating recalculation of optimal routes and eventually causing all routers to agree on these routes. Routing algorithms that converge slowly can cause routing loops or network outages.

5. Flexibility

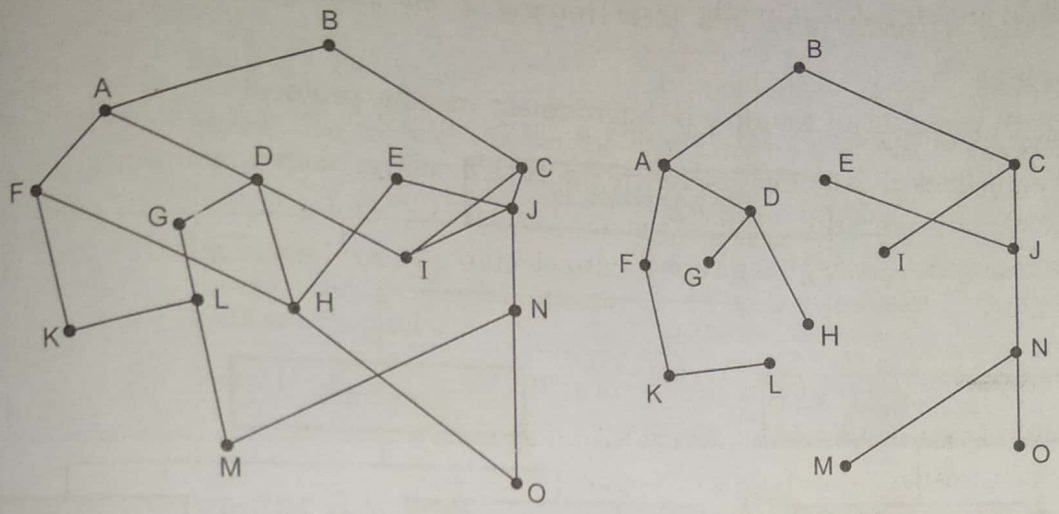
Routing algorithms should also be flexible. In other words, routing algorithms should quickly and accurately adapt to a variety of network circumstances. For example, assume that a network segment has gone down. Many routing algorithms, on becoming aware of this problem, will quickly select the next-best path for all routes normally using that segment. Routing algorithms can be programmed to adapt to changes in network bandwidth, router queue size, network delay, and other variables.

4.4.5 Optimally Principle

- Optimally principle 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.
- Suppose route from I to J is r_1 and rest of the route is called r_2. If a route better than r_2 is existed from J to K, it could be concatenated with r_1 to improve the route from I to K, so that r_1r_2 is optimal.

Scanned with CamScanner

- Fig. 4.5.1 shows the subnet and sink tree with distance metric is measured as the number of hops.



(a) Subnet (b) Sink tree for router B

Fig. 4.4.1 Subnet and sink tree

- Sink tree is not necessarily unique, other trees with the same path lengths may exist.
- Sink tree does not contain any loops, so each packet will be delivered within a finite and bounded number of hops.

University Question

1. *Write and explain statement of Optimality Principle with example.* **GTU : Winter-14, Marks 4**

4.5 Unicast Routing Protocol

GTU : Dec.-11, Winter-13, 14

- Routing table can be static or dynamic. Manual entries are done in static table.
- Dynamic table is updated automatically when there is a change somewhere in the internet.
- Now a day, dynamic table is used because of sudden changes in the internet. One of the routers in the internet may fail or link between any two routers is down. So because of these reasons dynamic table is required.
- Routing protocol is a combination of rules and procedures that let routers in the internet inform each other of changes.

4.5.1 Intra and Inter-domain Routing

- An internet is divided into autonomous systems. An **autonomous system** is a group of networks and routers under the authority of a single administration.
- Routing inside an autonomous system is referred to as **intradomain** routing.

Scanned with CamScanner

- Routing between autonomous system is referred to as **interdomain** routing.
- Distance vector and link state routing is the example of intradomain routing protocols.
- Path vector is an example of interdomain routing protocol.

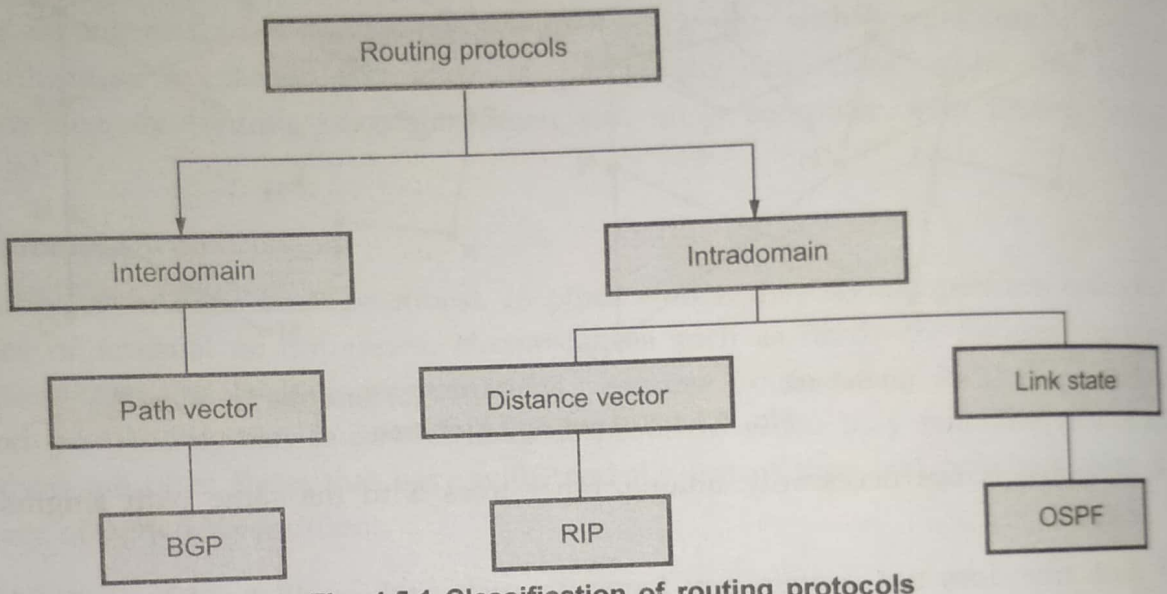


Fig. 4.5.1 Classification of routing protocols

- Fig. 4.5.2 shows an autonomous system.
- Only one interdomain routing protocol handles routing between autonomous systems.

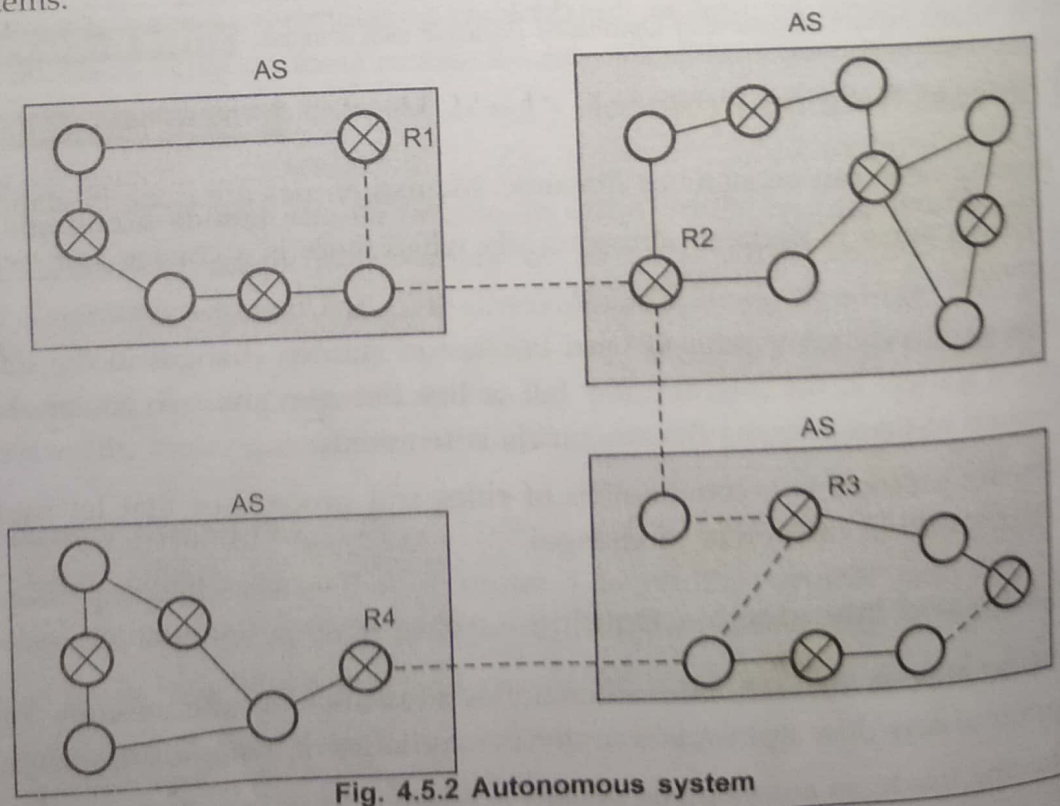


Fig. 4.5.2 Autonomous system

Scanned with CamScanner

4.5.2 Comparison between Intra and Inter-domain Routing

Sr. No.	Intra-domain routing	Inter-domain routing
1.	Routing within an AS.	Routing between AS's.
2.	Ignores the Internet outside the autonomous system.	Assumes that the Internet consists of a collection of interconnected AS's.
3.	Protocols for Intra-domain routing are also called Interior Gateway Protocols.	Protocols for inter-domain routing are also called Exterior Gateway Protocols.
4.	Popular protocols are RIP and OSPF.	Routing protocols are BGP.

4.6 Distance Vector Routing

GTU : Dec.-10, 11, June-11, Winter-12, Summer-13, 14

- Distance vector routing algorithm is the dynamic routing algorithm. It was designed mainly for small network topologies. Distance vector routing algorithm is sometimes called by other names, most commonly the distributed **Bellman-Ford** routing algorithm and the **Ford-Fulkerson** algorithm.
- The term distance vector derives from the fact that the protocol includes its routing updates with a vector of distances, or hop counts.
- In this algorithm, each router maintains a routing table indexed by, and containing one entry for, each router in the subnet. This entry contains two parts :
  a. The preferred outgoing line to use for that destination.
  b. 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, etc.
- Assume that delay is used as a metric and that the router knows the delay to each of its neighbours. All nodes exchange information only with their neighbouring nodes. Nodes participating in the same local network are considered neighbouring nodes.
- Once every 'T' msec each router sends to each neighbor a list of its estimated delays to each destination. It also

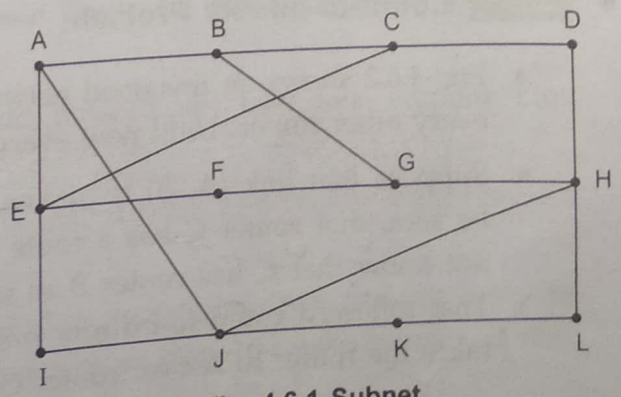


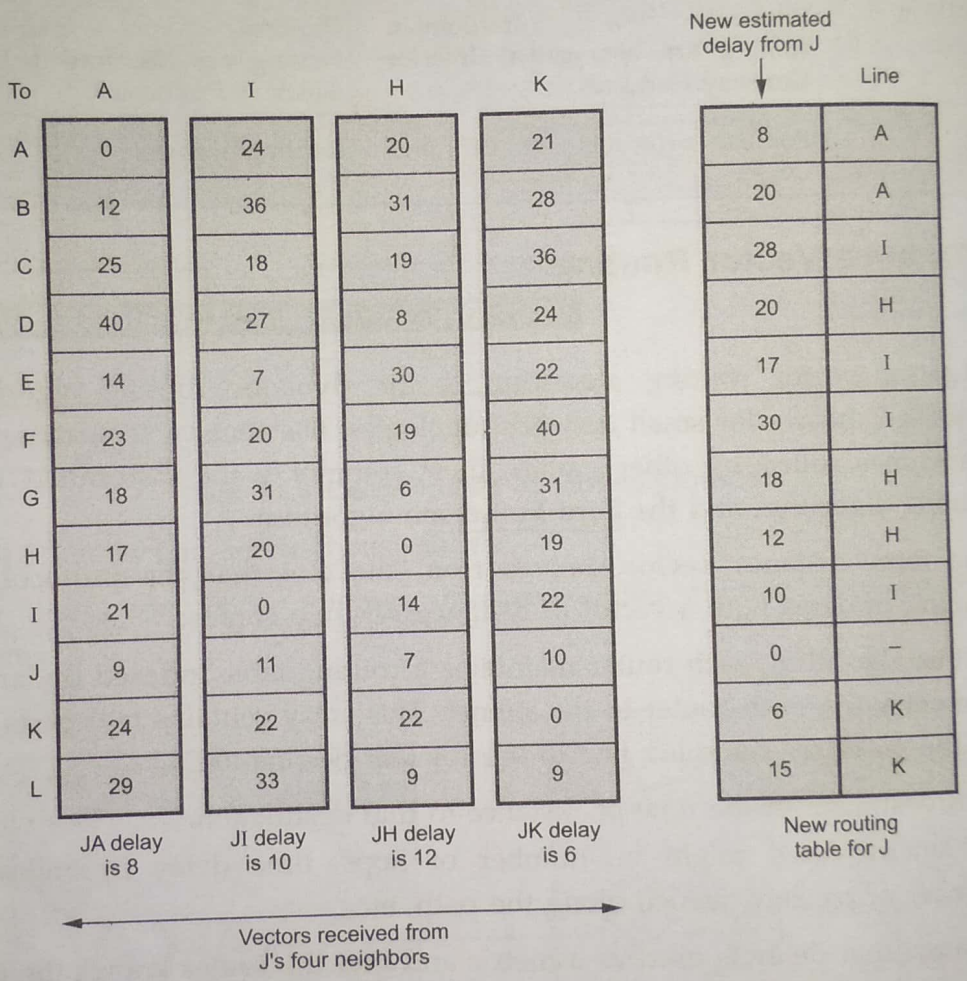
Fig. 4.6.1 Subnet

Scanned with CamScanner

receives a similar list from each neighbor.

- By performing 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. Old routing table is not used in the calculation.
- Fig. 4.6.1 shows the subnet with 12 routers.

Routing table is shown below.

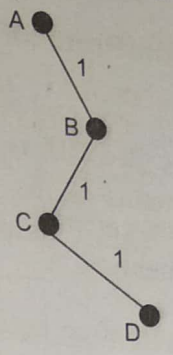


4.6.1 Count-to-Infinity Problem

- Fig. 4.6.2 shows an imagined network and denotes the distances from router A to every other router. Until now everything works fine.
- Suppose that link (A, B) is broken. Router B observed it, but in his routing table he sees, that router C has a route to A with 2 hops. The problem is, that B does not know that C has router B as successor in his routing table on the route to A. That followed **count-to-infinity** problem. Router B actualizes his routing table and takes the router to A over router C.

Scanned with CamScanner

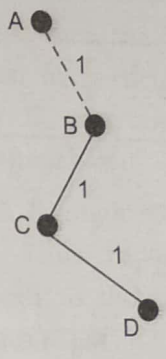
- In Fig. 4.6.2, we can see the new distances to A. In router C's routing the route to A contains router B as next hop router, so if B has increased his costs to A, C is forced to do so. Router C increases his cost to A about B + 1 = 4.
- Now we see the conseuqence of the distributed Bellman-Ford protocal : Because router B takes the path over C to A, he reactualizes his routing table and so on.
- There are several partial solutions to the count-to-infinity problem. The first one is to use some relatively small number as an approximation of infinity. For example, we might decide that the maximum number of hops to get across a certain network is never going to be more than 16 and so we could pick 16 as the value that represents infinity.



Routing table for A

B	C	D
1	2	3

Fig. 4.6.2



B	C	D
3	2	3

B	C	D
3	4	3

B	C	D
5	4	5

Fig. 4.6.3

- This at least bounds the amount of time that it takes to count to infinity. Of course, it could also present a problem if our network grew to a point where some nodes were separated by more than 16 hops.
- One technique to improve the time to stabilize routing is called **split horizon**. Split horizon technique implies that routing information about some network stored in the routing table of a specific router is never sent to the router from which it was received.

Issues with the distance vector routing

1. The primary drawback of this algorithm is its vulnerability to the 'Count-to-Infinity' problem. There have been proposed many partial solutions but none works under all circumstances.
2. Another drawback of this scheme is that it does not take into account Link Bandwidth.
3. Yet another problem with this algorithm is that it takes appreciably long time for convergence as the network-size grows.
4. A fallout of the Count-to-Infinity issue and slow convergence has been to limit the maximum number of hops to 15 which means more than 16-router subnets, it may not be appropriate routing algorithm.

Scanned with CamScanner

4.6.2 Routing Information Protocol

- In RIP, routing updates are exchanged between neighbours approximately every 30 seconds using a so-called **RIP response message**. The response message sent by a router or host contains a list of upto 25 destination networks within an Autonomous System (AS). Response messages are also known as **RIP advertisements**.
- Fig. 4.6.4 shows a portion of an autonomous system.

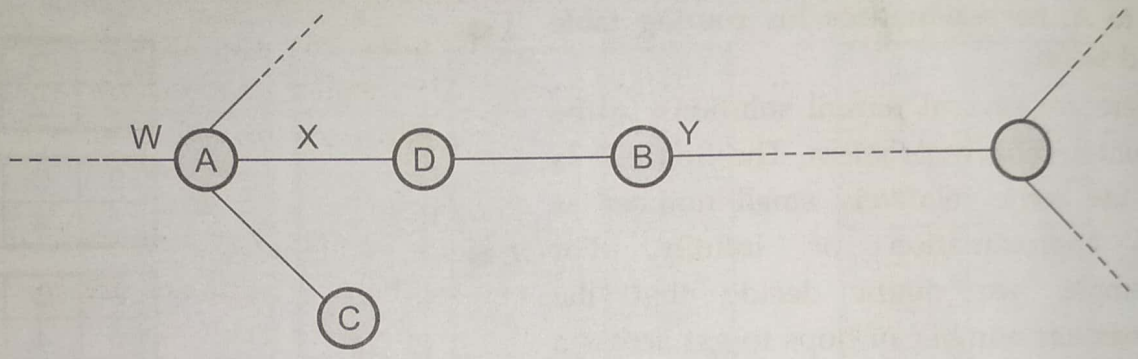


Fig. 4.6.4 Portion of AS

- Forwarding table in router D before receiving advertisement from router A. For this example, the table indicates that to send a datagram from router D to destination network W, the datagram should first be forwarded to neighbouring router A; the table also indicates that destination network W is two hops away along the shortest path.
- The Table 4.6.1 also indicates that network Z is seven hops away via router B.

Destination network	Next router	Number of hops to destination
W	A	2
Y	B	2
Z	B	7
X	–	1
.....	.....	.....

Table 4.6.1 Forwarding table

- Now suppose that 30 seconds later, router D receives from router A the advertisement shown in Table 4.6.2.

Scanned with CamScanner

Destination network	Next router	Number of hops to destination
Z	C	4
W	–	1
X	–	1
.....	.....	.....

Table 4.6.2 Advertisement from router A

- Note that the advertisement is nothing other than the forwarding table information from router A. This information indicates, in particular, that network Z is only four hops away from router A. Router D, upon receiving this advertisement, merges the advertisement with the old routing table.
- Router D learns that there is now a path through router A to network Z that is shorter than the path through router B. Thus, router D updates its forwarding table to account for the shorter shortest path, as shown in Table 4.6.3.

Destination network	Next router	Number of hops to destination
W	A	2
Y	B	2
Z	A	3
.....	.....	.....

Table 4.6.3

- RIP routers exchange advertisements approximately every 30 seconds. If a router does not hear from its neighbour atleast once every 180 seconds, that neighbour is considered to be no longer reachable; i.e. either the neighbour has died or the connecting link has gone down. When this happens, RIP modifies the local forwarding table and then propagates this information by sending advertisements to its neighbouring routers.
- A router can also request information about its neighbour's cost to a given destination using RIP's request message. Routers send RIP request and response messages to each other over UDP using port number 520.

Scanned with CamScanner

RIP message format

- Fig. 4.6.5 shows the RIP message format.

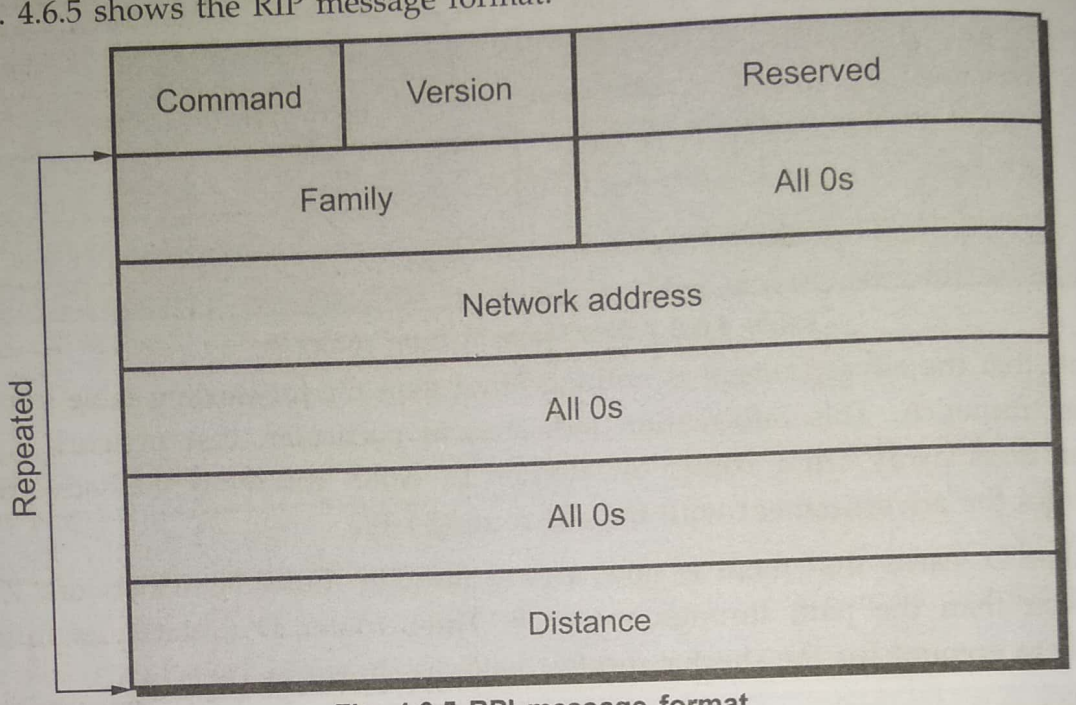


Fig. 4.6.5 RPI message format

1. **Command :** This is 8 bits field specifies the type of message: 1 for request and 2 for response.
2. **Version :** This is 8 bits field define the version.
3. **Family :** This 16 bits field defines the family of the protocol used. For TCP/IP the value is 2.
4. **Network address :** The address field defines the address of the destination network.
5. **Distance :** This 32 bits field defines the hop count from the advertising router to the destination network.

Request and response

- RIP support two types of messages : Request and Response.

Request

- A request message is sent by a router that has just comp up or by a router that has some time out entries.

Response

- A response message can be either solicited or unsolicited.

1. Solicited response

- Is sent only in answer to a request.

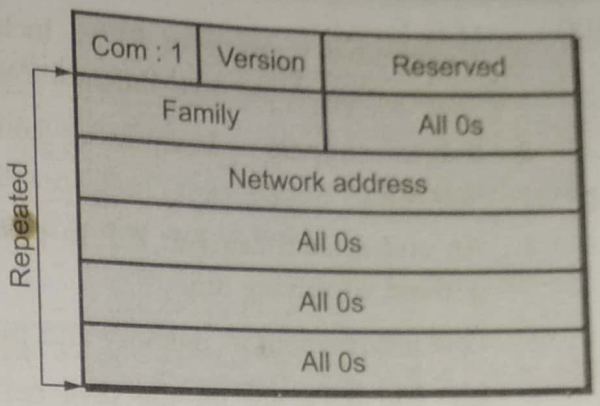
Scanned with CamScanner

- Containing information about the destination specified in the corresponding request.

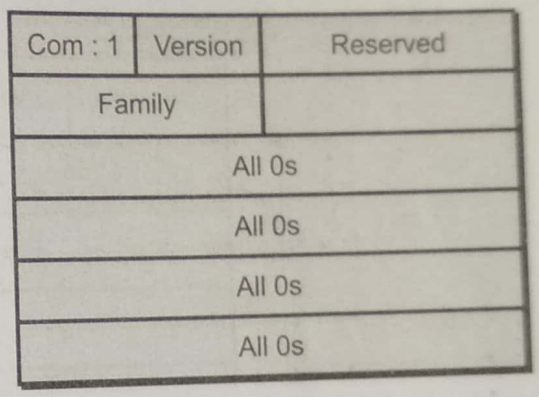
2. Unsolicited response

- Is sent periodically, every 30 seconds.
- **Containing** information covering the whole routing table

Fig. 4.6.6 shows the request message.



(a) Request for some



(b) Request for all

Fig. 4.6.6 Request message format

Timers in RIP

- RIP uses three timers to support its operation.

1. Periodic timer (25 - 35 sec)
2. Expiration (180 sec)
3. Garbage collection (120 sec).

1. **Periodic timer :** This type of timer controls the advertising of regular update messages. Each router has one periodic timer that is randomly set to a number between 25 to 35 seconds.
2. **Expiration timer :** The expiration timer governs the validity of a route. In normal situation, the new update for the route occurs every 30 seconds. But, if there is a problem on an internet and no update is received within the allotted 180 seconds, the route is considered expired and the hop count of the route is set to 16. Each router has its own expiration timer.
3. **Garbage collection timer :** When the information about a route becomes invalid, the router continues to advertise the route with a metric value of 16 and the garbage collection timer is set to 120 sec for that route. When the count reaches zero, the route is purged from the table.

RIPv2

- RIP version 2 was designed to overcome some of the shortcomings of version 1. Replaced fields in version 1 that were filled with 0s for the TCP/IP protocols with some new fields.
- **Advantages**

1. An AS can include several hundred routers with RIP-2 protocol.

Scanned with CamScanner

2. Compatible upgrade of RIPv1 including subnet routing, authentication, CIDR aggregation, route tags and multicast transmission.
3. Subnet support : Uses more convenient partitioning using variable-length subnets
4. An end system can run RIP in passive mode to listen for routing information without supplying any.
5. Low requirement in memory and processing at the node .
6. RIP and RIP2 are for the IPv4 network while the RIPng is designed for the IPv6 network.

Fig. 4.6.7 shows the message format.

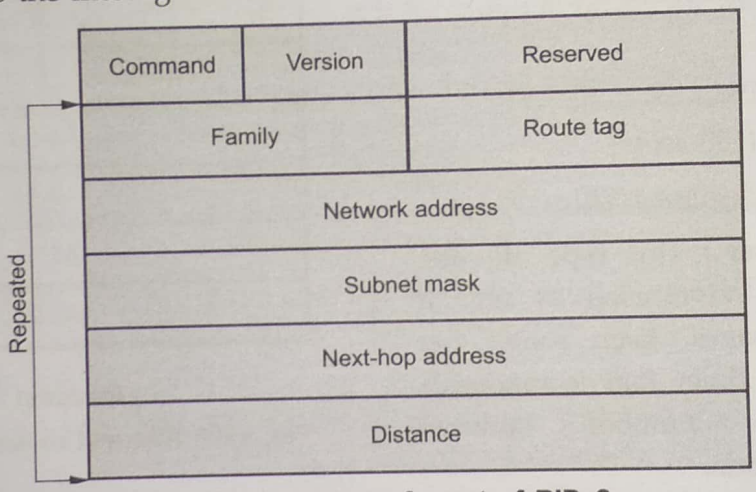


Fig. 4.6.7 Message format of RIPv2

1. **Command -** The command field is used to specify the purpose of the datagram.
2. **Version -** The RIP version number. The current version is 2.
3. **Identifier -** Indicates what type of address is specified in this particular entry.
4. **Route tag -** Attribute assigned to a route which must be preserved and readvertised with a route. The route tag provides a method of separating internal RIP routes from external RIP routes, which may have been imported from an EGP or another IGP.
5. **IP address -** The destination IP address.
6. **Subnet mask -** Value applied to the IP address to yield the non-host portion of the address. If zero, then no subnet mask has been included for this entry.
7. **Next hop -** Immediate next hop IP address to which packets to the destination specified by this route entry should be forwarded.
8. **Distance -** Represents the total cost of getting a datagram from the host to that destination.

Scanned with CamScanner

Authentication

- Authentication is added to protect the message against unauthorized advertisement. No new field is added to the packet.
- To indicate that the entry is authentication information and not routing information, the value of FFFFH is entered in the family field.
- Fig. 4.6.8 shows the authentication.

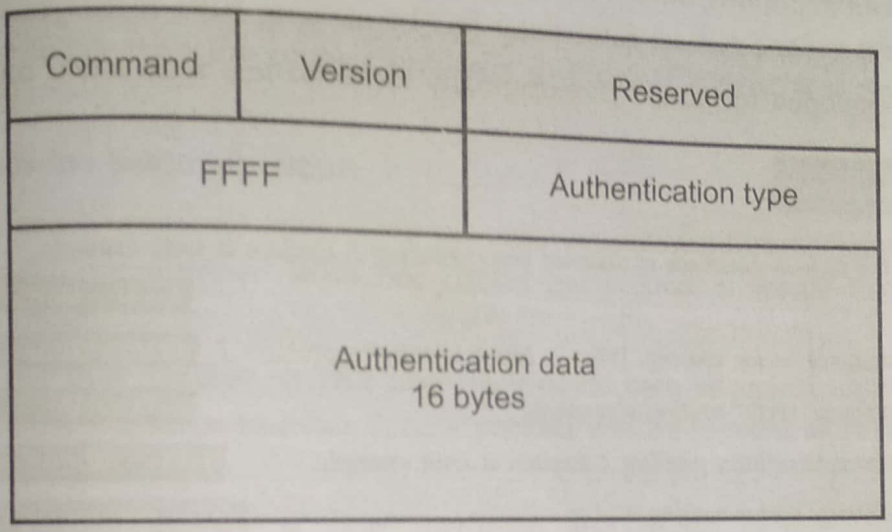


Fig. 4.6.8 Authentication

- Authentication type defines the protocol used for authentication.
- Authentication data is the actual data.

RIP2 - Disadvantages

1. RIP2 supports generic notion of authentication, but only "password" is defined so far. Still not very secure.
2. RIP2 packet size increases as the number of networks increases hence it is not suitable for large networks.
3. RIP2 generates more protocol traffic than OSPF, because it propagates routing information by periodically transmitting the entire routing table to neighbour routers.
4. RIP2 may be slow to adjust for link failures.

Advantages of RIP and Disadvantages of RIP version 1

Advantages of RIP

1. RIP is very useful in a small network, where it has very little overhead in terms of bandwidth used and configuration and management time.
2. Easy to implement than newer IGP's.

Scanned with CamScanner

3. Many implementations are available in the RIP field.

Disadvantages of RIP1

1. Minimal amount of information for router to route the packet and also very large amount of unused space.
2. Subnet support : Supports subnet routes only within the subnet network.
3. Not secure; anyone can act as a router just by sending RIP1 messages.

RIP1 was developed for an AS that originally included less than a 100 routers.

University Questions

1. *What is the serious drawback of distance vector routing ? Explain it with example.* **GTU : Dec.-10, Marks 3**
2. *Explain distance vector routing. What is count to infinity problem ?* **GTU : June-11, Marks 7**
3. *Explain distance vector routing algorithm.* **GTU : Dec.-11, Marks 5**
4. *What is count-to-infinity problem ? Explain it with example.* **GTU : Winter-12, Marks 7**
5. *Explain distance vector routing.* **GTU : Summer-13, Marks 7**
6. *Explain distance vector routing with example.* **GTU : Summer-14, Marks 7**

4.7 Link State Routing

GTU : Dec.-10, June-11, May-12, Winter-12, Summer-13

- Link state routing is the second major class of intradomain routing protocol. It is **dynamic** type routing algorithm.
- The idea behind link state routing is simple and can be stated as five parts. Each router must do the following :
  1. **Learning about the neighbors :** When a router is booted, it sends 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. When two or more routers are connected by a LAN, the LAN can be modeled as a node.
  2. **Measuring line cost :** To determine the cost for a line, a router sends a special ECHO packet over the line 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. Should the load be taken into account when measuring the delay ?
  3. **Building link state packets :** State packets may be built periodically, or when some significant event occurs, such as a line or neighbour going down or coming back up again.

Scanned with CamScanner

4. **Distributing the link state packets :** The basic algorithm

- Each state 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 (i.e., flooding). 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.

Problems with the basic algorithm

1. The sequence numbers may wrap around, causing confusion. Solution : Using a 32-bit sequence number. With one packet per second, it would take 137 years to wrap around.
2. If a router ever crashes, it will lose track of its own sequence number. If it starts again at the sequence number 0, new packets will be rejected as obsolete/duplicate by other routers.
3. If a sequence number is ever corrupted and 65,540 is received instead of 4 (a 1-bit error), packets 5-65540 will be rejected as obsolete.

The solution to router crashes and sequence number corruption is to associate an age with each state packet from any router and decrement the age once per second. When the age hits zero, the information from that router is discarded. Normally a new packet comes in every 10 seconds, so router information only times out when a router is down.

Some refinements to the basic algorithm make it more robust

When a state packet comes in to a router for flooding, it is put in a holding area to wait a short while first. If another state packet from the same source comes in before it is transferred, 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 lines, all state packets are acknowledged. When a line goes idle, the holding area is scanned in round robin to select a packet or acknowledgement to send.

5. **Computing the new routes :** Once a router has accumulated a full set of link state packets, it can construct the entire subnet graph. Then Dijkstra's algorithm can be run locally to construct the shortest path to all possible destinations.

- Link state routing protocols use event driven updates rather than periodic updates. Link state routing is widely used in actual networks. OSPF protocol uses in a link state algorithm.

Scanned with CamScanner

- Link state routing protocols are as follows :
  a. Open Shortest Path First (OSPF)
  b. Netware Link Services Protocol (NLSP).
  c. Apple's AURP.
  d. ISO's Intermediate System-Intemediate System (IS-IS).

4.7.1 Shortest Path Routing

- In shortest path routine the path length between each node is measured as a function of distance, bandwidth, average traffic, communication cost, mean queue length, measured delay etc.
- By changing the weighing function, the algorithm then computes the shortest path measured according to any one of a number of criteria or a combination of criteria. For this a graph of subnet is drawn. With each node of graph representing a router and each arc of the graph representing a communication link. Each link has a cost associated with it. To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

Two algorithms for computing the shortest path between two nodes of a graph are known.

i) Dijkstra's algorithm ii) Bellman-Ford algorithm.

i) Dijkstra's algorithm

Each node is labelled with its distance from the source node along the best known path. Initially no paths are known, so all nodes are labelled with infinity. The algorithm proceed in stages. As the algorithm proceeds, the paths are found, the labels are changed, reflecting better paths. Stepwise proceeding of algorithm is as follows.

Step-I : Source node is initialized and can be indicated as a filled circle.

Step-II : Initial path cost to neighbouring nodes (adjacent nodes) or link cost is computed and these nodes are relabelled considering source node.

Step-III : Examine the all adjacent nodes and find the smallest label, make it permanent.

Step-IV : The smallest label node is now working node, then step-II and step-III are repeated till the destination node reaches.

Following example illustrates Dijkstra's algorithm.

Scanned with CamScanner

Example 4.7.1 *Find the shortest path between node A and node H for the following Fig. 4.7.1 by applying Dijkstra's algorithm. Show each steps output.*

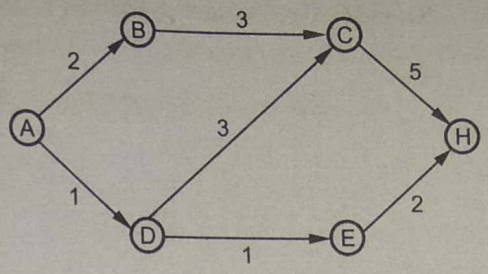


Fig. 4.7.1

Solution :

Step-I : Node A is initialized as source node.

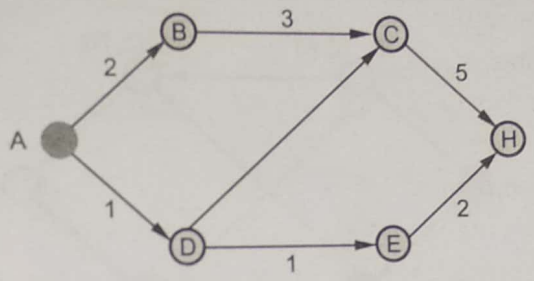


Fig. 4.7.1 (a)

Step-II : Link cost is computed for the adjacent node.

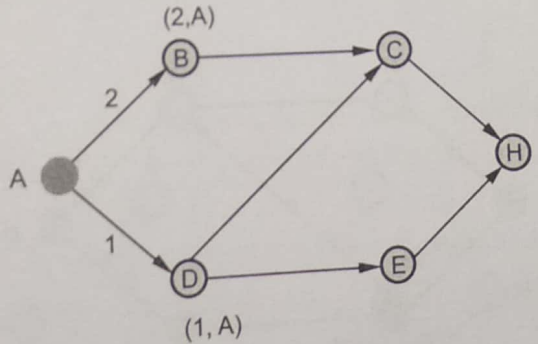


Fig. 4.7.1 (b)

Step-III : Since AD is smallest path, now D is working node.

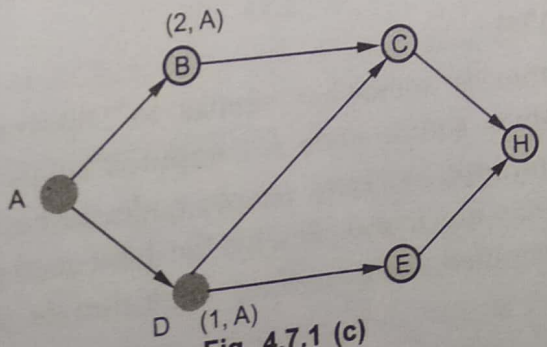


Fig. 4.7.1 (c)

Scanned with CamScanner

Step-IV : Adjacent nodes to D are C and E.

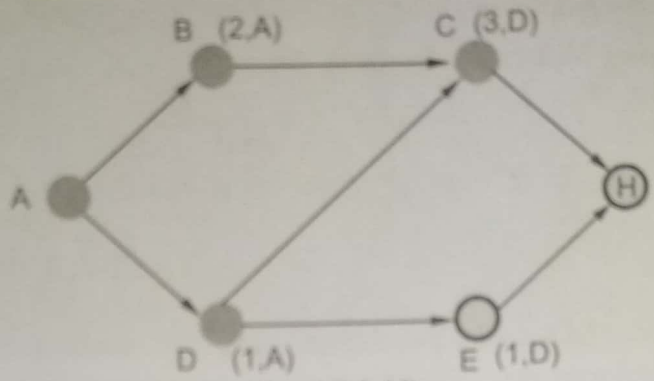


Fig. 4.7.1 (d)

Step-V : Since shortest is E, now E is working node.

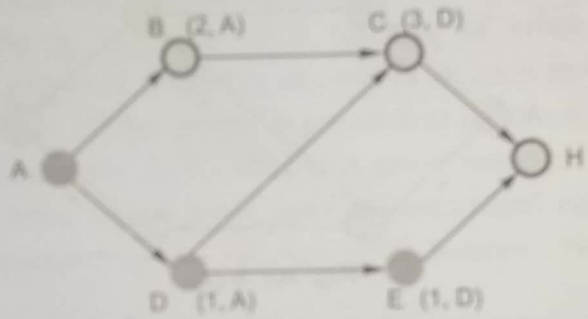


Fig. 4.7.1 (e)

Step-VI :

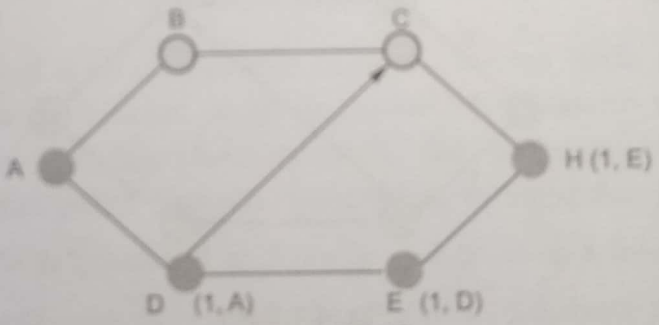


Fig. 4.7.1 (f)

Hence the shortest path between node A and node H is ADEH.

ii) Bellman-Ford algorithm

Bellman-Ford algorithm is somewhat similar to Dijkstra's algorithm but here the shortest paths from a given source node is computed subject to the constraint that the path contain at most one link, i.e. from source node, at each step least-cost path with maximum number of links are found. Finally the least-cost path to each node and the cost of that path is computed. Bellman-Ford algorithm is illustrated in the following example.

Scanned with CamScanner

Example 4.7.2 *Find the shortest path between node A and node H using Bellman-Ford algorithm, for the Fig. 4.7.1 shown in example 4.7.1.*

Solution : Step-1 :

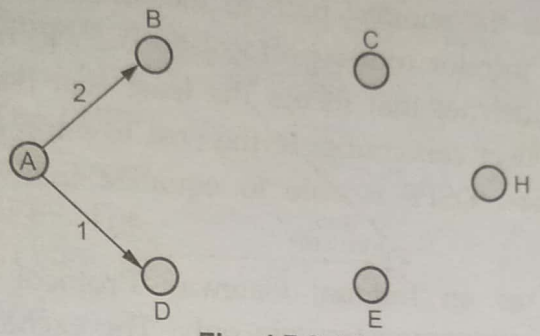


Fig. 4.7.2 (a)

Distance AD is shorter than AB. So route AD is chosen.

Step-2 :

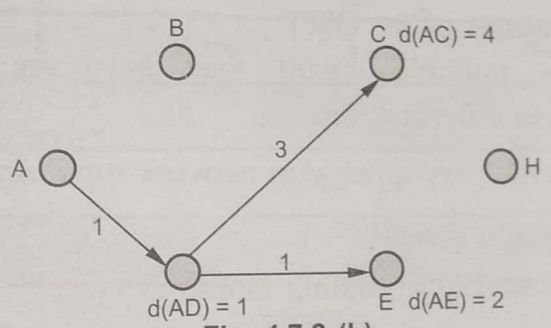


Fig. 4.7.2 (b)

$\because$ d(AE) < d(AC)

$\therefore$ d(AE) is chosen.

Step-3 :

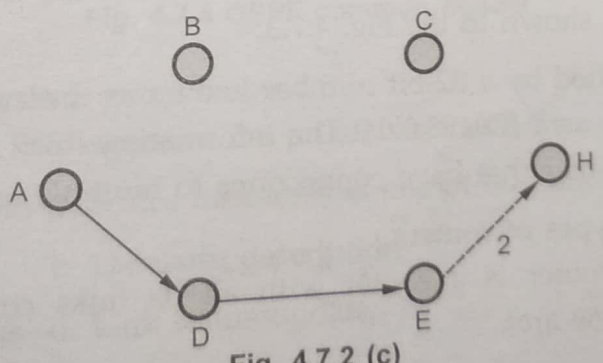


Fig. 4.7.2 (c)

So the shortest distance is ADEH, the result is same as in Dijkstra's algorithm.

4.7.2 Open Shortest Path First (OSPF)

- OSPF is a link state routing protocol. OSPF is based on the distributed map concept all nodes have a copy of the network map, which is regularly updated. Each node contains a routing directory database. This database contains informations about the routers interfaces that are operable, as well as status

Scanned with CamScanner

information about adjacent routers. This information is periodically broadcast to all routers in the same domain.

- The OSPF computes the shortest path to the other routers. OSPF protocol is now widely used as the interior router protocol in TCP/IP networks. OSPF computes a route through the internet that incurs the least cost based on a user-configurable metric of cost. The user can configure the cost to express a function of delay, data rate, or other factors. OSPF is able to equalize loads over multiple equal cost paths.
- OSPF is classified as an Internal Gateway Protocol (IGP) because it support routing within one autonomous system only. The exchange of routing information between autonomous systems is the responsibility of another protocol an External Gateway Protocol (EGP). OSPF can support one or many networks.
- Following is the **features** of the OSPF.
  1. OSPF supports multiple circuit load balancing because it can store multiple routes to a destination.
  2. OSPF can converge very quickly to network topology change.
  3. OSPF support multiple metrics.
  4. OSPF is not susceptible to routing loops.
  5. OSPF support for variable length subnetting by including the subnet mask in the routing message.
- OSPF introduces a two level hierarchy for improving scalability. It allows an AS to be partitioned into several groups called areas, that are interconnected by a central backbone area as shown in the Fig. 4.7.3.
- An area is identified by a 32-bit number known as the area ID. The backbone area is identified with area ID 193.0.0.0 The information from other area is summarized by area border router that have connections to multiple areas.
- OSPF uses four types of routers.
  1. An internal router is a router with all its links connected to the networks within the same area.
  2. An area border router is a router that has its links connected to more than one area.
  3. A backbone router is a router that has its links connected to the backbone.
  4. An Autonomous System Boundary Router (ASBR) is a router that has its links connected to another autonomous system.
- As shown in the Fig. 4.7.3 routers R_1, R_2 and R_7 are internal routers. Routers R_3, R_6, R_8 are area border routers. Routers R_3, R_4, R_5, R_6, R_8 are backbone routers. Router R_4 is an ASBR.

Scanned with CamScanner

- A hello protocol allows neighbours to be discovered automatically. Two routers are said to be neighbours if they have an interface to a common network. The OSPF protocol runs directly over IP, using IP protocol 89. The header format for OSPF is shown in the Fig. 4.7.4.

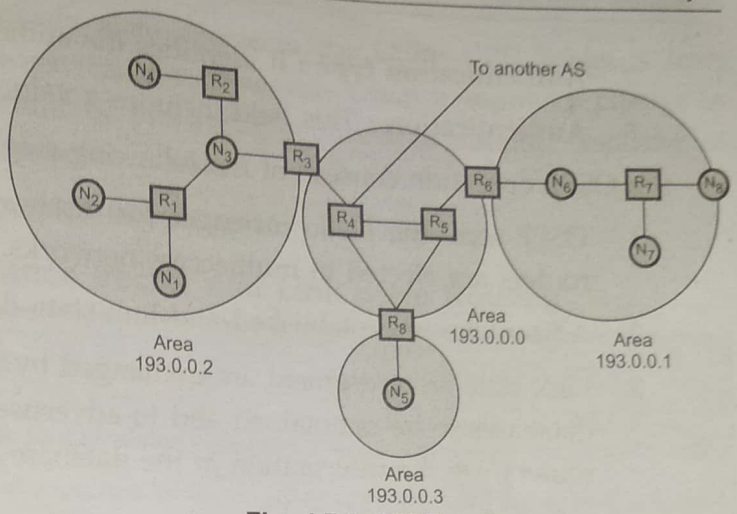


Fig. 4.7.3 OSPF areas

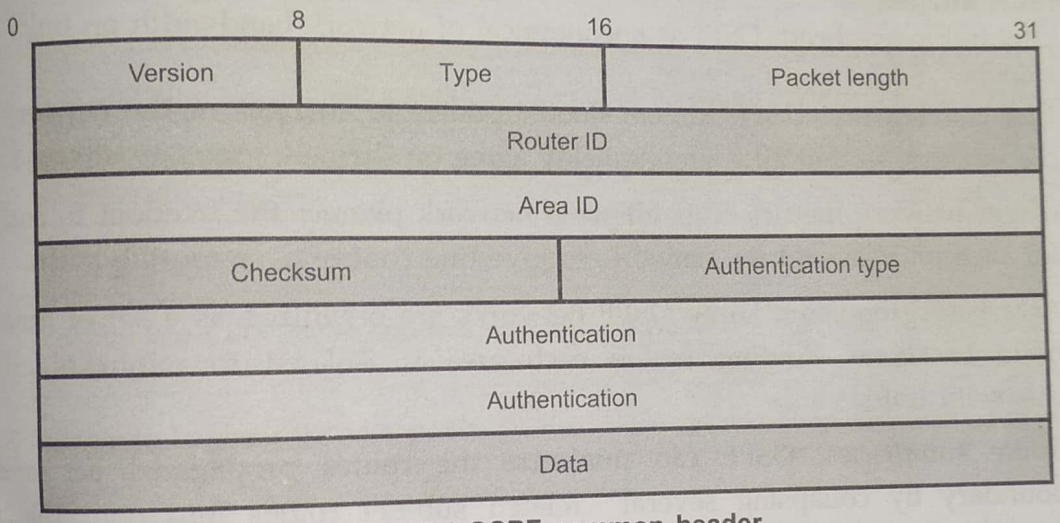


Fig. 4.7.4 OSPF common header

- OSPF header analysis is given below :
  1. **Version :** This field specifies the protocol version.
  2. **Type :** This field indicates messages as one of the following type.
     a. Hello b. Database description
     c. Link status d. Link status update
     e. Link status acknowledgement.
  3. **Packet length :** This field specifies the length of OSPF packet in bytes, including the OSPF header.
  4. **Router ID :** It identifies the sending router. This field is typically set to the IP address of one of its interfaces.
  5. **Area ID :** This field identifies the area this packet belongs to (Transmitted).
  6. **Checksum :** The checksum field is used to detect errors in the packet. The checksum is performed on the entire packet.

Scanned with CamScanner

7. **Authentication type :** It identifies the authentication type that is used.
8. **Authentication :** This field includes a value from the authentication type.

The OSPF cperation consists of the following stages.

1. OSPF send the Hello messages for discovering the neighbours and designated routers are elected in multiaccess networks.
2. Adjacencies are established and link state databases are synchronized.
3. Link state advertisement are exchanged by adjacent routers to allow topological databases to be maintained and to advertise inter area and inter AS routes. The routers use the information in the database to generate routing tables.

OSPF Advantages

1. Low traffic overhead. OSPF is enconomical of network bandwidth on links between routers.
2. Fast convergence. OSPF routers flood updates to changes in the network around the internet, so that all routers quickly agree on the new topology after a failure.
3. Larger network metrics. This allows a network planner the freedom to assign costs for each path around the network, to give fine control over routing paths.
4. Area based topology. Large OSPF networks are organized as a set of areas linked by a backbone. Routing within each area is isolated to minimize cross area discovery traffic.
5. Route summaries. OSPF can minimize the routes propagated across an area boundary by collapsing several related sub-net routes into one. This reduces routing table sizes, and increases the practical size of a network.
6. Support for complex address structures. OSPF allows variable size sub-netting within a network number, and sub-nets of a network number to be physically disconnected. This reduces waste of address space, and makes changing a network incrementally much easier.
7. Authentication. OSPF supports the use of passwords for dynamic discovery traffic, and checks that paths are operational in both directions. The main use for this is to prevent misconfigured routers from "poisoning" the routing tables throughout the internet.

OSPF Disadvantages

1. Memory overhead. OSPF uses a link state database to keep track of all routers and networks within each attached area. With a complex topology, this database can be much larger than the corresponding routing pool, and may limit the maximum size of an area.

Scanned with CamScanner

2. Processor overhead. During steady state operation the OSPF CPU usage is low, mainly due to the traffic between routers. However, when a topology change is detected, there is a large amount of processing required to support flodding of changes, and recalculation of the routing table.
3. Configuration. OSPF can be complex to configure.

4.7.3 Difference between Distance Vector and Link State Routing

Sr. No.	Distance vector	Link state
1.	Bellman-ford algorithm used to calculate the shortest cost path.	Dijkstra's algorithm used to calculate link state cost.
2.	Sends message to their neighbors.	Sends message to every other node in the network.
3.	It is decentralized routing algorithm.	It is centralized global routing algorithm.
4.	Sends larger updates only to neighbouring routers.	Send small updates every where.
5.	Protocol example - RIP	Protocol example - OSPF and BGP.
6.	Require less CPU power and less memory space.	Require more CPU power and more memory space.
7.	Simple to implement and support.	Expensive to implement and support.

University Questions

1. *Explain and compare distance vector routing and link state routing algorithm.* **GTU : Dec.-10, Marks 7**
2. *Explain link state routing algorithm in detail.* **GTU : June-11, Marks 7**
3. *Explain link state routing with all the five steps in detail.* **GTU : May-12, Marks 7**
4. *Write about OSPF (Open Shortest Path First). Which four classes of routers are distinguished by OSPF ?* **GTU : May-12, Marks 7**
5. *Explain the link state routing protocol with suitable example.* **GTU : Winter-12, Marks 7**
6. *Explain in detail OSPF.* **GTU : Summer-13, Marks 7**
7. *Explain shortest path routing protocol with suitable example.* **GTU : Winter-14, Marks 7**
8. *Explain shortest path routing algorithm.* **GTU : Dec.-11, Marks 5**
9. *What is routing ? Explain shortest path routing with example.* **GTU : Winter-13, Marks 7**

Scanned with CamScanner

4.8 Hierarchical Routing

GTU : Summer-14

- 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 other router, so the routing will have to be done hierarchically. When this routing is used, the routers are divided into regions. It contains all the details about how to route packets to destinations within its own region.

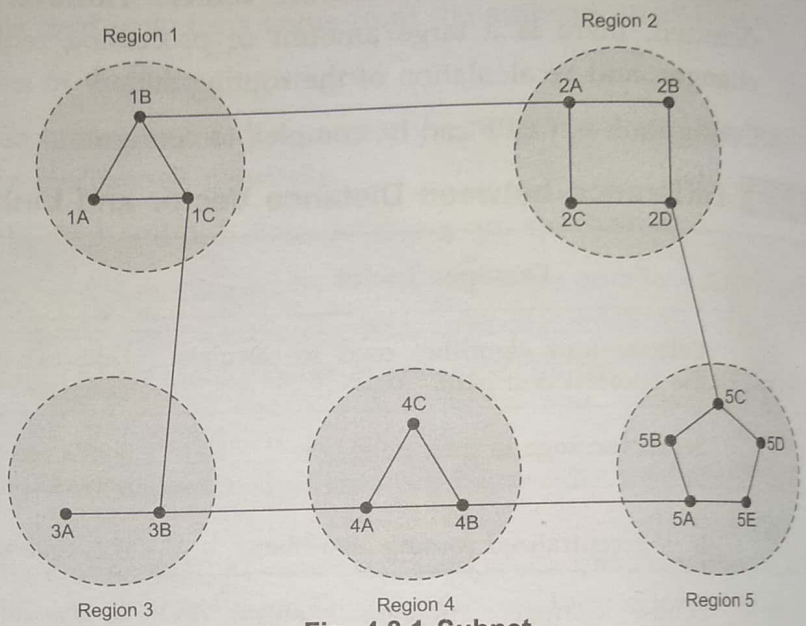


Fig. 4.8.1 Subnet

- Some time, two-level hierarchy may be insufficient. It is necessary to group the regions into clusters, the clusters into zones and zones into groups and so on. Fig. 4.8.1 shows routing in a two-level hierarchy.

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	5

Full table for 1A

Dest	Line	Hops
1A	-	-
1B	1B	1

Scanned with CamScanner

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	5
5B	1C	6
5C	1B	5
5D	1C	7
5E	1C	6

- For router 1A has 17 entries in full routing table. When hierarchical routing is used, only 7 entries valid for 1A. There are entries for local routers but all other regions have been condensed into a single router. Hierarchical routing increases the saving the table space.

University Question

1. *Explain hierarchical routing.* **GTU : Summer-14, Marks 4**

4.9 Flooding

- This technique requires no network information. A packet is sent by a source node to all its adjacent nodes. At each node, every incoming packet is retransmitted on every outgoing links, except the link that it arrived from.
- Flooding generates large number of duplicate packets. One way to prevent this is for each node to renumber the identity of those packets it has already sent. When duplicate packets arrive they are discarded.
- One such measure is to have a hop counter contained in the header of each packet, which is decremented at each hop. When the count reaches zero, the packet is discarded. The counter is set to maximum value i.e. diameter of the subnet.

Selective flooding

- In selective flooding, the routers do not retransmit every incoming packets on all links but only on those links that are in the right direction.

Scanned with CamScanner

- The main disadvantage of flooding is the total traffic load that it generates, is directly proportional to the connectivity of the network. Also flooding requires much large bandwidth.
- The advantage of flooding is that it is highly robust. This property finds application in military network that is subjected to extensive damage and in distributed database applications where it is necessary to update database concurrently.
- Fig. 4.9.1 shows the flooding process for the subnet.

A) Subnet :

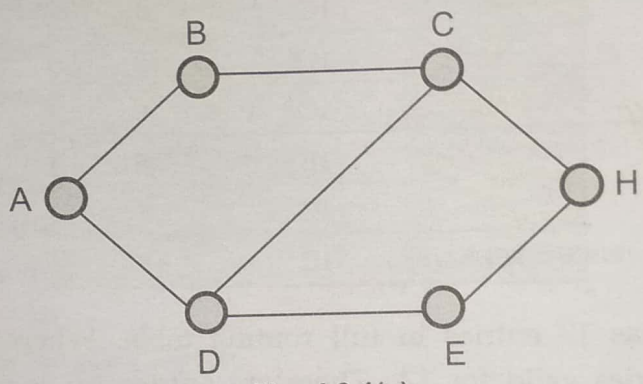


Fig. 4.9.1(a)

B) First hop :

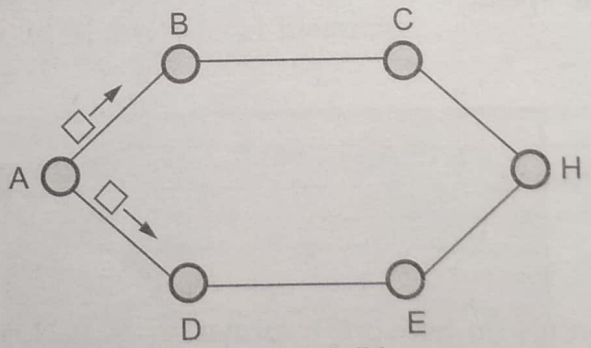


Fig. 4.9.1 (b)

C) Second hop :

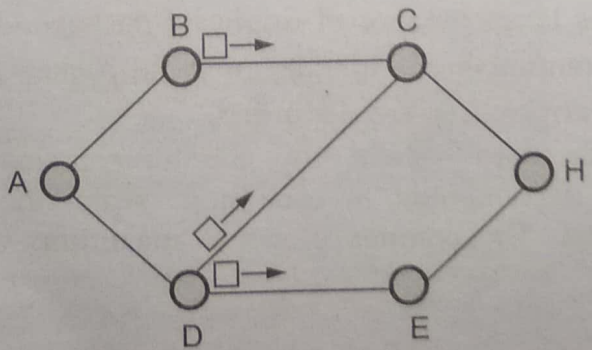


Fig. 4.9.1 (c)

Scanned with CamScanner

D) Third hop :

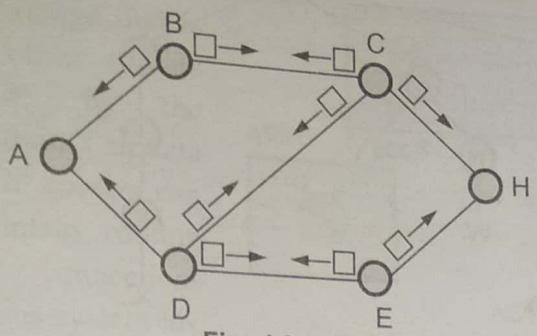


Fig. 4.9.1 (d)

4.10 Broadcast Routing

- Transmitting data to the multidestinations simultaneously called broadcasting. Various methods of broadcasting are as follows :
  (i) Flooding (ii) Multidestination routing (iii) Reverse path forwarding.

i) Flooding

This technique is already discussed in previous sections.

ii) Multidestination routing

In this technique, each packet contains entire destination addresses. When a packet arrives at a router, the router checks the addresses and selects proper links for transmission. Router generates new copy of packets for each links with selected destination addresses. After few hops each packet will carry only one destination address and it is just as a normal packet. This process is just like separately addressed packets.

iii) Reverse path forwarding

- The broadcast packet is transmitted by a source if arrives at a router, the router checks the packet whether it is from preferred path and router sends it on the best route path. A tree like structure is formed by reverse path forwarding.
- The main advantage of reverse path forwarding is that it is efficient and simple to implement. To maintain destination address by router is not required. Following Fig. 4.10.1 illustrates the reverse path forwarding. (See Fig. 4.10.1 on next page).
- During the first hop, A sends packets to B, C, D, E as indicated in Fig. 4.11.1. Each packets arrived on preferred path to A, so indicated by a circle around the letter. On the second hop, seven packets are generated, two from routers B, C, E and one from D. The packets arrived on preferred paths are then generates further packets. Packets received on F, G, H, J are on preferred paths. In the third hop, thirteen packets are generated, packet W is not on preferred path so it is rejected. This process continues and after specific numbers of hops the broadcasting terminates.

Scanned with CamScanner

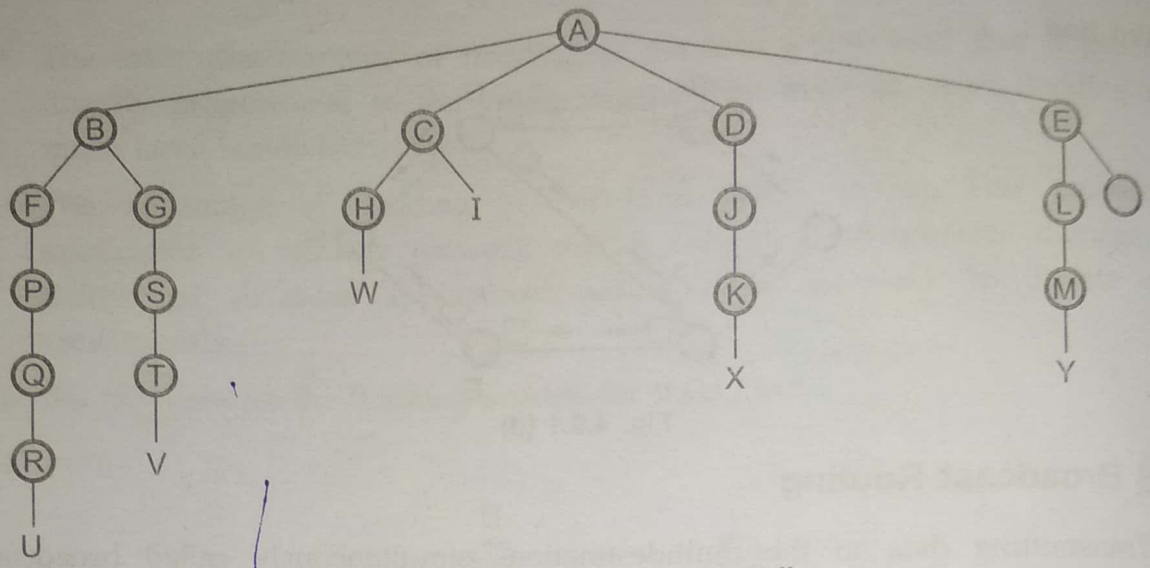


Fig. 4.10.1 Reverse path forwarding

4.11 Border Gateway Protocol (BGP)

- The purpose of an exterior gateway protocol is to enable two different Autonomous System (AS) to exchange routing information so that IP traffic can flow across the autonomous system border. BGP was developed for use in conjunction with internets that employ the TCP/IP protocol suite. The BGP is an interdomain routing protocol that is used to exchange network reachability information among BGP routers (Also called BGP speakers). Each BGP speaker establishes a TCP connection with one or more BGP speakers (routers). Two routers are considered to be neighbours if they are attached to the same subnetwork. If the two routers are in different autonomous systems, they may wish to exchange routing information.
- BGP performs three functional procedures.
  1. Neighbour acquisition
  2. Neighbour reachability
  3. Network reachability.
- Neighbour acquisition procedures used for exchanging the routing information between two routers in different Autonomous Systems (AS). To perform neighbour acquisition, one router sends an open message to another. If the target router accepts the request, it returns a keepalive message in response.
- Once a neighbour relationship is established, the neighbour reachability procedure is used to maintain the relationship. Both sides needs to be assured that the other side still exists and is still engaged in the neighbour relationship. For this purpose, both routers send keepalive messages to each other. Both sides router maintains a database of the subnetworks that it can reach and the perferred route for reaching that subnetwork.

Scanned with CamScanner

- If the database changes, router issues an update message that is broadcast to all other routers implementing BGP. By the broadcasting of these updata message, all the BGP routers can build up and maintain routing information. BGP connections inside an autonomous system are called internal BGP (iBGP) and BGP connections between different autonomous systems are called external BGP (eBGP).

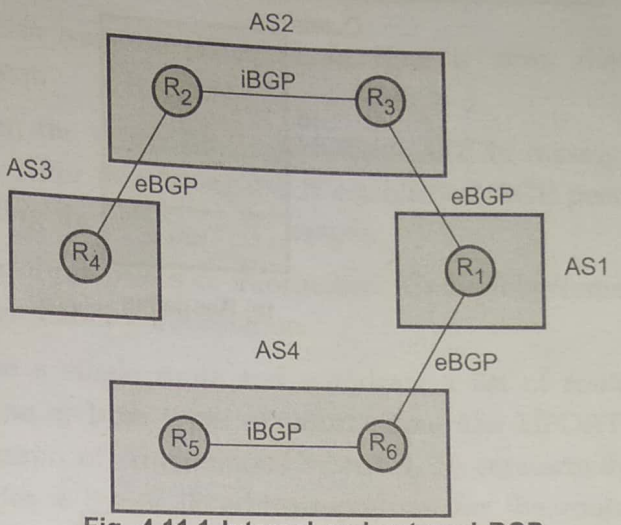


Fig. 4.11.1 Internal and external BGP

- Fig. 4.11.1 shows the internal and external BGP.

BGP messages : Header of the all BGP messages is fixed size that identifies the message type. Fig. 4.11.2 shows the BGP message header format.

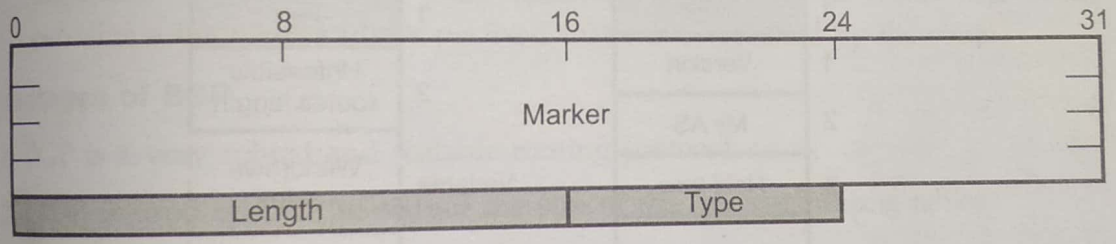


Fig. 4.11.2 BGP header format

1. **Marker :** Marker field is used for authentication. The sender may insert value in this field that would be used as part of an authentication mechanism to enable the recepient to verify the identity of the sender.
2. **Length :** This field indicates the total length of the message in octets, including the BGP header. Value of the length must be between 19 and 4096.
3. **Type :** Type field indicates type of message. BGP defines four message type.

   a) OPEN b) UPDATE c) NOTIFICATION d) KEEPALIVE

- Following Fig. 4.11.3 shows the four types of BGP message formats.
- To acquire a neighbour, a router first opens a TCP connection to the neighbour router of interest. It then sends the open message. This message identifies the AS (Autonomous System) to which the sender belongs and provides the IP address of the router. It also includes a Hold time parameter. If the recipient is prepared to open a neighbour relationship, it calculate a value of Hold Timer that is the minimum of its Hold Time in the open message. This calculated value is the

Scanned with CamScanner

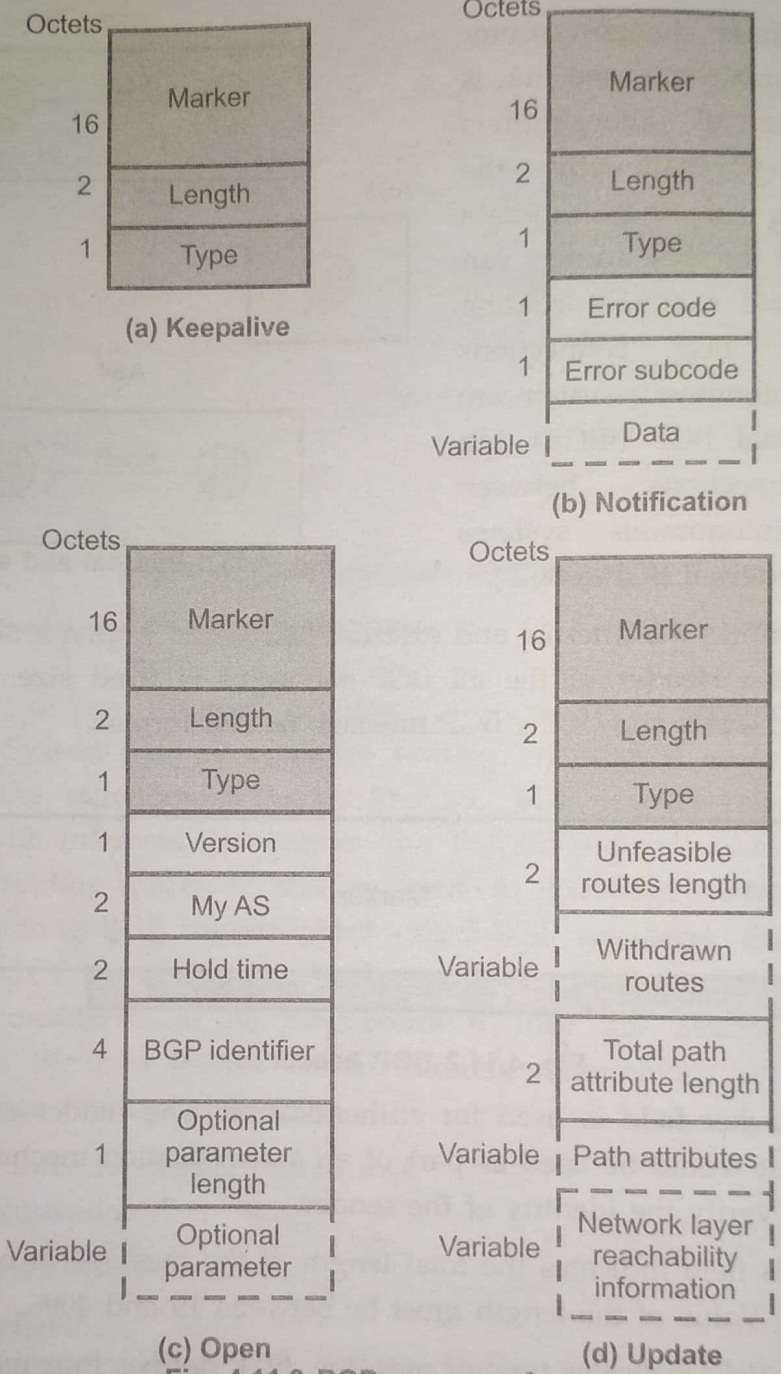


Fig. 4.11.3 BGP message format

maximum number of seconds that may elapse between the receipt of successive keepalive and update message by the sender.

- The KEEPALIVE message is just the BGP header with the type field set to 4. The KEEPALIVE messages are exchanged often enough as to not cause the hold timer to expire. A recommended time between successive KEEPALIVE messages is one-third of the hold time interval. This value ensures that KEEPALIVE messages arrive at the receiving router almost always before the hold timer expires even if

Scanned with CamScanner

the transmission delay of a TCP is variable. If the hold time is zero, then KEEPALIVE messages will not be sent.

- When a BGP router detects an error, the router sends a NOTIFICATION message and then close the TCP connection. After the connection is established, BGP peers exchange routing information by using the UPDATE messages.
- The UPDATE messages may contain three pieces of information. Unfeasible routes, path attributes and network layer reachability information
- An UPDATE message can advertise a single route and withdraw a list of route. An update message may contain one or both types of information. The UPDATE messages are used to construct a graph of Autonomous System (AS) connectivity. The withdrawn routes field provides a list of IP address prefixes for the routes that need to be withdrawn from BGP routing tables. The unfeasible routes length field indicates the total length of the withdrawn routes field in actets.
- An UPDATE message can withdraw multiple unfeasible routes from service. A BGP router uses Network Layer Reachability Information (NLRI), the total path attributes length and the path attributes to advertise a route. The NLRI field contains a list of IP address perfixed that can be reached by the route.

Advantages of BGP

1. BGP is a very robust and scalable routing protocol.
2. CIDR is used by BGP to reduce the size of the Internet routing tables.
3. BGP easily solves the count-to-infinity problem.

Disadvantages of BGP

1. BGP is complex.
2. BGP routes to destination networks, rather than to specific hosts or routers.

4.12 Comparison of RIP and OSPF

Sr. No.	RIP	OSPF
1.	RIP is easy to configure.	OSPF is complicated to configure and requires network design and planning.
2.	An end system (a system with only one network interface) can run RIP in passive mode to listen for routing information.	OSPF does not have a passive mode.
3.	RIP may be slow to adjust for link failures.	OSPF is quick to adjust for link failures.

Scanned with CamScanner

4.	RIP generates more protocol traffic than OSPF.	OSPF generates less protocol traffic than RIP.
5.	RIP is not well suited to large networks, because RIP packet size increases as the number of networks increases.	OSPF works well in large networks.
6.	RIP is distance vector routing protocol.	OSPF is link state routing protocol.

4.13 Multicast Routing

GTU : Summer-14

- To send messages to well defined groups that are numerically large is size but small compared to the network as a whole. Sending message to such a group is called **multicasting** 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.
- To do multicasting routing, each router computes a spanning tree occuring all other routers. This is shown in Fig. 4.13.1. (See Fig. 4.13.1 on next page).
- Various ways of constructing the spanning tree are possible. The simplest one can be used if link state routing is used and each router is aware of the complete topology, including which hosts belong to which groups.
- 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 host periodically.
- Either way, routers learn about which of their hosts are in which groups. Routers tell their neighbours, so the information propagates through the subnet. (See Fig. 4.13.1 on next page).

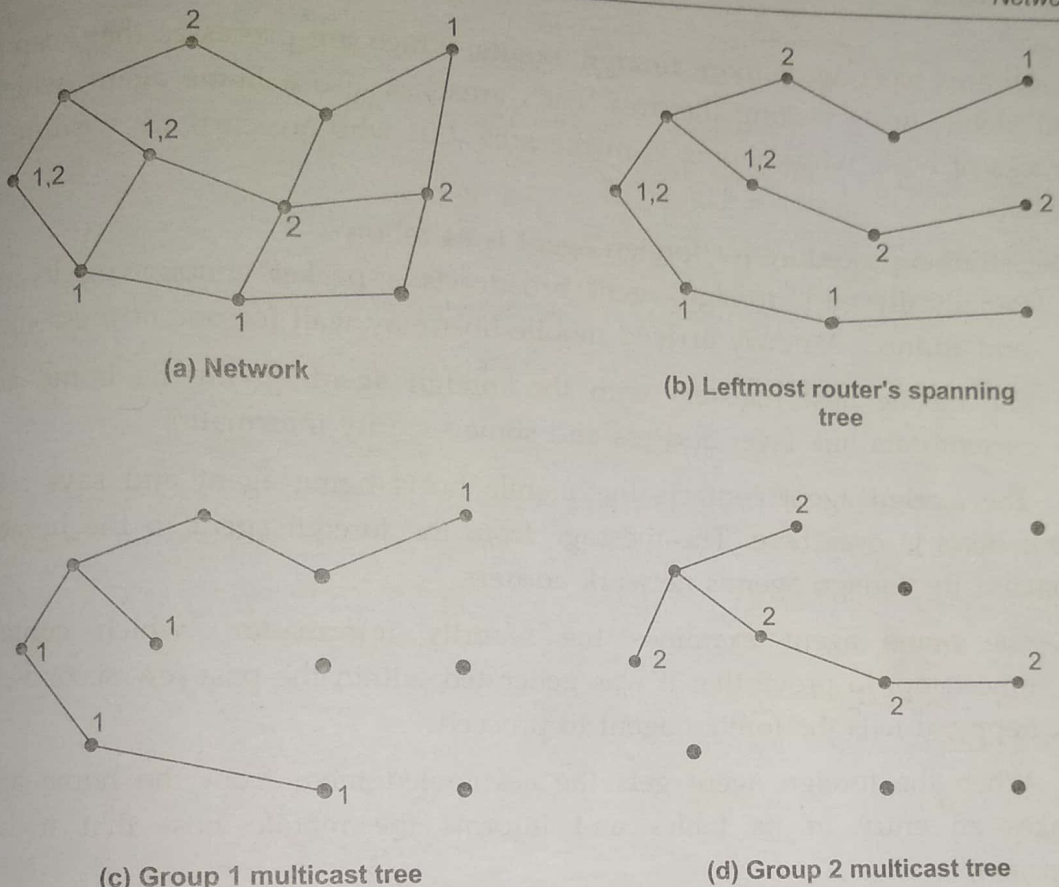
University Question

1. *Explain multicast routing.* GTU : Summer-14, Marks 3

4.14 Routing for Mobile Hosts

- WAN consisting of routers and hosts. Hosts that never move are said to be stationary. They are connected to the network by copper wires or fibre optics.
- **Migratory hosts** are basically stationary hosts who move from one fixed site to another from time to time but use the network only when they are physically connected to it.
- **Roaming hosts** actually compute on the run and want to maintain their connections as they move around.

Scanned with CamScanner



(a) Network

(b) Leftmost router's spanning tree

(c) Group 1 multicast tree

(d) Group 2 multicast tree

Fig. 4.13.1 (a,b,c,d) Multicast routing

- All hosts are assumed to have a permanent home location that never changes. Hosts also have a permanent home address that can be used to determine their home locations.
- Fig. 4.14.1 shows the model of the world which is divided into small units. This small units is called as area. Area is typically LAN or wireless cell.

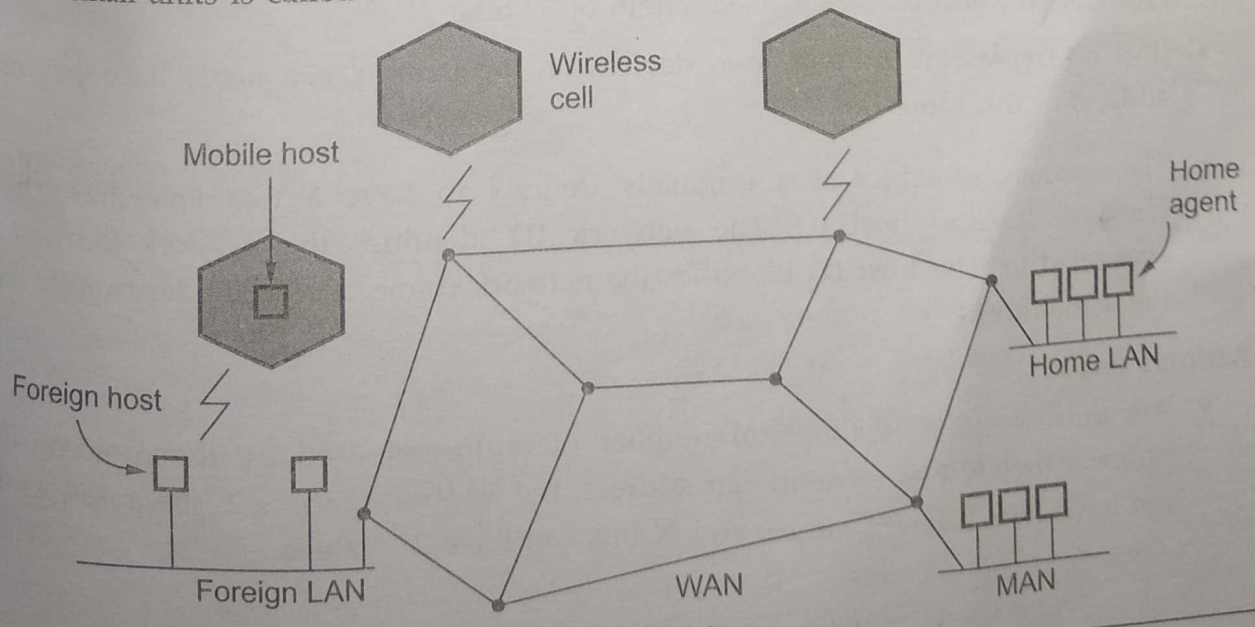


Fig. 4.14.1

Scanned with CamScanner

- Each area has one or more **foreign agents**, which are processes that keep track of all mobile hosts visiting the area. Each area has also a **home agent**, which keeps tracks of hosts whose home is in the area, but who are currently visiting another area.
- Registration procedure for foreign agent is as follows.
  1. Periodically each foreign agent broadcasts a packet announcing its existence and address. A newly arrived mobile host may wait for one of these messages.
  2. The mobile host registers with the foreign agent, giving its home address, current data link layer address and some security information.
  3. The foreign agent contacts the mobile hosts home agent and says : One of your hosts is over here. The message from the foreign agent to the home agent contains the foreign agent's network address.
  4. The home agent examines the security information, which contains a timestamp, to prove that it was generated within the past few seconds. If it is happy, it tells the foreign agent to proceed.
  5. When the foreign agent gets the acknowledgment from the home agent, it makes an entry in its tables and informs the mobile host that it is now registered.
- When a packet is sent to a mobile host, it is routed to the host's home LAN because that is what the address says should be done.

4.15 IPv4 Addresses

GTU : Dec.-10,11, Winter-12,14, Summer-14

- IP corresponds to the network layer in the OSI reference model and provides a connectionless best effort delivery service to the transport layer. An Internet Protocol (IP) address has a fixed length of 32 bits.
- IPv4 addresses are unique. Two devices on the internet can never have the same address at the same time.
- The address structure was originally defined to have a two level hierarchy : Network ID and host ID. The **network ID** identifies the network the host is connected to. The **host ID** identifies the network connection to the host rather than the actual host.

Address space

- An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2^N because each bit can have two different values and N bits can have 2^N values.

Scanned with CamScanner

- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4, 294, 967, 296.
- IP addresses are usually written in **dotted decimal** notation so that they can be communicated conveniently by people. The address is broken into four bytes with each byte being represented by a decimal number and separated by a dot.
- For example, an IP address of 10000000 10000111 01000100 00000100 is written as 128.135.68.4 in dotted-decimal notation.

  The address 193.32.216.9 in binary notation is

  11000001 00100000 11011000 00001001
- An IP address is a numeric identifier assigned to each machine on an IP network. IP address is a software address, not a hardware address, which is hard-coded in the machine or NIC. An IP address is made up of 32 bits of information. These bits are divided into four parts containing 8 bit each.
- There are three method for depicting an IP address.
  1. Dotted-decimal as in 131.57.30.57
  2. Binary, as, 10000010.00111001.00011110.00111000
  3. Hexadecimal, as in 8B.39.C2.43
- The 32-bit IP address is a structured or hierachical address. The network address uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. The IP address 131.57.30.57, the 131.57 is the network address and 30.57 is the node address. The node address is assigned to and uniquely identifies, each machine on a network.

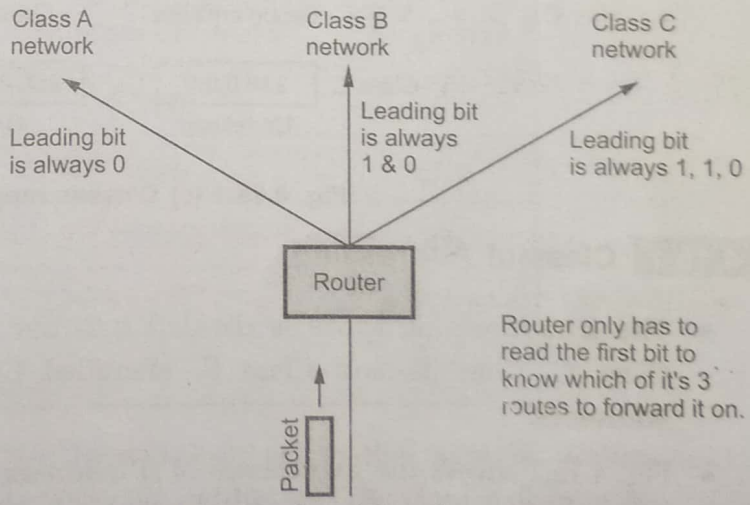


Fig. 4.15.1 (a) Leading bits of a network address

The router might able to speed a packet on its way after reading only the first bits of address. The format used for IP address are shown in Fig. 4.15.1 (b).

Scanned with CamScanner

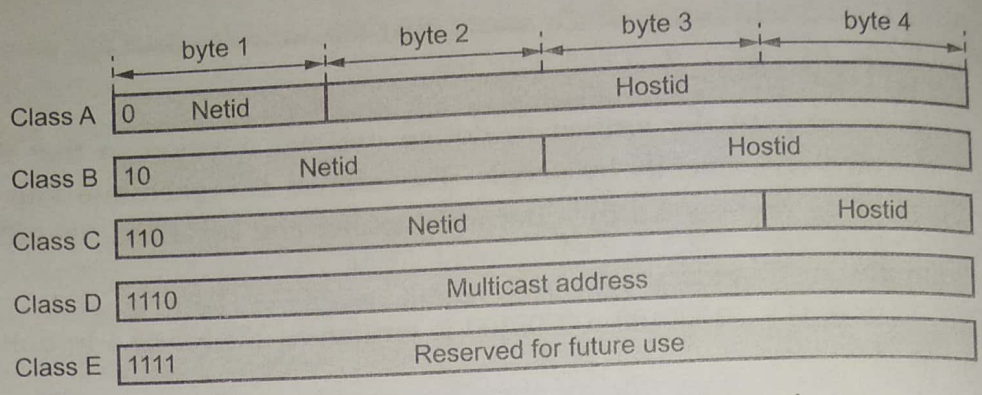


Fig. 4.15.1 (b) Internet classes (IP addresses)

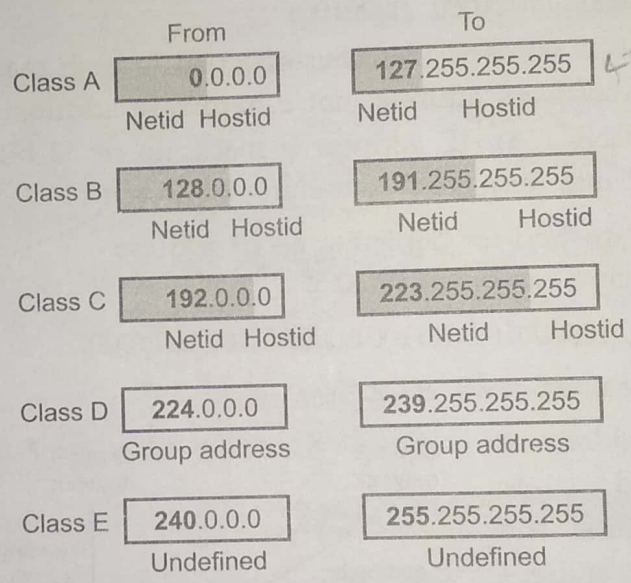


Fig. 4.15.1 (c) Classes range of IP

4.15.1 Classful Addressing

- The IP address structure is divided into five address classes : Class A, Class B, Class C, Class D and Class E, identified by the most significant bits of the addresses.
- Fig. 4.15.2 shows the five classes of IP addresses. (See Fig. 4.15.2 on next page).
- Class D addresses are used for multicast services that allow a host to send information to a group of hosts simultaneously. Class E addresses are reserved for future use.
- Class A addresses were designed for large organizations with a large number of attached hosts or routers.
- Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.

Scanned with CamScanner

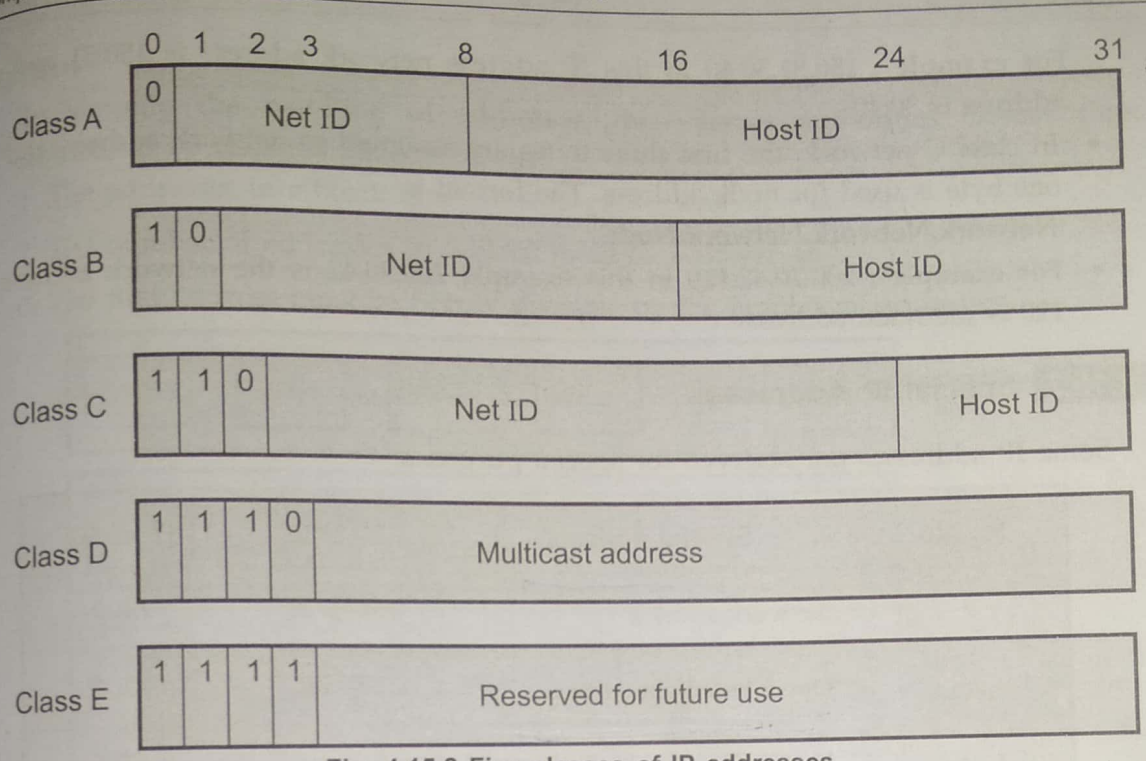


Fig. 4.15.2 Five classes of IP addresses

- One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size.

Class	Number of blocks	Block size
A	128	16777216
B	16384	65536
C	2097152	256
D	1	268435456
E	1	268435456

- In a class A network, the first byte is assigned to the network address and the remaining three bytes used for the node addresses. The class A format is **Network.Node.Node.Node**

  For example : 14.28.101.120 in this IP address 14 is the network address and 28.101.120 is the node address.
- In class B network, the first two bytes are assigned to the network address and the remaining two bytes are used for node addresses. The format is **Network.Network.Node.Node**

Scanned with CamScanner

For example : 150.51.30.40 in this IP address network address is 150.51 and node address is 30.40.

- In class C network, the first three bytes are assigned to network address and only one byte is used for node address. The format is
  Network.Network.Network.Node
- **For example :** 200.20.42.120 in this example 200.20.42 is the network address and 120 is the node address.

4.15.2 Special IP Addresses

Some IP addresses are reserved for special purposes.

Sr. No.	Special address	Net ID	Host ID
1.	Network address	Specific	All 0
2.	Direct broadcast address	Specific	All 1
3.	Limited broadcast address	All 1s	All 1
4.	This host on this network	All 0s	All 0
5.	Loopback address	127.	Any
6.	Specific host on this network	All 0s	Specific

4.15.3 Classless Addressing

- In classless addressing variable length blocks are assigned that belong to no class. In this, the entire address space is divided into blocks of different sizes. An organization is granted a block suitable for its purposes.
- Fig. 4.15.3 shows the architecture of classless addressing.

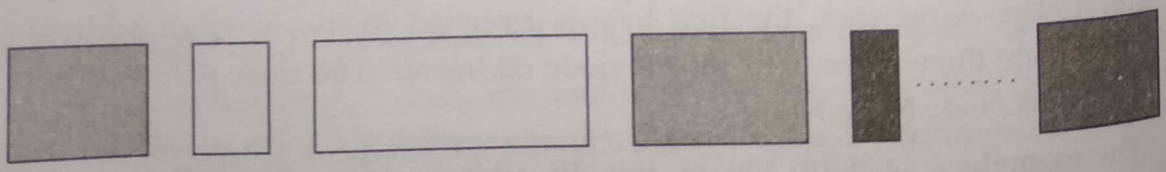


Fig. 4.15.3 Architecture of classless addressing

In classless addressing, when an entity, small or large, needs to be connected to the internet it is granted a block of addresses. The size of the block varies based on the nature and size of the entity.

Scanned with CamScanner

Restriction

- To simplify the handling of addresses, the internet authorities impose three restrictions on classless address blocks.
  1. The addresses in a block must be contiguous, one after another.
  2. The number of addresses in a block must be a power of 2.
  3. The first address must be evenly divisible by the number of addresses.

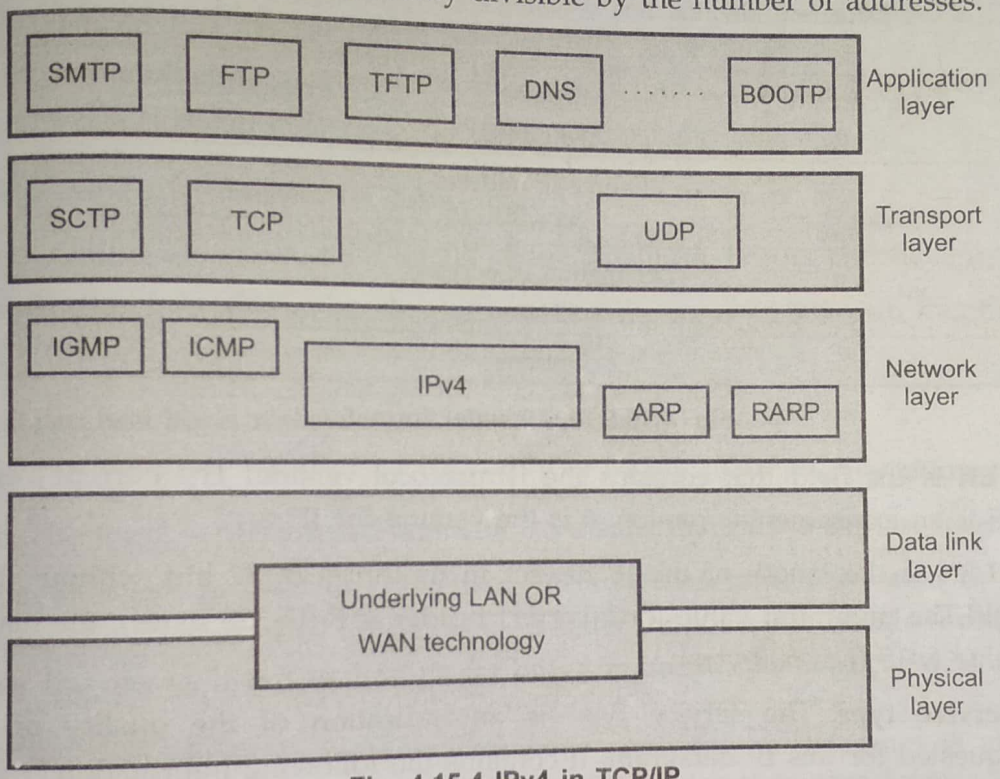


Fig. 4.15.4 IPv4 in TCP/IP

- In IPv4 addressing, a block of addresses can be defined as x.y.z.t/n in which x.y.z.t defines one of the addresses and the /n define the mask. The address and the /n notation completely define the whole block.
- IPv4 is the delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol.
- IPv4 is also a connectionless protocol for a packet switching network that uses the datagram approach.
- Fig. 4.15.4 shows the positions of IPv4 in TCP/IP protocol suite.

4.15.4 Header Format

- Packets in the IPv4 layer are called datagrams. A datagram is a variable length packet consisting of two parts : Header and data.

Scanned with CamScanner

- Fig. 4.15.5 shows IPv4 header format

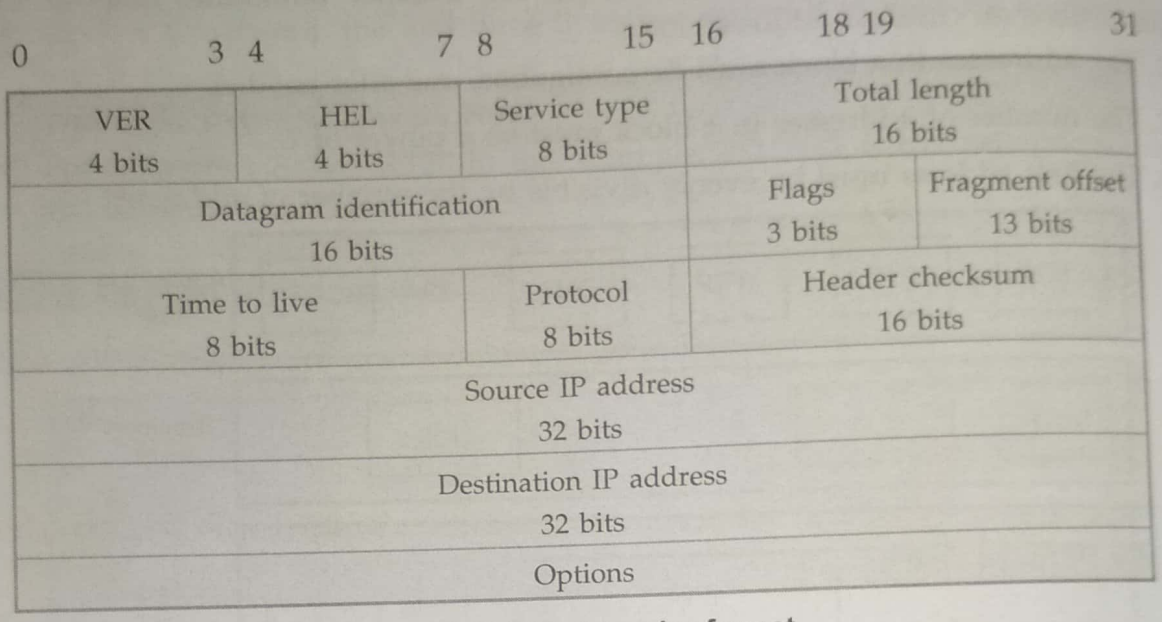


Fig. 4.15.5 IPv4 header format

1. **VER** is the field that contains the IP protocol version. The current version is 4.5 is an experimental version. 6 is the version for IPv6.
2. **HLEN** is the length of the IP header in multiples of 32 bits without the data field.The minimum value for a correct header is 5 (i.e. 20 bytes), the maximum value is 15 (i.e., 60 bytes).
3. **Service type** The service type is an indication of the quality of service requested for this IP datagram. It contains the following information.

Precedence	Types of service	R

Precedence specifies the nature / priority :

000	Routine
001	Priority
010	Immediate
011	Flash
100	Flash override
101	Critical
110	Internetwork control
111	Internetwork control

Scanned with CamScanner

TOS specifies the type of service value :

TOS bits	Description
1000	Minimize delay
0100	Maximum throughout
0010	Maximize reliability
0001	Minimize monetary cose
0000	Normal service

The last bit is reserved for future use.

4. **Total length** specifies the total length of the datagram, header and data, in octets.
5. **Identification** is a unique number assigned by the sender used with fragmentation.
6. **Flags** contain control flags :
   a. The first bit is reserved and must be zero;
   b. The 2nd bit is DF (Do not Fragment), 0 means allow fragmentation;
   c. The third is MF (More Fragments), 0 means that this is the last fragment.
7. **Fragment offset** is used to reassemble the full datagram. The value in this field contains the number of 64-bit segments (header bytes are not counted) contained in earlier fragments. If this is the first (or only) fragment, this field contains a value of zero.
8. **TTL** (Time To Live) specifies the time (in seconds) the datagram is allowed to travel. In practice, this is used as a hop counter to detect routing loops.
9. **Protocol number** indicates the higher level protocol to which IP should deliver the data in this datagram. E.g., ICMP = 1; TCP = 6; UDP = 17.
10. **Header checksum** is a checksum for the information contained in the header. If the header checksum does not match the contents, the datagram is discarded.
11. **Source/Destination IP addresses** are the 32-bit source/destination IP addresses.
12. **IP options** is a variable-length field (there may be zero or more options) used for control or debugging and measurement. For instance :
    a. The **loose source routing** option provide a means for the source of an IP datagram to supply explicit routing information;
    b. The **timestamp** option tell the routers along the route to put timestamps in the option data.
13. **Padding** is used to ensure that the IP header ends on a 32 bit boundary. The padding is zero.

Scanned with CamScanner

4.15.5 IP Fragmentation

- IP provides fragmentation/reassembly of datagrams. The maximum length of an IP datagram is 65,535 octets. When an IP datagram travels from one host to another, it may pass through different physical networks. Each physical network has a maximum frame size, called Maximum Transmission Unit (MTU), which limits the datagram length.
- A fragment is treated as a normal IP datagram while being transported to their destination. Thus, fragments of a datagram each have a header. If one of the fragments gets lost, the complete datagram is considered lost. It is possible that fragments of the same IP datagram reach the destination host via multiple routes. Finally, since they may pass through networks with a smaller MTU than the sender's one, they are subject to further fragmentation. Fig. 4.15.6 shows the MTU.

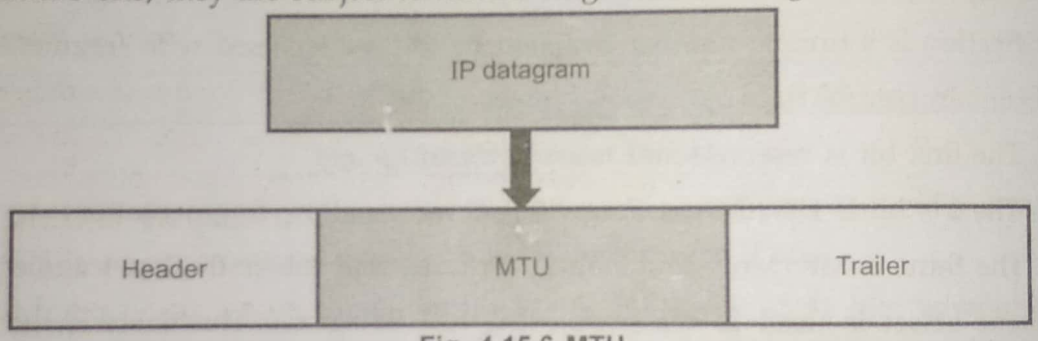


Fig. 4.15.6 MTU

Fragmentation process

- The DF flag bit is checked to see if fragmentation is allowed. If the bit is set, the datagram will be discarded and an ICMP error returned to the originator.
- Based on the MTU value, the data field is split into two or more parts. All newly created data portions must have a length that is a multiple of 8 octets, with the exception of the last data portion. Each data portion is placed in an IP datagram.

  Fig. 4.15.7 shows the examples of fragmentation.

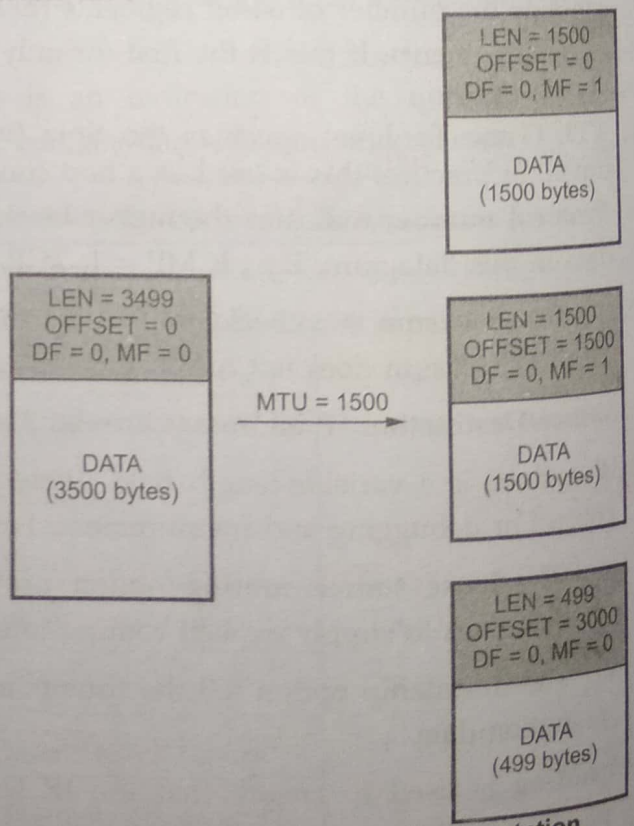


Fig. 4.15.7 Examples of fragmentation

Scanned with CamScanner

Modification to the headers of fragments :

a. The MF flag is set in all fragments except the last;

b. The fragment offset field is updated;

c. If options were included in the original datagram, they may be copied to all fragment datagram's or only the first datagram (depends on the option);

d. The header length field is set;

e. The total length field is set;

f. The header checksum is re-calculated.

- At the destination host, data are reassembled into the original datagram. The identification field set by the sending host is used together with the source and destination IP addresses in the datagram. Fragmentation does not alter this field.
- In order to reassemble the fragments, the receiving host allocates a storage buffer when the first fragment arrives. The host also starts a timer. If the timer is exceeded and fragments remain outstanding, the datagram is discarded. When subsequent fragments of the datagram arrive, data are copied into the buffer storage at the location indicated by the fragment offset field. When all fragments have arrived, the original unfragmented datagram is restored and passed to upper layers, if needed.

Problem in fragmentation

1. The end node has no way of knowing how many fragments there be. The end node has to manage enough buffer space to handle reassembly process.
2. If any fragments lost, all datagram must be discarded.
3. End node starts a timer when received the first fragment, if any fragments fails to arrive (usually 30 secs), all datagram's must be discarded.
4. Since the IP service is connectionless. No attempt is made by IP to recover these situations, through ICMP error message may be generated.

4.15.6 Options

The header of the IPv4 datagram is made of two parts : A fixed part and a variable part. Options used in IPv4 are as follows

1. **No operation** option is 1-byte option used as a filter between options.
2. **End of option** is a 1-byte option used for padding at the end of the option field.

Scanned with CamScanner

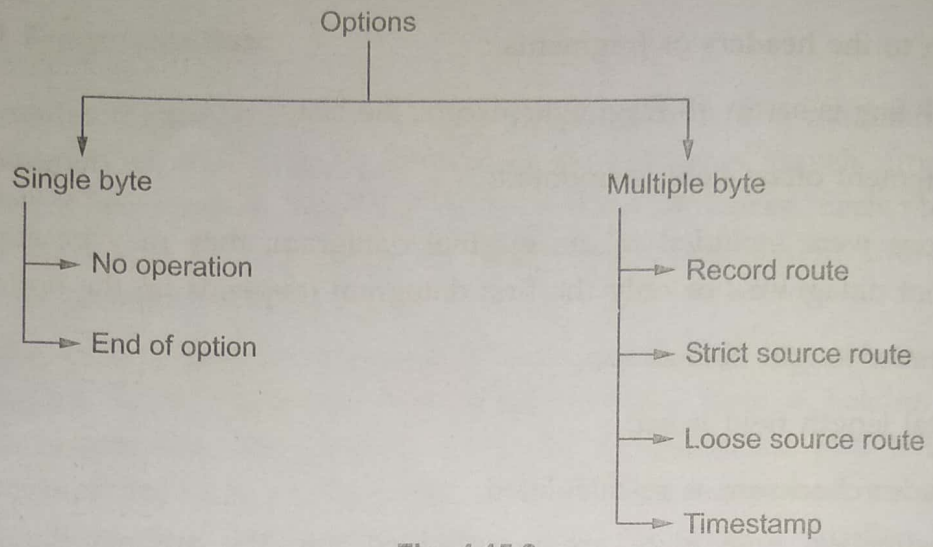


Fig. 4.15.8

3. Record route option is used to record the internet routers that handle the datagram. It can list upto nine router addresses. It can be used for debugging and management purposes.
4. Strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet.
5. Loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.
6. Timestamp option is used to record the time of datagram processing by a router.

4.15.7 Subnetting a Network

- If a organization is large or if its computers are geographically dispersed, it makes good sense to divide network into smaller ones, connected together by routers. The benefits for doing things this way include.
  1. Reduced network traffic
  2. Optimized network performance
  3. Simplified network management
  4. Facilities spanning large geographical distances.
- If Network Information Center (NIC) assign only one network address to an organization which having multiple network, that organization has a problem. A single network address can be used to refer to multiple physical networks. An organization can request individual network address for each one of its physical networks. If these were granted, there wouldn't be enough to go around for everyone.

Scanned with CamScanner

- Another problem is, if each router on the internet needed to know about each existing physical network, routing tables would be impossibly huge. This is physical overhead on the router. To solve this type of problem, the subnet addressing method is used.

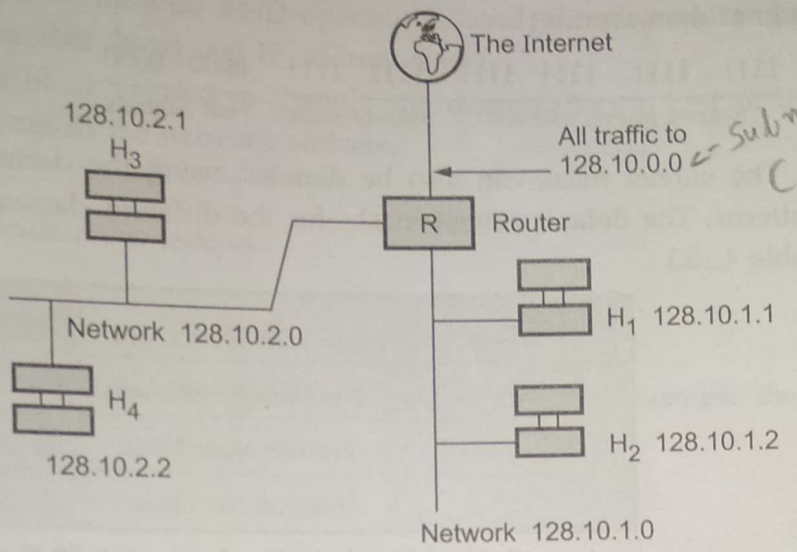


Fig. 4.15.9 Multiple network

- To allow a single network address to span multiple physical networks is called **subnet addressing** or **subnet routing** or **subnetting**. Subnetting is a required part of IP addressing.
- To understand subnet addressing, consider the next example. Consider the site has a single class B IP network address assigned to it, but the organization has two or more physical networks. Only local routers know that there are multiple physical networks and how to route traffic among them.
- In the example, the organization is using the single class B network address for two networks. For the subnet address scheme to work, every machine on the network must know which part of the host address will be used as the subnet address. This is accomplished by assigning each machine a subnet mask.
- The network administrator creates a 32-bit subnet mask comprised of ones and zeros. The ones in the subnet mask represent the positions that refer to the network or subnet addresses. The zeros represent the positions that refer to the host part of the address. Class B address format is Net.Net.Node.Node. The third byte, normally assigned as part of the host address is now used to represent the subnet address. Hence, these bit positions are represented with ones in the subnet mask. The fourth byte is the only part in example that represents the unique host address.

Subnet mask code

1 = Positions representing network or subnet addresses.

0 = Positions representing the host address.

Scanned with CamScanner

Subnet mask format

1111 1111. 1111 1111 1111 1111. 0000 0000

Network address positions *Subnet positions* *Host positions*

The subnet mask can also be denoted using the decimal equivalents of the binary patterns. The default subnet masks for the different classes of networks are as below in Table 4.15.1

Class	Format	Defult subnet mask
A	Net.Node.Node.Node	255.0.0.0
B	Net.Net.Node.Node	255.255.0.0
C	Net.Net.Net.Node	255.255.255.0

Table 4.15.1 Default subnet mask of IP address

Masking

- A process that extracts the address of the physical network from an IP address is called Masking. If we done the subnetting, then masking extracts the subnetwork address from an IP address.
- To find the subnetwork address, two method are used. There are boundary level masking and non-boundary level masking, we take one by one.
- In boundary level masking, two masking numbers are consider (i.e. 0 or 255). In non-boundary level masking other value of masking is used Apart from 0 and 255.

A. Rules for boundary level masking

1. In this mask number is either 0 or 255.
2. If the mask number is 255 in the mask IP address, then the IP address is repeated in subnetwork address.
3. If the mask number is 0(zero) in the mask IP address, then the 0 is repeated in subnetwork address.

B. Rules for non-boundary level masking

1. In this mask numbers are not 0 or 255 mask number is greater than 0 or less than 255.
2. If the mask number is 255 in the mask IP address, then the original IP address (byte) is repeated in subnetwork address.
3. If the mask number is 0 in the mask IP address, then the 0 is repeated in the subnetwork address.

Scanned with CamScanner

4. For any other mask numbers, bit-wise AND operator is used. Bit-wise ANDing is done in between mask number (byte) and IP address (byte).

- The first address in the block is used to identify the organization to rest of the Internet. This address is called the **network address.**

1. How many subnets ?

- Number of subnet is calculated as follows :

Number of subnet = 2^x

where x is the number of masked bits or the 1s (ones).

For example 11100000, the number of 1s gives us 2^3 subnets. In this example there are 8 subnets.

2. How many host per subnet ?

Number of host per subnet = $2^y - 2$

Where y is the number of unmasked bits or the 0s (zeros).

For example 11100000, the number of 0s gives us $2^5 - 2$ hosts. In this example there are 30 hosts per subnet. You need to subtract 2 for subnet address and the broadcast address.

3. What are the valid subnets ?

For valid subnet = 256 – Subnet mask = Block size. An example would be

256 – 224 = 32. The block size of a 224 mask is always 32.

Start counting at zero in block of 32 until you reach the subnet mask value and these are your subnets. 0, 32, 64, 96, 128, 160, 192, 224.

4. What is the broadcast address for each subnet ?

Our subnets are 0, 32, 64, 96, 128, 160, 192, 224, the broadcast address is always the number right before the next subnet. For example, the subnet 0 has a broadcast address of 31 because next subnet is 32. the subnet 32 has a broadcast address of 63 because next subnet is 64.

5. What are the valid hosts ?

Valid hosts are the numbers between the subnets, omitting the all 0s and all 1s. For example, if 32 is the subnet number and 63 is the broadcast address, then 32 to 63 is the valid host range. It is always between the subnet address and the broadcast address.

Example 4.15.1 *What is the sub-network address if the destination address is 200.45.34.56 and the subnet mask is 255.255.240.0 ?*

Scanned with CamScanner

Solution : Using AND operation, we can find sub-network address.

1. Convert the given destination address into binary format :

   200.45.34.56 => 11001000 00101101 00100010 00111000

2. Convert the given subnet mask address into binary format :

   255.255.240.0 => 11111111 11111111 11110000 00000000

3. Do the AND operation using destination address and subnet mask address.

   200.45.34.56 => 11001000 00101101 00100010 00111000

   255.255.240.0=> 11111111 11111111 11110000 00000000

   11001000 00101101 00100000 00000000

Subnetwork address is 200.45.32.0

Example 4.15.2 *For a network address 192.168.10.0 and subnet mask 255.255.255.224 then calculate :*

i) Number of subnet and number of host

ii) Valid subnet

Solution : Given network address 192.168.10.0 is class C address. Subnet mask address is 255.255.255.224. Here three bits is browse for subnet.

i) Number of subnet and number of host :

255.255.255.224 convert into binary => 11111111 11111111 11111111 11100000

Number of subnet $= 2^x$

$= 2^3$

$= 8$

So there are 8 subnet.

Number of host per subnet $= 2^y - 2$

$= 2^5 - 2$

$= 30$

ii) Valid subnets :

For valid subnet = 256 – Subnet mask = Block size. An example would be 256 – 224 = 32. The block size of a 224 mask is always 32.

Start counting at zero in block of 32 until you reach the subnet mask value and these are your subnets. 0, 32, 64, 96, 128, 160, 192, 224.

Scanned with CamScanner

Example 4.15.3 *Find the sub-network address for the following :*

Sr. No.	IP address	Mask
a)	140.11.36.22	255.255.255.0
b)	120.14.22.16	255.255.128.0

Solution :

a) IP address 140.11.36.22

Mask 255.255.255.0

The values of mask (i.e. 255.255.255.0) is boundary-level. So

IP address 140.11.36.22

Mask 255.255.255.0

140.11.36.0

b) IP address 120.14.22.16

Mask 255.255.128.0

The byte 1, byte 2 and byte 4 is boundary values and byte 3 is non-boundary value.

Example 4.15.4 *Find the sub-network address for the following.*

Sr. No.	IP address	Mask
a)	141.181.14.16	255.255.224.0
b)	200.34.22.156	255.255.255.240
c)	125.35.12.57	255.255.0.0

Solution :

a)

141.181.14.16 IP address

255.255.224.0 Mask

141.181.0.0 Sub-network address

b)

200.34.22.156 IP address

255.255.255.240 Mask

200.34.22.144 Sub-network address

Scanned with CamScanner

c)

125.35.12.57	IP address
255.255.0.0	Mask
125.35.0.0	Sub-network address

(i.e. 128) So for byte-3 value use bit-wise AND operator. It is shown below.

120.14.22.16	IP address
255.255.128.0	Mask
120.14.0.0	Sub-network address

In the above example, the bit wise ANDing is done in between 22 and 128. it is as follows

22	Binary representation	0 0 0 1 0 1 1 0
128	Binary representation	1 0 0 0 0 0 0 0
0		0 0 0 0 0 0 0 0

Thus the sub-network address for this is 120.14.0.0.

Example 4.15.5 *Find the class of the following address.*

a) 1.22.200.10 b) 241.240.200.2 c) 227.3.6.8 d) 180.170.0.2

Solution :

a) 1.22.200.10 Class A IP address

b) 241.240.200.2 Class E IP address

c) 227.3.6.8 Class D IP address

d) 180.170.0.2 Class B IP address

Example 4.15.6 *Find the netid and Hostid for the following.*

a) 19.34.21.5 b) 190.13.70.10 c) 246.3.4.10 d) 201.2.4.2

Solution :

a) netid ⇒ 19 Hostid ⇒ 13.70.10

b) netid ⇒ 190.13 Hostid ⇒ 70.10

c) No netid and No Hostid because 246.3.4.10 is the class E address.

d) netid ⇒ 201.2.4 Hostid ⇒ 2

Scanned with CamScanner

Example 4.15.7 *An organization is granted the block 211.17.180.0/24. The administrator wants to create 32 subnets.*

a) Find the subnet mask. *b) Find the number of addresses in each subnets.*

c) Find the first and last addresses in subnet 1.

d) Find the first and last addresses in subnet 32. **GTU : Dec.-10, Marks 7**

Solution : a) Find the subnet mask :

log 232 = 5 Extra 1s = **5** Possible subnets : **32** Mask : **/29** (24 + 5)

Subnet mask is 255.255.255.248

b) Find the number of addresses in each subnets :

2^{32-29} = 8 Addresses per subnet.

c) Find the first and last addresses in subnet 1 :

Subnet 1 : The first address is the begining address of the block or **211.17.180.0. First address in subnet 1 : 211. 17. 180 . 0**

Number of addresses : 0. 0. 0. 7

Last address in subnet 1 : 211. 17. 180. 7

d) Find the first and last addresses in subnet 32 :

Subnet 32 : To find the first address in subnet 32, we need to add 248 (31 × 8) in base 256 (0.0.0.248) to the first address in subnet 1. So that 211.17.180.0 + 0.0.0.248.

OR

211.17.180.248. Now we can calculate the last address in subnet 32.

First address in subnet 32 : 211. 17. 180. 248.

Number of addresses : 0.0.0.7.

Last address in subnet 32 : 211.17.180.255.

4.15.8 Network Address Translation (NAT)

- Within the company, every machine has a unique address of the form 10.X.Y.Z. when a packet leaves the company premises, it passes through the NAT box that convert the internal IP source address 10.0.0.1. NAT box is often combined in a single device with a firewall. It is also possible to integrate the NAT box into the company router.
- Whenever an outgoing packet enters the NAT box, the 10.X.Y.Z. SA is replaced by the company true IP address. In, addition, TCP source port field is replaced by an

Scanned with CamScanner

index into the NAT box 65536 entry translation table. This table entry contains the original IP address and original source port. Finally both the IP and TCP header checksums are recomputed and inserted into the packet.

- Fig. 4.15.10 shows the placement of NAT box.

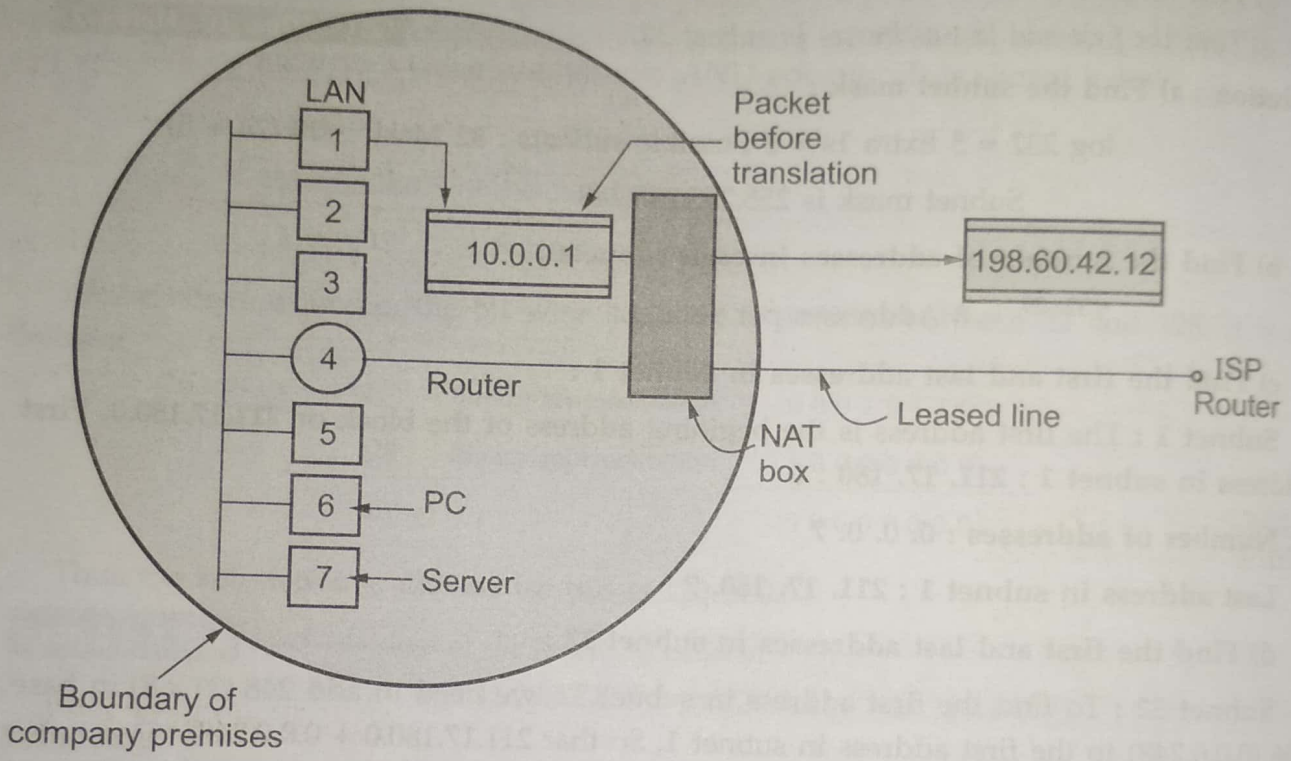


Fig. 4.15.10 NAT

- When process want to establish a TCP connection with a remote process, it attached itself to an unused TCP port on its own machine. This is called a source port and tells the TCP code where to send incoming packets belonging to this connection. The process also supplies a destination port to tell who to give the packet to on the remote side.

4.15.9 Classless InterDomain Routing (CIDR)

- Dividing the IP address space into A, B and C classes turned out to be inflexible. IP is rapidly becoming a victim of its own popularity, it is running out of addresses. In 1993 the classful address space restriction was lifted. An arbitrary prefix length to indicate the network number, known as Classless InterDomain Routing (CIDR), was adopted in place of the classful scheme.
- Using a CIDR notation, a prefix 205.100.0.0 of length 22 is written as 205.100.0.0/22. The corresponding prefix range runs from 205.100.0.0 through 205.100.3.0. The /22 notation indicates that the network mask is 22 bits, or

Scanned with CamScanner

255.255.252.0. CIDR routes packets according to the higher order bits of the IP address.

- The entries in a CIDR routing table contain a 32-bit IP address and a 32-bit mask. CIDR uses a technique called supernetting so that a single routing entry covers a block of classful addresses.
- For example, address of class C i.e. 205.100.0.0, 205.100.2.0, 205.100.2.0 and 205.100.3.0, CIDR allows a single entry 205.100.16.0/22. The use of variable length prefixed requires that the routing tables be searched to find the longest prefix match. For example, a routing table may contain entries for the above supernet 205.100.0.0/22 as well as for 205.100.0.0/20. This situation may arise when a large number of destinations have been aggregated into the block 205.100.0.0/20, but packets destined to 205.100.16.0/22 are to be routed differently. A packet with destination address 205.100.1.1 will match both of these entries, so the algorithm must select the match with the longest prefix.

University Questions

1. *Explain following with respect to IP address. Also give proper examples of each.*
   1) Structure of IPv4 address
   2) Subnet mask.
   3) Default gateway **GTU : Winter-14, Marks 7**
2. *What is subnet ? Explain various classes of IPv4 address with respect to bits reserved for host_id and net_id. Give example and ranges of each class of IPv4 address.* **GTU : Winter-14, Marks 7**
3. *Draw IP headed and explain the each filed of the header. Also explain the concept of fragmentation in detail.* **GTU : Dec.-10, Marks 7**
4. *An organization is granted the block 211.17.180.0/24. The administrator wants to create 32 subnets.*
   a) Find the subnet mask. *b) Find the number of addresses in each subnets.*
   c) Find the first and last addresses in subnet 1.
   d) Find the first and last addresses in subnet 32. **GTU : Dec.-10, Marks 7**
5. *Explain : MAC address and IP address.* **GTU : Dec.-11, Marks 5**
6. *Explain the IP addressing scheme in detail.* **GTU : Winter-12, Marks 7**
7. *What is IP address ? What is subnet ? Explain different IP address classes.* **GTU : Summer-14, Marks 7**
8. *Explain fragmentation and its use.* **GTU : Summer-14, Marks 4**

Scanned with CamScanner

4.16 IPv6

GTU : Dec.-11, Winter-12, Summer-13

- IPv4 provides the host to host communication between systems in the Internet. IPv4 has played a central role in the internetworking environment for many years. It has proved flexible enough to work on many different networking technologies.
- In the early 1990 the IETF began to work on the successor of IPv4 that would solve the address exhaustion problem and other scalability problems.

Advantages of IPv6

1. Larger address space
2. Better header format
3. Security capabilities
4. Support for resource allocation
5. New options
6. Allowance for extension

- IPv6 addresses are 128 bits in length. Addresses are assigned to individual interface on nodes, not to the node themselves. A single interface may have multiple unique unicast addresses. The first field of any IPv6 address is the variable length format prefix, which identifies various categories of addresses.

IPv6 addresses

- A new notation has been devised for writing 16-byte addresses. They are written as eight groups of four hexadecimal digits with colons beween the groups, like this
  8000 : 0000 : 0000 : 0000 : 0123 : 4567 : 89AB : CDEF

Optimization

1. Leading zeros within a group can be omitted so 0123 can be written as 123.
2. One or more groups of 16 zero bits can be repalced by a pair of colons. The address new becomes

   8000 : : 123 : 4567 : 89AB : CDEF

4.16.1 Address Types

- IPv6 allows three types of addresses.
  1. Unicast 2. Anycast 3. Multicast

1. Unicast : An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address.

2. Anycast : An identifier for a set of interfaces. A packet sent to an anycast address is delivered to one of the interfaces identified by the address.

3. Multicast : An identifier for a set of interfaces. A packet sent to a multicast address is delivered to all interfaces identified by that address.

Scanned with CamScanner

- Following Table 4.16.1 shows the current allocation of addresses based on the format prefix.
- The first field of any IPv6 address is the variable-length fromat prefix, which identifies various categories of addresses.

Allocation space	Prefix (binary)	Fraction of address space
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reversed for NSAP allocation	0000 001	1/128
Reversed for IPX allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Unassigned	001	1/8
Provider-based unicast address	010	1/8
Unassigned	011	1/8
Reserved for geographic-based unicast addresses	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link local use addresses	1111 1110 10	1/1024
Site local use addresses	1111 1110 11	1/1024
Multicast addresses	1111 1111	1/256

Table 4.16.1 Address allocation

4.16.2 Packet Format

- The IPv6 packet is shown in Fig. 4.16.1. Each packet is composed of a mandatory base header followed by the payload. The payload consists of two parts : Optional and data.

Scanned with CamScanner

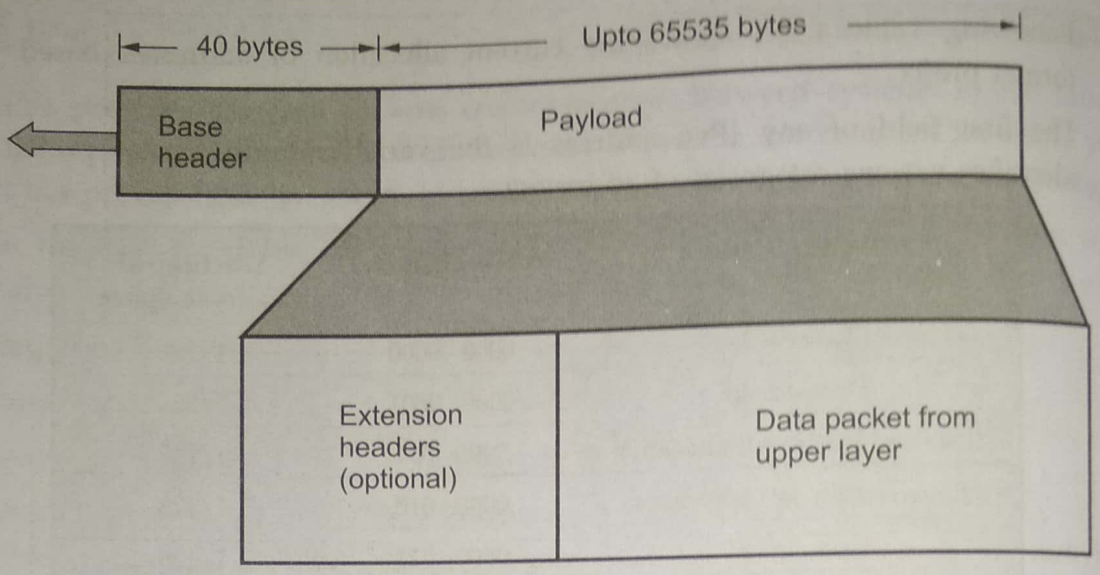


Fig. 4.16.1 IPv6 datagram header of payload

- Fig. 4.16.2 shows the IPv6 datagram header format. (See Fig. 4.16.2 on next page).

1. **Versions :** This 4 bits field defines the version number of the IP. The value is 6 for IPv6.
2. **Priority :** The 4 bits priority field defines the priority of the packet with respect to traffic congestion.
3. **Flow label :** It is 24 bits field that is designed to provide special handling for a particular flow of data.
4. **Payload length :** The 16 bits payload length field defines the length of the IP datagram excluding the base header.
5. **Next header :** It is an 8 bits field defining the header that follows the base header in the datagram.
6. **Hop limit :** This 8 bits hop limit field serves the same purpose as the TTL field in IPv4.
7. **Source address :** The source address field is a 128 bits internet address that identifies the original.
8. **Destination address :** It is 128 bits Internet address that usually identifies the final destination of the datagram.

- Next header codes for IPv6

Code	Next header
0	Hop by hop option
2	ICMP
6	TCP

Scanned with CamScanner

17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null
60	Destination option

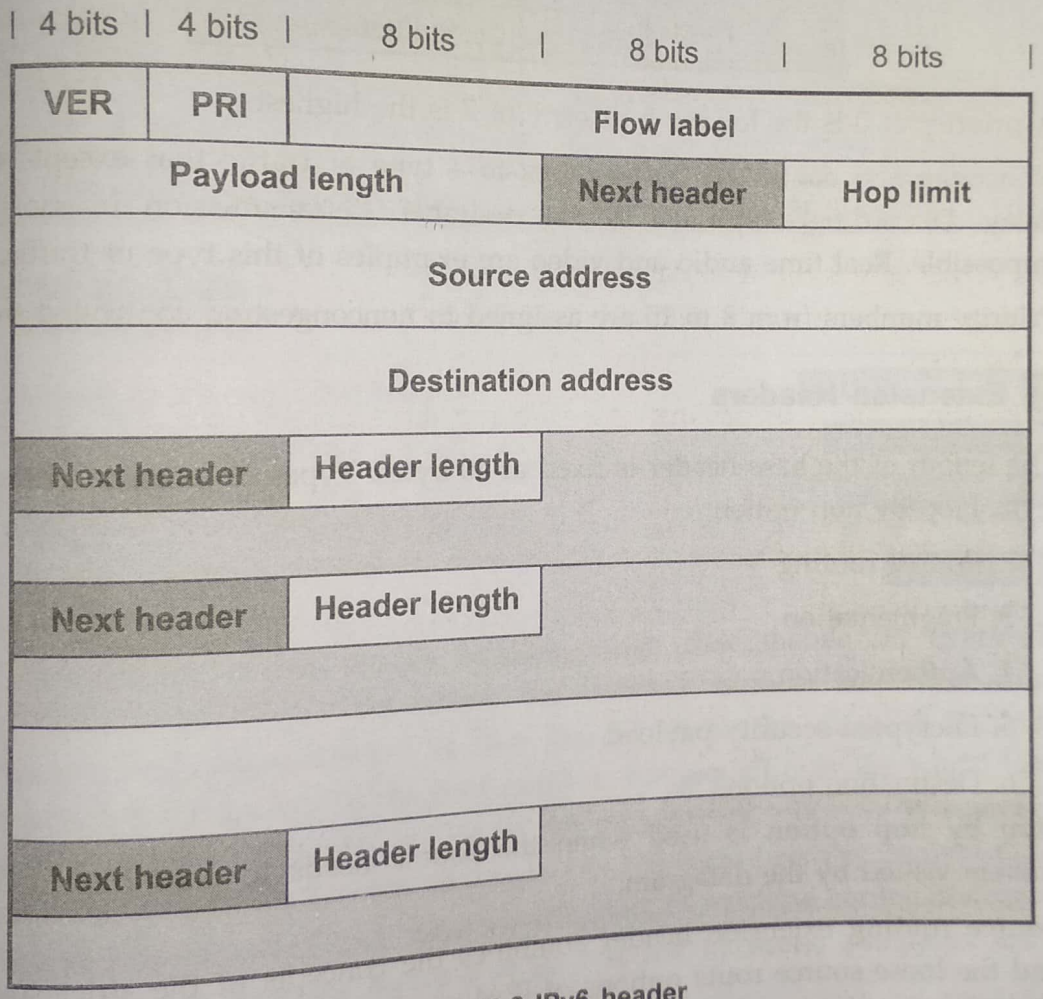


Fig. 4.16.2 IPv6 header

Priority

- The priority field defines the priority of each packet with respect to other packets from the same source. IPv6 divides traffic into two broad categories
  1. Congestion controlled 2. Noncongestion controlled
- If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion controlled traffic. congestion controlled data are assigned priorities from 0 to 7.

Scanned with CamScanner

Priority	Meaning
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

- A priority of 0 is the lowest; a priority of 7 is the highest.
- Noncongestion controlled traffic refers to a type of traffic that excepts minimum delay. Discarding of packets is not desirable. Retransmission in most cases is impossible. Real time audio and video are examples of this type of traffic.
- Priority numbers from 8 to 15 are assigned to noncongestion controlled traffic.

4.16.3 Extension Headers

- The length of the base header is fixed at 40 bytes. Types of extension headers are
  1. Hop by hop option
  2. Source routing
  3. Fragmentation
  4. Authentication
  5. Encrypted security payload
  6. Destination option
- **Hop by hop option** is used when the source needs to pass information to all routers visited by the datagram.
- **Source routing** extension header combines the concepts of the strict source route and the loose source route options of IPv4.
- The concept of **fragmentation** is the same as that in IPv4. In IPv6, only the original source can fragment.
- The **authentication** header has a dual purpose : It validates the message sender and ensures the integrity of data.
- The **encrypted security payload** is an extension that provides confidentiality and guards against eavesdropping.

Scanned with CamScanner

- The **destination option** is used when the source needs to pass information to the destination only. Intermediate routers are not permitted access to this information.

4.16.4 Comparison between IPv4 and IPv6

Sr. No.	IPv4	IPv6
1.	Header size is 32 bits.	Header size is 128 bits.
2.	It cannot support autoconfiguration.	Supports autoconfiguration
3.	Cannot support real time application.	Supports real time application.
4.	No security at network layer.	Provides security at network layer.
5.	Throughput and delay is more.	Throughput and delay is less.

University Questions

1. *Compare the IPv4 and IPv6 header.* **GTU : Winter-12, Marks 7**
2. *Compare : IPv4 and IPv6.* **GTU : Dec.-11, Marks 5**
3. *Explain in detail IPv6.* **GTU : Summer-13, Marks 7**

4.17 Mobile IP

GTU : Winter-12,13

- IP addressing system makes working easier than mobile IP. Every IP address contains three fields. The class, the network number and host number. For example, xxx.yyy.zzz.www in this xxx.yyy gives the class of IP and network number; the zzz.www is the host number. Routers all over the world have routing tables telling which line to use to get to network xxx.yyy. Whenever a packet comes in with a destination IP address of the form (xxx.yyy.zzz.www) it goes out on that line. If new IP address is configured to machine corresponding to its new location is unattractive because large number of users, programs and database would have to be informed of the change.
- Another approach is to have the routers use complete IP address for routing, instead of just the class and network. For this, each router requires millions of table entries.
- Following are some rules for using mobile IP.
  1. Each mobile host must be able to use its home IP address anywhere.
  2. Software changes to the fixed hosts were not permitted.
  3. Changes to the router software and tables were not permitted.

Scanned with CamScanner

4. No overhead should be incurred when a mobile host is at home.

- In mobile IP, every site that wants to allow its users to roam has to create a home agent and create foreign agent for visitors. When a packet arrives at the user's home local area network, it comes in at some router attached to the LAN. The router then tries to locate the host in the usual way, by broadcasting an ARP packet asking. The home agent responds to query by giving its own ethernet address. The router then sends packet for that address to the home agent. At the time the mobile host moves, the router probably has its ethernet address cached.
- To replace that ethernet address with the home agent's a trick called gratuitions ARP is used. Cryptographic authentication protocol are used for security.

University Questions

1. *Explain the mobile-IP.* GTU : Winter-12,13, Marks 7
2. *Write short note on : Mobile IP* GTU : Winter-13, Marks 4

4.18 Study of Router

- Routers are devices that forward data packets along networks. A router is connected to at least two networks, commonly two LANs or WANs or a LAN and its ISP's network.
- Routers are located at gateways, the places where two or more networks connect. Routers use headers and forwarding tables to determine the best path for forwarding the packets, and they use protocols such as ICMP to communicate with each other and configure the best route between any two hosts. Very little filtering of data is done through routers.
- Cisco routers have various components that are controlled by the Cisco Internetwork Operating System (IOS). Components are memory, interfaces, and ports. Each component provides some added functionality to a router.

1. Memory : A router contains different types of memory, where it can store images, configuration files, and microcode. Different types of memory are as follows :

a. Random Access Memory : Often referred to as dynamic random-access memory. RAM is the working area of memory storage used by the CPU to execute Cisco IOS software and to hold the running configuration file, routing tables, and ARP cache. The running configuration file contains the current configuration of the software. Information in RAM is cleared when the router is power-cycled or reloaded.

b. Read Only Memory : ROM is sometimes referred to as erasable programmable read-only memory. It is hard-wired read-only memory in the router. ROM contains

Scanned with CamScanner

power-on self-test diagnostics and the bootstrap or boot-loader software. This code allows the router to boot from ROM when it cannot find a valid Cisco IOS software image. This is known as ROM Monitor mode. This is a diagnostic mode that provides a user interface when the router cannot find a valid image.

c. Flash : Available as EPROM's, single in-line memory modules, or PCMCIA cards. Flash is the default location where a router finds and boots its IOS image. On some platforms, additional configuration files or boot images can be stored in Flash. The contents of Flash are retained when the router is power-cycled or reloaded.

d. Nonvolatile Random Access Memory : NVRAM stores the startup configuration file, which is used during system startup to configure the software. In addition, NVRAM contains the software configuration register, a configurable setting in Cisco IOS software that determines which image to use when booting the router. The contents of NVRAM are retained when the router is power-cycled or reloaded.

2. POST : Power On Self Test is stored in ROM microcode. It checks for basic functionality of router hardware and determines which interfaces are present.

3. Config-Register : It controls how router boots; value can be seen with "show version" command; is typically 0x2102, which tells the router to load the IOS from flash memory and the startup-config file from NVRAM.

- **Reasons why you would want to modify the config-register :**
  1. Force the router into ROM monitor mode.
  2. Select a boot source and default boot filename.
  3. Enable/Disable the break function.
  4. Control broadcast addresses.
  5. Set console terminal baud rate.
  6. Load operating software from ROM.
  7. Enable booting from a TFTP server.

Router Configurations :

- Router always has two configurations :
  1. Running configuration
     - In RAM, determines how the router is currently operating
     - Is modified using the configure command
     - To see it : show running-config
  2. Startup confguration
     - In NVRAM, determines how the router will operate after next reload

Scanned with CamScanner

- Is modified using the copy command
- To see it : show startup-config

Router Access Modes

1. User EXEC mode : limited examination of router.

```
Routername>
```

2. Privileged EXEC mode: detailed examination of router, debugging, testing, file manipulation.

```
Routername#
```

3. ROM monitor : Useful for password recovery and new IOS upload session.
4. Setup mode : Available when router has no startup-config file.

4.18.1 Router Interfaces and Ports

- Routers contain different types of interfaces and ports. Interfaces assist the router in routing packets and bridging frames between network segments, and they provide a connection point to different types of transmission media. Ports, on the other hand, provide management access to the router.
- Routers common interface types are as follows :
  1. Serial 2. Ethernet 3. Token ring 4. Asynchronous 5. FDDI
- Ports on the router enable a user to connect to the router for management and configuration purposes. You can connect either a terminal (DTE) or a modem (DCE) to these ports. Some of the common ports are : Console and auxiliary.
- The console and auxiliary ports are physical ports on the router that provide management access to the router. In addition to these, there are also Virtual Terminal (VTY) lines, which are software-defined lines that allow Telnet access to the router.
- The default VTY configuration is VTY lines 0 through 4, allowing five simultaneous Telnet sessions to the router. Passwords can be configured on each VTY line to secure access to the router.

External Configuration Sources

1. Console : Direct PC serial access
2. Auxilliary port : Modem access
3. Virtual terminals : Telnet access
4. TFTP Server : Copy configuration file into router RAM
5. Network Management Software : CiscoWorks

Scanned with CamScanner

4.18.2 Command Line Interface (CLI)

- CLI is more flexible than setup mode.
- If you choose to skip setup mode, you will be taken to the command line and the status of all the interfaces will be shown to the screen.
- CLI is the primary interface used to configure, manage, and troubleshoot Cisco devices. This user interface enables you to directly execute IOS commands, and it can be accessed through a console, modem, or Telnet connection. Access by any of these methods is generally referred to as an **EXEC session.**
- To use the CLI, just say "No" to entering the initial configuration dialog. Initial prompt consists of two parts :
  a. Hostname b. Greater than symbol (>).
- The first prompt will look like ***Routername>*** the greater than sign at the prompt tells you that you are in user mode. In user mode you can only view limited statistics of the router.
- To change configurations you first need to enter privileged EXEC mode. This is done by typing ***enable*** at the ***Routername>*** prompt, the prompt then changes to ***Routername#.***

```
Routername>
Routername>enable
Routername#
```

- This mode supports testing commands, debugging commands, and commands to manage the router configuration files.
- To go back to user mode type ***disable*** at the ***Routername#*** prompt.

```
Routername# disable
Routername>
```

- If you want to leave completely, type ***logout*** at the user mode prompt. You can also exit from the router while in privileged mode by typing ***exit*** or ***logout*** at the ***Routername#*** prompt.

```
Routername>logout
```

- To configure from a CLI, global changes are required to the router by typing configure terminal, which puts you in global configuration mode and changes what's known as the running-config. A global command is set only once and affects the entire router.

```
Routername#conf t
Routername (config) #
```


Scanned with CamScanner

CLI Prompts :

- To change the running-config-the current configuration running in DRAM, you use the configure terminal. To change the startup-config-the configuration stored in NVRAM-you use the configure memory command. If you want to change a router configuration stored on a TFTP host, you use the configure network command.

Interfaces :

To make changes to an interface, you use the interface command from global configuration mode :

```
Routername(config)#interface ?
  Async              Async interface
  BVI                Bridge-Group Virtual Interface
  Dialer             Dialer interface
  FastEthernet       FastEthernet IEEE 802.3
  Group-Async        Async Group interface
  Lex                Lex interface
  Loopback           Loopback interface
  Multilink          Multilink-group interface
  Null               Null interface
  Port-channel       Ethernet Channel of interfaces
  Tunnel             Tunnel interface
  Virtual-Template   Virtual Template interface
  Virtual-TokenRing  Virtual TokenRing
Routername(config)#interface fastethernet 0/0
Routername(config-if)#
```

- Subinterfaces allow you to create virtual interfaces within the router. The prompt then changes to *Routername(config-subif)#.*
- Listed the enhanced editing commands available on a Cisco router and options for Router command history are as follows :

Command	Meaning
Ctrl + A	Moves your cursor to the beginning of the line.
Ctrl + E	Moves your cursor to the end of the line.
Esc + B	Moves back one word.
Ctrl + B	Moves back one character.
Ctrl + F	Moves forward one character.
Esc + F	Moves forward one word.
Ctrl + D	Deletes a single character.
Backspace	Deletes a single character.
Ctrl + R	Redisplays a line.

Scanned with CamScanner

Ctrl + U	Erases a line.
Ctrl + W	Erases a word.
Ctrl + Z	Ends configuration mode and returns to EXEC.
Tab	Finishes typing a command for you.
?	Gives you the possible options for finishing the command.

Table 4.18.1 Enhanced editing commands

Command	Meaning
Ctrl + P or Up arrow	Shows last command entered
Ctrl + N or Down arrow	Shows previous commands entered
Show history	Shows last 10 commands entered by default
Show terminal	Configurations and history buffer size
Terminal history size	Changes buffer size (max 256)

Table 4.18.2 Router-command history

- To configure user mode passwords, use the line command. The prompt then becomes Routername (config-line)#.

```
Routername#config t
```

- To configure routing protocols like RIP and IGRP, use the prompt (configrouter)#.

```
Routername#config t
```

- You can use the Cisco advanced editing features to help you configure your router. By using a question mark (?) at any prompt, you can see the list of commands available from that prompt.

```
Routername#?
```

Exec commands :

access-enable	Create a temporary Access-List entry
access-profile	Apply user-profile to interface
access-template	Create a temporary Access-List entry
bfe	For manual emergency modes setting
clear	Reset functions
clock	Manage the system clock
configure	Enter configuration mode
connect	Open a terminal connection
copy	Copy configuration or image data

Scanned with CamScanner

debug	Debugging functions (see also 'undebug')
disable	Turn off privileged commands
disconnect	Disconnect an existing network
connection	
enable	Turn on privileged commands
erase	Erase flash or configuration memory
exit	Exit from the EXEC
help	Description of the interactive help
system	
lock	Lock the terminal
login	Log in as a particular user
logout	Exit from the EXEC
mrinfo	Request neighbour and version information from a multicast router

--More—

- To find commands that start with a certain letter, use the letter and the question mark (?) with no space between them.

```
Routername#c?
```

clear clock configure connect copy

- Use the command show history to see the last 10 commands entered on the router.

```
Routername #sh history
en
sh history
show terminal
sh cdp neig
sh ver
sh flash
sh int e0
sh history
sh int s0
sh int s1
```

- The command show version will provide basic configuration for the system hardware as well as the software version, the names and sources of configuration files, and the boot images.

```
Routername#sh version
```

❑❑❑

Scanned with CamScanner

5 The Link Layer and Local Area Networks

Syllabus

Introduction and link layer services, Error-detection and correction techniques, Multiple access protocols, Addressing, Ethernet, Switches.

Contents

5.1 Introduction and Link Layer Services ..... **Dec.-10, 11, Summer-13, 14, Winter-14,** ..... Marks 7

5.2 Error Correction and Detection Techniques ..... **Dec.-10, May-12,Winter-14,** Marks 7

5.3 HDLC ..... **Dec.-11, Summer-13,** ..... Marks 7

5.4 Point-to-Point Protocol

5.5 Multiple Access ..... **Dec.-10, June-11, May-12, Summer-13, 14, Winter-13,14,** ..... Marks 7

5.6 Random Access

5.7 Controlled Access ..... **May-12,** ..... Marks 3

5.8 IEEE Standard 802.3 ..... **Dec.-11,** ..... Marks 5

5.9 Bridged Ethernet ..... **May-12, Winter-13, Summer-14,** ..... Marks 7

5.10 Fast Ethernet

5.11 Gigabit Ethernet ..... **Dec.-10, 11, June-11, May-12, Summer-13,14, Winter-14,** Marks 7

5.12 Switching and Bridging

Scanned with CamScanner

5.1 Introduction and Link Layer Services

GTU : Dec.-10,11, Summer-13,14, Winter-14

- Some important functions of data link layer include well defined service interface to the network layer, framing, flow control, error detection and error control, frame formatting and sequencing. All these are very important functions for reliable communication and plays a vital role in designing data link layer.

5.1.1 Services Provided to the Network Layer

- The primary responsibility of data link layer is to provide services to the network layer. The principle service is transferring data from the network layer on the source machine to the network layer on the destination machine.
- The two data link layer communicates with each other by data link control protocol.
- Following are the important services provided by data link layer to the network layer.
  1) Unacknowledged connectionless service.
  2) Acknowledged connectionless service.
  3) Acknowledged connection-oriented service.

1) Unacknowledged connectionless service

- As the name suggests, it is unacknowledged form of transmission. Here the source machine sends the data to the destination machine without any acknowledgement. For this, no connection is either established or released. If the data is lost due to noise or interference, the lost data is not even recovered by the layer.

2) Acknowledged connectionless service

- In acknowledged connectionless service each data frame is acknowledged by the destination machine. If any data frame is lost or not arrived in time the same can be transmitted again. In this service no connection are used.

3) Acknowledged connection-oriented service

- Acknowledged connection service establishes a connection prior to data transmission. Each frame is numbered before transmission and corresponding acknowledgement is also received. The transmission is carried out in distinct phases.

5.1.2 Framing

- Framing in the data link layer separates a message from one source to a destination or from other messages to other destinations by adding a sender address and a destination address.

Scanned with CamScanner

- To service the network layer, data link layer uses the service provided to it by the physical layer.
- Physical layer accepts the raw bit stream and delivers it to the destination. This bit stream may contain error i.e. number of bits received may not be equal to number of bits transmitted.
- The data link layer breaks the stream into discrete frames and computes the checksum for each frame.
- At the destination the checksum is recomputed.
- The breaking of bit stream by inserting spaces or time gaps is called framing. Since it is difficult and risky to count on timing and mark the start and end of each frame.

Fixed-size framing

- Frames can be of fixed or variable size. In fixed size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter.
- ATM is the example of fixed size framing.

5.1.2.1 Variable Size Framing

- In variable size framing, end of the frame and the beginning of the next frame is defined.
- Two methods are used for this purposes.
  1. Character oriented 2. Bit oriented

5.1.2.2 Character Oriented Protocol

- In this type, data to be carried are 8-bit characters from a coding system such as ASCII. Header contains source and destination address and other control information are also multiple of 8 bits. Trailer contains error detection or error correction redundant bits are also multiples of 8 bits.
- Fig. 5.1.1 shows character oriented protocol frame.

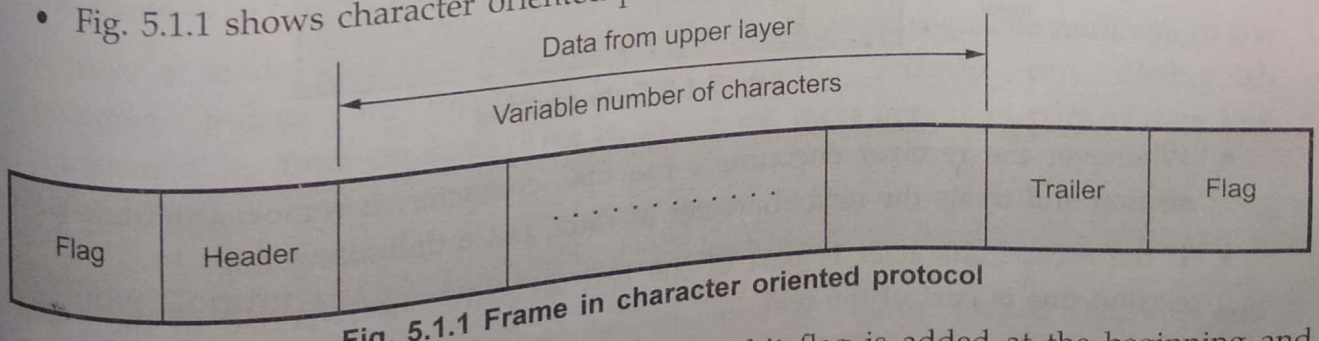


Fig. 5.1.1 Frame in character oriented protocol

- To separate one frame from the next, an 8-bit flag is added at the beginning and The flag consists of protocol dependent special characters,

Scanned with CamScanner

- Character oriented framing was suitable only for text data transmission. The flag could be selected to be any character not used for text communication.
- When we send other types of information such as graphs, audio and video, the flag could also be part of the information. So it creates problem for receiver. When receiver encounters this pattern in the middle of the data, thinks it has reached the end of the frame.
- To solve this problem, a **byte stuffing** was used.

Byte stuffing

- A special byte is added to the data section of the frame. When there is a character with the same pattern as flag. The data section is stuffed with an extra byte. This byte is usually called the **Escape Character (ESC)**; which has a predefined bit pattern.
- Fig. 5.1.2 shows byte stuffing.

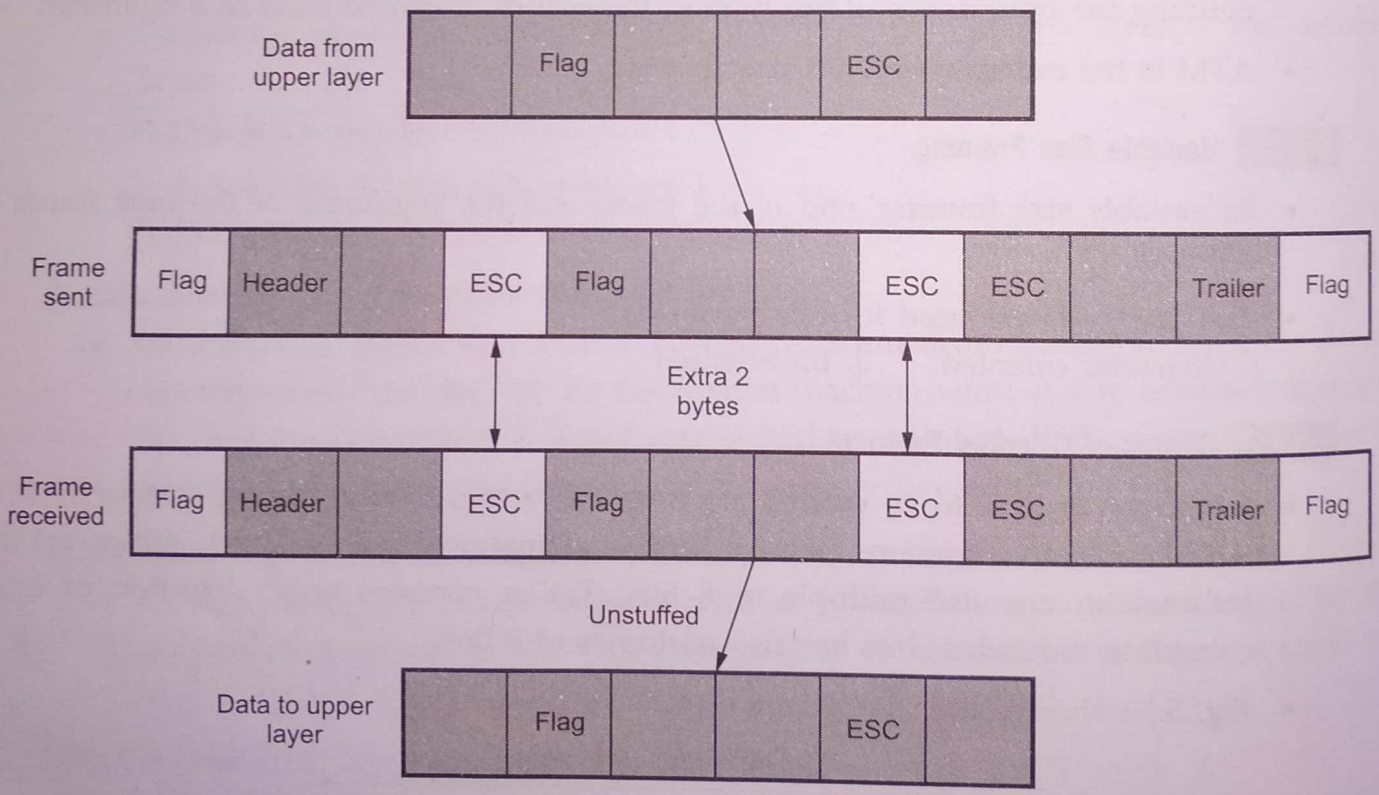


Fig. 5.1.2 Byte stuffing

- Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.
- If the escape character is part of the text an extra one is added to show that the second one is part of the text.

Scanned with CamScanner

5.1.2.3 Bit Oriented Protocols

- In this protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio and video. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame.
- In bit stuffing a specific bit is stuffed into the outgoing character stream. The format is as follows :

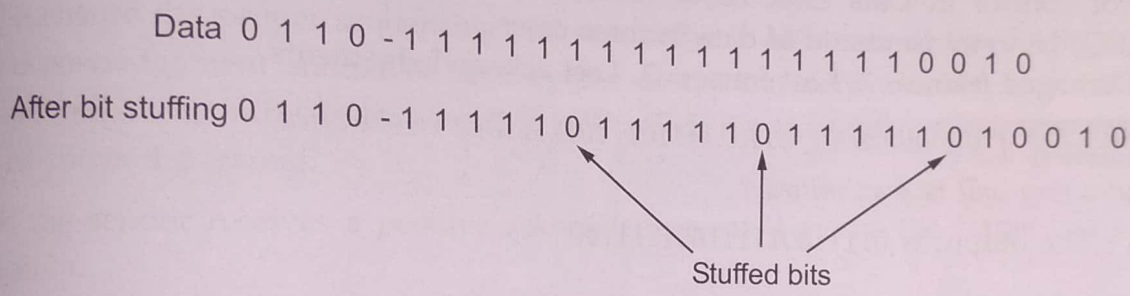


Fig. 5.1.3 Bit stuffing

- Each frame begins and ends with a special bit pattern, 01111110 called a flag byte. When five consecutive 1's are encountered in the data, it automatically stuffs a '0' bit into outgoing bit stream.

5.1.3 Error Control

- To ensure the proper sequencing and safe delivery of frames at the destination, an acknowledgement should be sent by the destination network. The receiver sends back special control frames bearing positive or negative acknowledgements about the incoming frames.
- If the sender receives a positive acknowledgement it means the frame has arrived safely.
- If a negative acknowledgement arrives that means, something has gone wrong and the frame is to be retransmitted.
- A timer at sender's and receiver's end is introduced. Also sequence numbers to the outgoing frames are maintained so that the receiver can distinguish retransmissions from originals. This is one of the most important part of data link layer duties.

5.1.4 Flow Control

- When the sender is running on fast machine or lightly loaded machine and receiver is on slow or heavily loaded machine. Then the transmitter will transmit [illegible] the receiver can accept them.

Scanned with CamScanner

- Even if the transmission is error free at a certain point the receiver will simply not be able to handle the frames as they arrive and will start to lose some.
- To prevent this, flow control mechanism is incorporated which includes a feedback mechanism requesting transmitter a retransmission of incorrect message block.
- The most common retransmission technique is known as Automatic-Repeat -Request.
- Error control in Data Link Layer (DLL) is based on Automatic Repeat Request (ARQ) i.e. retransmission of data in three cases.
  1. Damaged frames 2. Lost frames 3. Lost acknowledgements.

Example 5.1.1 *If the bit string 0110111110111011111010 is subjected to bit stuffing, what output string will be transmitted ?*

Solution : The output is 011110011111001111100

University Questions

1. *Enlist various framing techniques used in data link layer. Explain any one in detail with suitable example.* **GTU : Winter-14, Marks 7**
2. *What is framing ? Explain the various methods used for carrying out the framing in detail.* **GTU : Dec.-10, Marks 7**
3. *Design issues of data link layer.* **GTU : Dec.-11, Marks 5, Summer-14, Marks 7**
4. *Using example explain flow control.* **GTU : Summer-13, Marks 7**

5.2 Error Correction and Detection Techniques

GTU : Dec.-10, May-12, Winter-14

- Data transmission from one device to another device with complete accuracy is possible through network. An unavoidable noise and interference is added to the communication channel. Error is reduced using the digital transmission system but complete control of error is not possible. Error bit rate for copper wires is 10^{-6}. Modem optical fiber cable have the error bit rate 10^{-9}, which is less than copper wires.
- It is more likely that some part of a message will be changed in transmission. Many factors like noise, electromagnetic interference can alter the given data unit. In this topic we discuss parity check codes, the Internet checksum and polynomial codes that are used in error detection. Data link layer or transport layer of the OSI model support the error detection and error correction method.
- **Reasons for error :**
  1) If the power supply in the system is not exactly at the specified voltage component may not operate perfectly.

Scanned with CamScanner

2) System may be operating at its low or high temperature limit.
3) Crosstalk from adjacent signals can corrupt the signal.
4) Because of resistance, inductance and capacitance, the data signal in the wire becomes weak.
5) Voltage level is not proper after crash the system or trip through the wire.
6) Errors occur in modern system by the laws of probability, specially with computer memory circuit.

- To ensure the proper sequencing and safe delivery of frames at the destination, an acknowledgement should be sent by the destination network. The receiver sends back special control frames bearing positive or negative acknowledgements about the incoming frames.
- If the sender receives a positive acknowledgement it means the frame has arrived safely.
- If a negative acknowledgement arrives that means, something has gone wrong and the frame is to be retransmitted.
- A timer at sender's and receiver's end is introduced. Also sequence numbers to the outgoing frames are maintained so that the receiver can distinguish retransmissions from originals. This is one of the most important part of data link layer duties.

5.2.1 Types of Errors

- Two general types of errors can occur
  1) Single bit error 2) Burst error

1) Single bit error

- It means that only 1 bit of a given data unit is changed from 1 to 0 or from 0 to 1. A single bit error is an isolated error condition that alters one bit but does not affect nearby bits.
- A single bit error can occur in the presence of white noise, when a slight random deterioration of the signal to noise ratio is sufficient to confuse the receiver's decision of a single bit. Single bit errors are the least likely type of error in serial data transmission.

2) Burst error

- The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.

Scanned with CamScanner

- Burst errors are more common and more difficult to deal with errors. Burst errors can be caused by impulse noise. Note that the effects of burst errors are greater at higher data rates.

5.2.1.1 Error Detection

- The simplest form of error detection is to append a single bit, called a **parity check**, to a string of data bits. This parity check bit has the value 1 if the number of 1's in the bit string is odd and has the value 0 (zero) otherwise. In other words, the parity check bit is the sum, modulo 2, of the bit values in the original bit string. In the ASCII character code, characters are mapped into strings of seven bits and then a parity check is appended as an eighth bit as shown in Fig. 5.2.1.

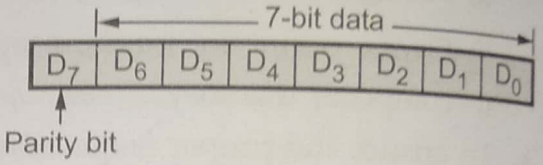


Fig. 5.2.1

- Note that the total number of 1's in an encoded string is always even. If an encoded string is transmitted and a single error occurs in transmission, then whether a 1 is changed to a 0 (zero) or a 0 to a 1 the resulting number of 1's in the string is odd and the error can be detected at the receiver. But receiver cannot tell which bit is in error, nor how many errors occurred. It simply knows that errors occurred because of the odd number of 1's.
- It is rather remarkable that for bit strings of any length, a single parity check enables the detection of any single error in the encoded string. Two errors in an encoded string always leave the number of 1's even so that the error cannot be detected. In general, any odd number of errors are detected and any even number are undetected.
- It is observed that, error detection requires redundancy in that the amount of information that is transmitted is over and above the required minimum. For a single parity check code of lengh k + 1, k bits are information bits and one bit is the parity bit. Therefore, the fraction $1/(k+1)$ of the transmitted bits is redundant.
- Second observation is that every error detection technique will fail to detect some errors. The effectiveness of an error detection code is measured by the probability that the system fails to detect an error.

5.2.1.2 Redundancy

- Redundancy is a form of error detection where each data unit is sent multiple times, i.e. twice. At the receiver side, the two units are compared and if they are same, it is assumed that no transmission errors have occurred. Redundancy is a character redundancy and message redundancy.

Scanned with CamScanner

- When the data unit is a single character, it is called **character redundancy,** whereas if the data unit is the entire message, it is called **message redundancy.**
- Another type of redundancy used with short messages is to transmit the same message several times. At the receiver side, if a given number of the messages are the same, it is assumed to be a successful transmission.

5.2.1.3 Detection versus Correction

- The ability to detect when a transmission has been changed is called **error detection**. When an error is detected it may actually be fixed without a second transmission. This is called **error correction.**
- The correction of errors is more difficult than the detection.
- The error detection, users are looking only to see if any error has occurred. A single bit error is the same as a burst error.
- In error correction, user need to know the exact number of bits that are corrupted and their location in the message.

5.2.1.4 Forward Error Correction versus Retransmission

- In **Forward Error Correction** (FEC), the transmitted bits contain the actual data along with checking bits which allow both the detection and correction of many errors. In an FEC system, the transmitted data are encoded so that the receiver can correct as well as detect errors. FEC techniques are used to correct errors on simplex channels.
- FEC is preferred on system with large transmission delays. There is no requirement for retransmitting the messages as long as the errors are infrequent. A burst type of interference destroying several bits cannot be corrected by this method. Whenever highest level of data integrity and confidence is needed, FEC is considered.
- Correction by **restransmission** is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.

5.2.1.5 Coding

- Redundancy is achieved through various coding schemes. Coding schemes are divided into two class.
  a) Block coding b) Conventional coding

Scanned with CamScanner

- Fig. 5.2.2 shows the general idea of the coding.

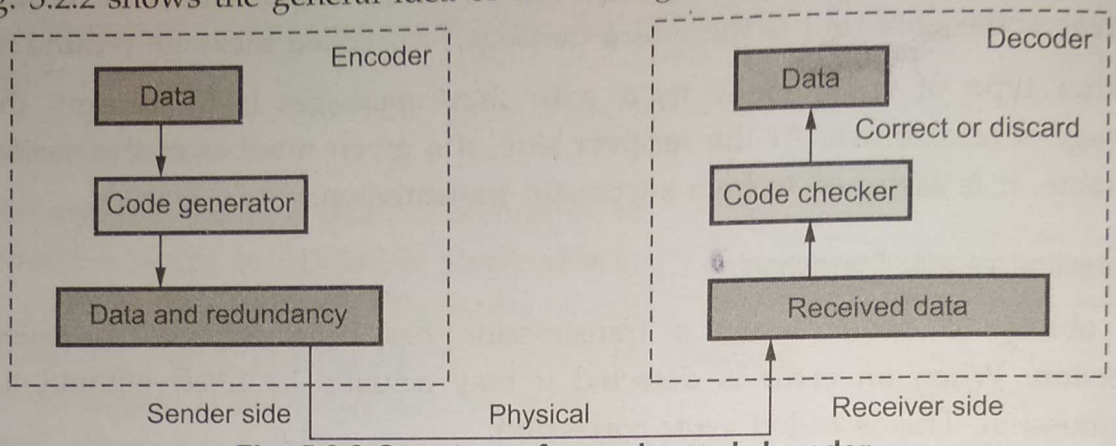


Fig. 5.2.2 Structure of encoder and decoder

5.2.1.6 Modular Arithmetic

- In modulo-N arithmetic, we use only the integers in the range 0 to N–1. For example, if the modulus is 12, we use only the integers 0 to 11.
- Addition and subtraction of modulo-2 is shown here.
- There is no carry when you add/subtract two digits in a column.

Addition

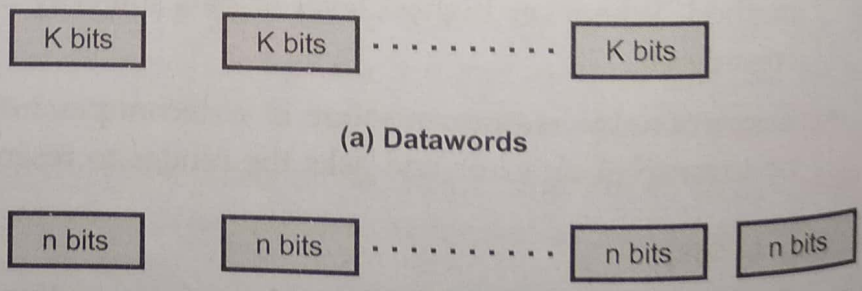
$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +\,1 \\ \hline 0 \end{array}$$

Subtraction

$$\begin{array}{r} 0 \\ -\,0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ -\,1 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ -\,0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ -\,1 \\ \hline 0 \end{array}$$

5.2.2 Block Coding

- In block coding, message is divided into blocks. Each block size is K bits and called as **datawords**. Redundant bits (r) is add to each block to make the length n = K + r. The resulting n-bit blocks are called **codewords**.
- Fig. 5.2.3 shows the datawords and codewords in block coding.
- With K bits, combination of 2^K datawords are possible and with n bits, 2^n codewords combination are possible. The block coding process is one-to-one; the same dataword is always encoded as the same codeword.



(a) Datawords

(b) Codewords

Fig. 5.2.3 Datawords and codewords

5.2.2.1 Error Detection

Scanned with CamScanner

1) The receiver has a list of valid codewords.

2) The original codeword has changed to an invalid one.

Fig. 5.2.4 shows the role of block coding in error detection.

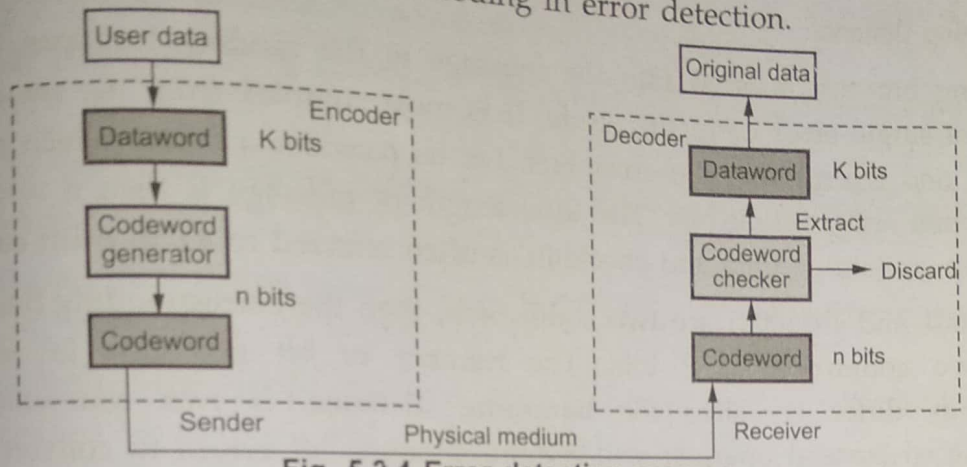
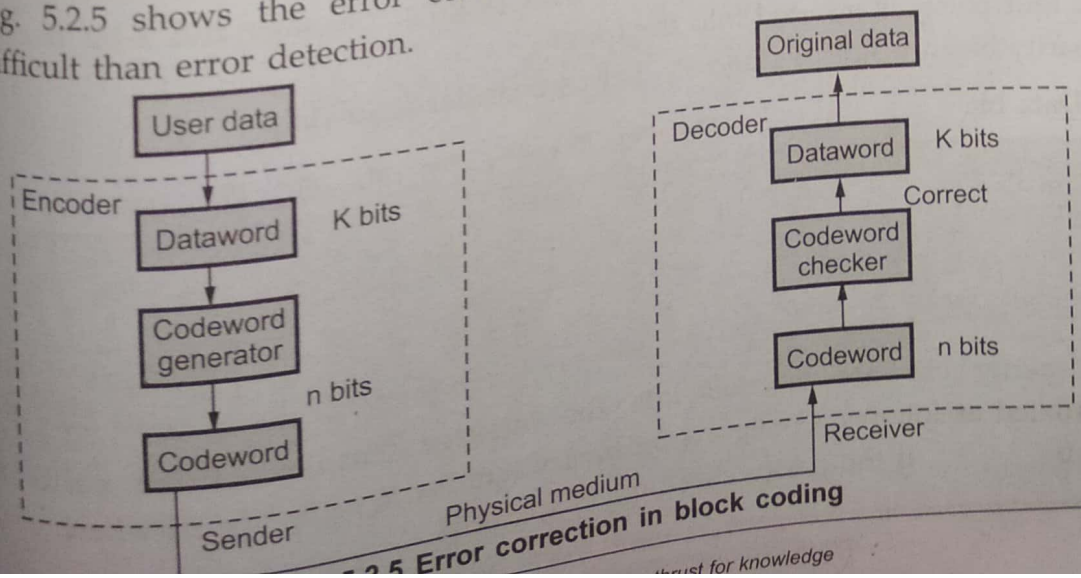


Fig. 5.2.4 Error detection process

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

5.2.2 Error Correction

- Fig. 5.2.5 shows the error correction process. Error correction is much more difficult than error detection.



Error correction in block coding

Scanned with CamScanner

- In error correction, the receiver needs to find the original codeword sent. More number of redundant bits are required for error correction than for error detection.

5.2.2.3 Hamming Distance

- Hamming bits are inserted into the message at the random locations. Hamming code is a single error correcting code. It is most complex from the stand point of creating and interpreting the error bits. Let us consider a frame which consists of m data bits and r check bits. The total length of message is then n = m + r. An n-bit unit containing data and checkbits is often referred to as an **n-bit codeword.**
- If 10001001 and 10110001 are two codewords, then the corresponding bits differ in these two codewords is 3 bits. The number of bit positions in which two codewords differ is called the **hamming distance.** If two codewords are a hamming distance d apart, it will require d single bit errors to convert one into the other. Determining the placement and binary value of the hamming bits can be implemented using hardware, but it is often more practical to implement them using software. The number of bits in the message are counted and used to determine the number of hamming bits to be used. The equation is used to count the number of hamming bits.

$$2^H \geq M + H + 1 \quad \ldots (5.2.1)$$

where M = Number of bits in a message

H = Hamming bits

- After calculating the number of hamming bits, the actual placement of the bits into the message is performed. Hamming code works as follows : Supose that frame consists of eight bits say $m_1\, m_2\, m_3\, m_4\, m_5\, m_6\, m_7\, m_8$. If n parity checks are used, there are 2^n possible combinations of failures and successes. If we use 4-bit parity checks, then there are 16 possible combinations of parity successes and failures. Total 12 bits are sent which contain 8-bit original message and 4-bit parity bits. The four parity is inserted into the frame.

Four parity bits are $P_1\, P_2\, P_3$ and P_4. Let us consider following.

Data bit $m_1\ m_2\ m_3\ m_4\ m_5\ m_6\ m_7\ m_8$

Hamming code	P_1	P_2	m_1	P_3	m_2	m_3	m_4	P_4	m_5	m_6	m_7	m_8
	1	2	3	4	5	6	7	8	9	10	11	12

- The parity bits are inserted into the message. Position of the parity bit is calculated as follows. Create a 4 bit binary number $b_4\, b_3\, b_2$ and b_1 where

$b_i = 0$ if the parity check for P_i succeeds

Scanned with CamScanner

$b_i = 1$ otherwise

for i = 1, 2, 3, or 4.

1) The parity bit P_1 is inserted at bit position 1 for even parity for bit positions 1, 3, 5, 7, 9, 10. In these bit positions it contains even number of 0s or 1s.
2) The parity bit P_2 is inserted at bit position 2, for even parity for bit positions 2, 3, 6, 7, 10, 11.
3) The parity bit P_3 is inserted at bit position 4, for even parity of the bit positions 4, 5, 6, 7, 12.
4) The parity bit P_4 is inserted at bit position 8 for even parity of the bit positions 8, 9, 10, 11, 12.

- For inserting the parity bit even or odd parity can be used. Each parity bit is determined by the data bits it checks. When a receiver gets a transmitted frame, it performs each of the parity checks. The combination of failures and successes then determines whether there was no error or in which position an error occurred. Once the receiver knows where the error occurred, it changes the bit value in that position and the error is corrected.

Minimum hamming distance (d_{min})

- The minimum hamming distance is the smallest hamming distance between all possible pairs in a set of words.
- To find the value of d_{min}, we find the hamming distances between all words and select the smallest one.

Example 5.2.1 *Find the minimum hamming distance of the coding scheme given below.*

Dataword	*Codeword*
00	*000*
01	*011*
10	*101*
11	*110*

Solution : Hamming distances of given codeword is

d(000, 011) = 2, d(000, 101) = 2, d(000, 110) = 2

d(011, 101) = 2, d(011, 110) = 2, d(101, 110) = 2

The d_{min} = 2

Example 5.2.2 *What is the hamming distance for each of the following codewords :*

a. d (10000, 00000) b. d (10101, 10000) c. d (11111, 11111) d. d (000, 000)

Solution : **a.** d(10000, 00000) = 1 **b.** d(10101, 10000) = 2

d. d(000, 000) = 0

Scanned with CamScanner

Example 5.2.3 *Using the code in Table, what is the dataword if one of the following codewords is received ?*

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

a. 01011 b. 11111 c. 00000 d. 11011

Solution : a. 01 **b.** Error **c.** 00 **d.** Error

Example 5.2.4 *We need a dataword of at least 11 bits. Find the values of k and n in the hamming code C(n, k) with* $d_{min} = 3$.

Solution : We need to find k = 2m – 1 – m; Y 11. We use trial and error to find the right answer :

a. Let m = 1 k = 2 m – 1 – m = 21 – 1 – 1 = 0 (not acceptable)

b. Let m = 2 k = 2m – 1 – m = 22 – 1 – 2 = 1(not acceptable)

c. Let m = 3 k = 2m – 1 – m = 23 – 1 – 3 = 4 (not acceptable)

d. Let m = 4 k = 2m – 1 – m = 24 – 1 – 4 = 11 (acceptable)

Comment : The code is **C(15, 11)** with $\mathbf{d_{min} = 3}$.

Example 5.2.5 *Assuming even parity, find the parity bit for each of the following data units.*

a. 1001011 b. 0001100 c. 1000000 d. 1110111

Solution : We need to add all bits modulo-2 (X-ORing). However, it is simpler to count the number of 1s and make them even by adding a 0 or a 1. We have shown the parity bit in the codeword in color and separate for emphasis.

Dataword	Number of 1s	Parity	Codeword
1001011	4	0	**0** 1001011
0001100	2	0	**0** 0001100
1000000	1	1	**1** 1000000
1110111	6	0	**0** 1110111

5.2.3 Linear Block Coding

- In a linear block code, the exclusive OR(XOR) of any two valid codewords creates another valid codeword. Almost all block codes used today belong to a subset called linear block codes.

Scanned with CamScanner

5.2.3.1 Minimum Distance for Linear Block Codes

- The minimum hamming distance is the number of 1s in the nonzero valid codeword with the smallest number of 1s.
- For example,

Datawords	Codewords
00	000
01	011
10	101
11	110

The number of 1s in the above nonzero codewords are 2, 2 and 2. (i.e. 011, 101, 110). So the minimum hamming distance is $d_{min} = 2$

- Let us consider the dataword and codeword

Dataword	Codeword
00	00000
01	01011
10	10101
11	11110

- In the above table, the numbers of 1s in the nonzero codewords are (01011) = 3, (10101) = 3 and (11110) = 4. So for this code we have $\mathbf{d_{min} = 3}$

5.2.4 Cyclic Redundancy Check

- Parity method detects only odd numbers of errors. To overcome this weakness polynomial codes error detection method is used. Polynomial codes involve generating check bits in the form of a Cyclic Redundancy Code (CRC). Therefore **polynomial also called Cyclic Redundancy Codes (CRCs).**
- The theory of polynomial code is derived from a branch of mathematics called algebra theory. The theory of CRC checksums is developed by using algebra and polynomials. These polynomials are equations which have the form of powers of X:

$$X^N + X^{N-1} + \ldots + X^2 + X^1 + X^0$$

- Polynomial codes are used with frame transmission schemes. A single set of check ted for each frame transmitted, based on the contents of the frame

Scanned with CamScanner

and is appended by the transmitter to the tail of the frame. The receiver then performs a similar computation on a complete frame and check digits. If no errors have been induced, a known result should always be obtained, if a different answer is found, this indicates an error. Consider an example for binary, the polynomial for binary 10011001 is

$$X^7 + X^4 + X^3 + X^0 \left(X^0 = 1\right)$$

- The polynomial which represents the data bits is called the message polynomial, usually shown as G(X). There is a second polynomial, called the generator polynomial P(X). G(X) and P(X) both having same format. Combine two polynomials P(X) and G(X) to produce the CRC checksum polynomial F(X) calculating CRC error as follows :
  a) Multiply the G(X) by X^{n-k}, where $n-k$ is the number of bits in the CRC checksum.
  b) Divide the resulting product $X^{n-k}\,[G(X)]$ by the generator polynomial P(X).
  c) Add the remainder C(X) to the product to give the F(X), which is representated as $X^{n-k}\,[G(X)]+C(X)$.
  d) The division is performed in binary without carrying or borrowing. In this case, the remainder is always 1 bit less than the divisor. The remainder is the CRC and the divisor is the generator polynomial.

Working of CRC

- Let's now describe how CRC works. Suppose we want to send the bit string **1101011** and the generator polynominal is $\mathbf{G(x) = x^4 + x^3 + 1}$

Step 1 : Append 0s to the end of the string. The number of 0s is the same the degree of the generator polynominal G(x) (in this case, 4). Thus the string becomes 1101011000.

Step 2 : Divide B(x) by G(x). We can write this algebraically as

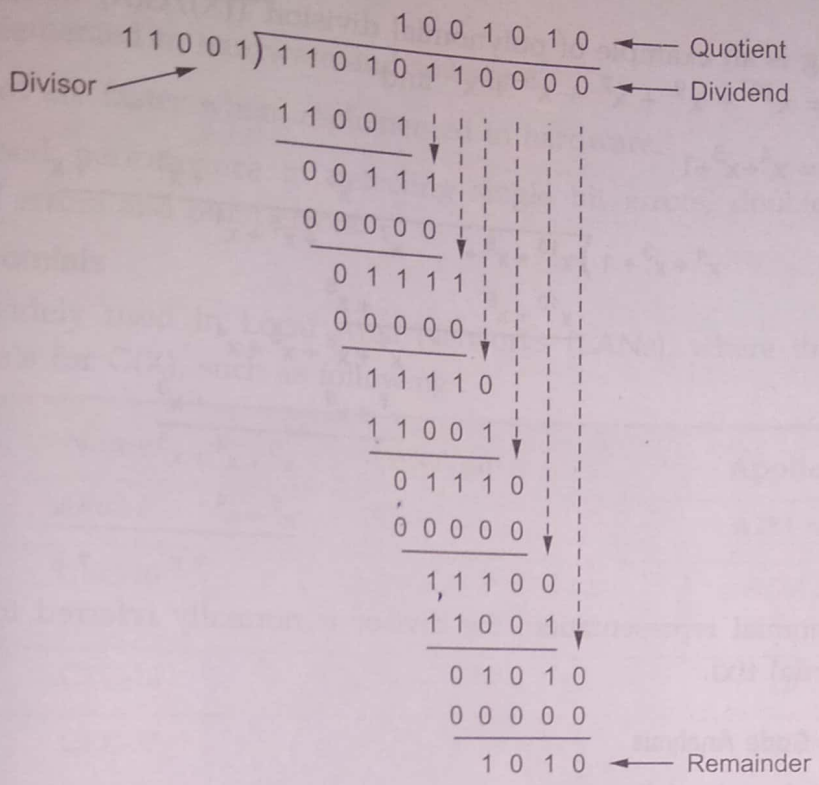
$$\frac{B(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

where Q(x) represent the quotient.

$G(x) = x^4 + x^3 + 1 = 11001$

String = 1101011 = After appending 11010110000

Scanned with CamScanner



Step 3 : Define $T(x) = B(x) - R(x)$. In this case,

Note that the string T is actually the same as string B with the appended 0s replaced by R. The sender transmit the string T.

5.2.4.1 Polynomials

- A pattern of 0s and 1s can be represented as a polynomial with coefficients of 0 and 1. The power of each term shows the positions of the bit, the coefficient shows the value of the bit.
- In general it interprets the bit string

$b_{n-1}\, b_{n-2}\, b_{n-3}$ ..... $b_2 b_1 b_0$

as the polynomial

$b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + b_{n-3}x^{n-3} + \ldots + b_2x^2 + b_1x + b_0$

- For example, the bit string 10010101110 is interpreted as

$x^{10} + x^7 + x^5 + x^3 + x^2 + x^1$

Since each b_i is either 0 or 1, we just write x^i when b_i is 1 and do not write any term when b_i is 0.

5.2.4.2 Degree of Polynomial

- The degree of polynomial is the highest power in the polynomial. For example, the degree of polynomial $x^5 + x + 1$ is 5.

Scanned with CamScanner

- Following is an example of polynomial division T(X)/G(X) where
  $T(X) = x^{10} + x^9 + x^7 + x^5 + x^4$ and

  $G(X) = x^4+x^3+1$

$$
\begin{array}{r|l}
x^4 + x^3 + 1 & x^6 + x^3 + x \\
\hline
x^{10} + x^9 + x^7 + x^5 + x^4 & \\
x^{10} + x^9 + x^6 & \\
\hline
x^7 + x^6 + x^5 + x^4 & \\
x^7 + x^6 + x^3 & \\
\hline
x^5 + x^4 + x^3 & \\
x^5 + x^4 + x & \\
\hline
x^3 + x &
\end{array}
$$

- In polynomial representation, the divisor is normally referred to as the **generator polynomial t(x).**

5.2.4.3 Cyclic Code Analysis

- Let us define the followings

  f(x) = Polynomial with binary coefficients

  d(x) = Dataword

  c(x) = Codeword

  g(x) = Generator

  e(x) = Error

  s(x) = Syndrome

- If s(x) is not zero, then one or more bits is corrupted. However, if s(x) is zero, either no bit is corrupted or the decoder failed to detect any errors.
- Let us first find the relationship among the sent codeword, error, received codeword and the generator we can say :

  Received codeword = c(x) + e(x)

- The received codeword is the sum of the sent codeword an error. The receiver divides the received codeword by g(x) to get the syndrome. We can write as this as

$$\frac{\text{Received codeword}}{g(x)} = \frac{c(x)}{g(x)} + \frac{e(x)}{g(x)}$$

- A single bit error is $e(x) = x^i$, where i is the position of the bit. If the single bit is caught, then x^i is not divisible by g(x). If g(x) has at least two terms and the coefficient of x^0 is not zero, then e(x) cannot be divided by g(x).

Scanned with CamScanner

5.2.4.4 Advantages of Cyclic Codes

1) Easily implemented in hardware and software.
2) Cyclic codes are faster when implemented in hardware.
3) It give good performance in detecting single bit errors, double errors, an odd number of errors and burst errors.

Standard polynomials

- CRC is widely used in Local Area Networks (LANs), where there are standard polynomials for G(X), such as following :

Sr. No.	Name	Polynomial	Application
1.	CRC-8	$x^8 + x^2 + x + 1$	ATM header
2.	CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^2 + 1$	ATM AAL
3.	CRC-16	$x^{16} + x^{12} + x^5 + 1$	HDLC
4.	CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	LANs

Example 5.2.6 *Generate the CRC code for message 1101010101. Given generator polynomial*

$$g(x) = x^4 + x^2 + 1$$

Solution : For polynomial division $T(X)/G(X)$

where $T(X) = 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1$
$(x^9 + x^8 + x^6 + x^4 + x^2 + 1)$

$G(X) = x^4 + x^2 + 1 = 1\ 0\ 1\ 0\ 1$

Polynomial division is done from an algebra.
Rules for addition and subtraction.

1. Addition

$0 + 0 = 0$
$1 + 0 = 1$
$0 + 1 = 1$
$1 + 1 = 0$

2. Subtraction

$0 - 0 = 0$
$1 - 0 = 1$
$0 - 1 = 1$
$1 - 1 = 0$

The steps are as follows :

Step 1 : Append 0 to the end of the string T(X).
The degree of polynomial $G(X) = x^4 + x^2 + 1 = 4$.
So we append 4 zeros to string $T(X)$.

Scanned with CamScanner

The string becomes

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

Step 2 : Divide $B(X)$ by $G(X)$. After appending 0s to $T(X)$ it becomes $B(X)$. (Actually it is new $T(X)$ divided by $G(X)$).

```
                1110001110
          ┌──────────────
   10101 ) 11010101010000
           11010
           10101
           ─────
           011111
            10101
            ─────
            010100
             10101
             ─────
             000011010
                 10101
                 ─────
                 011110
                  10101
                  ─────
                  010110
                   10101
                   ─────
                   000110      ← Remainder

  11010101010000
+          0110
  ──────────────
  11010101010110      ⇐ Codeword
```

Example 5.2.7 *Information to be transmitted is 110011 and the generator polynomial is represented as $g(x) = 11001$. Do a CRC check.*

Solution : Append by 4 bit 0 because coefficient of g(x) is 4.

The binary equivalent of d(x) = 1 1 0 0 1 1 0 0 0 0

```
                100001
          ┌──────────
   11001 ) 1100110000
           11001
           11001
           ─────
           0000010000
               11001
               ─────
               01001      ← Remainder
```


Scanned with CamScanner

Remainder is added to d(x) to give f(x) i.e.

1 1 0 0 1 1 0 0 0 0 + 0 1 0 0 1 = 1 1 0 0 1 1 1 0 0 1 ← f(x)

f(x) is transmitted.

Example 5.2.8 *The message 11001001 is to be transmitted, using CRC error detection algorithm. Assuming the CRC polynomial to be $x^3 + 1$, determine the message that should be transmitted. If the second left most bit is corrupted, show that it is detected by the receiver*

Solution : We take the message 11001001, append 000 to it and divide by 1001. The remainder is 011; what we transmit is the original message with the remainder appended, or 1100 1001 011.

```
          11010011
        ------------
1001 ) 11001001000
         1100
         1001
        ------
          1011
          1001
        ------
           1000
           1001
          ------
            1100
            1001
           ------
             1010
             1001
            ------
              011
```

Message transmit = 11001001000 + 011 = 11001001011

Example 5.2.9 *A bit stream 1101001011 is transmitted using standard CRC method. The generator polynomial is $x^4 + x + 1$. Show the actual bit string transmitted. Also explain the error detecting and correcting code with example.* **GTU : Dec.-10, Marks 7**

Solution : Bit stream = 1101001011

Generator polynomial = 10011

Actual bit string transmitted.

Scanned with CamScanner

```
      00001100011001
10011|11010010110000
       11010
       10011
       -----
       10010
       10011
       -----
        11011
        10011
        -----
        10000
        10011
        -----
        11000
        10011
        -----
         1011
```

Example 5.2.10 *Frame = 1 1 0 1 0 1 1 1 1 1, Generator = 1 0 0 1 1. Calculate CRC.*

GTU : Winter-14, Marks 7

Solution :

```
                    1100001110
              ----------------
10011 ) 11010111110000
        11010
        10011
        -----
        010011
         10011
         -----
         011110
          10011
          -----
          11010
          10011
          -----
          10010
          10011
          -----
             10
```

```
1 1 0 1 0 1 1 1 1 1 0 0 0 0
+                       1 0
-----------------------------
1 1 0 1 0 1 1 1 1 1 0 0 1 0   <= CRC
```

University Questions

1. *What is the difference between error detection and correction ? Explain any one error correction technique wth suitable example.* **GTU : Winter-14, Marks 7**
2. *What is cyclic redundancy check ? Show the calculation polynomial code checksum for a frame 1101011011 using the generator $x^4 + x + 1$.* **GTU : May-12, Marks 3**

Scanned with CamScanner

5.3 HDLC : High-level Data Link Protocol control (HDLC)

GTU : Dec.-11, Summer-13

- HDLC is the most important data link control protocol, also it is the basis for many other important data link control protocols, which use the same or similar formats and the same mechanisms as employed in HDLC.
- The HDLC protocol is an international standard that has been defined by ISO for use on both point-to-point and multipoint data links.
- It supports full duplex, transparent mode operation and is now extensively used in both multipoint and computer networks.
- Although the acronym HDLC is now widely accepted, a number of large manufacturers and other standards bodies still use their own acronyms. These include IBM's SDLC (Synchronous Data Link Control) and ADCCP (Advanced Data Communications Control Procedure), which is used by the American National Standards Institute (ANSI).
- To satisfy a variety of applications, HDLC defines three types of stations. These are, Station Type

1) **Primary station :** Primary station has the responsibility for controlling the operation of the link. Frames issued by the primary are called **command**.
2) **Secondary station :** Secondary station operates under the control of the primary station. Frames issued by a secondary are called responses. The primary maintains separate logical links with each secondary station of the line.
3) **Combined station :** It combines the features of primary and secondary. A combined station may issue both commands and **responses**.

- Since HDLC has been defined as a general purpose data link control protocol. The stations can be configrated in different network configurations as
  i) Point-to-point with single primary and secondary.
  ii) Multipoint with single primary and multiple secondaries.
  iii) Point-to-point with two primaries and two secondaries.

  All above configurations are illustrated in Fig. 5.3.1 (a) to 5.3.1 (c).

i) Point-to-point with single primary and secondary

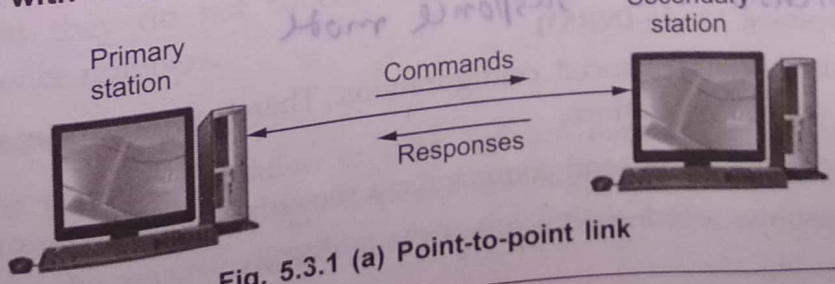


Fig. 5.3.1 (a) Point-to-point link

Scanned with CamScanner

ii) Multipoint with single primary and multiple secondaries

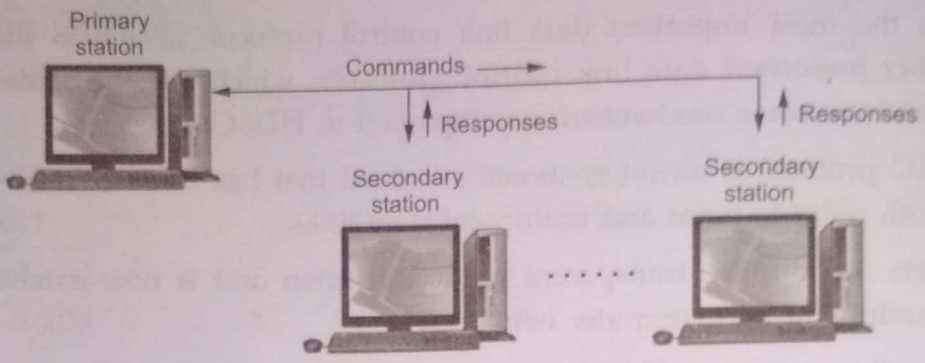


Fig. 5.3.1 (b) Multipoint link

iii) Point-to-point with two primaries and two secondaries

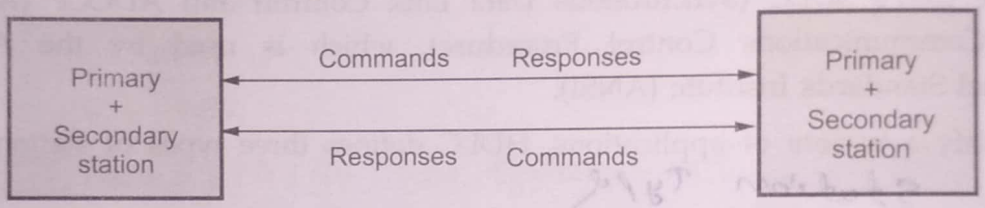


Fig. 5.3.1 (c) Point-to-point link between combined stations

- The frames sent by primary station to the secondary station are known as **commands** and those from the secondary to the primary as **responses**.
- Two configurations shown in part (i) and (ii) have a single primary station are known as **unbalanced configurations.** Unbalanced configuration supports both full duplex and half duplex transmission.
- The configuration in part (iii) has two primary stations and is known as **balanced configuration**. Balanced configuration supports both full duplex and half duplex transmission. Since each station has both a primary and a secondary, they are also known as **combined stations.**

5.3.1 Operational Mode of HDLC

HDLC has following data transfer modes :

1) Normal Response Mode (NRM).
2) Asynchronous Balanced Mode (ABM).

– Asynchronous response mode

1) Normal Response Mode (NRM)

- This is used in unbalanced configurations. There are one primary station and multiple secondary stations.
- A primary station can send commands; a secondary station can only respond.
- Fig. 5.3.2 shows a Normal Response Mode (NRM).

Scanned with CamScanner

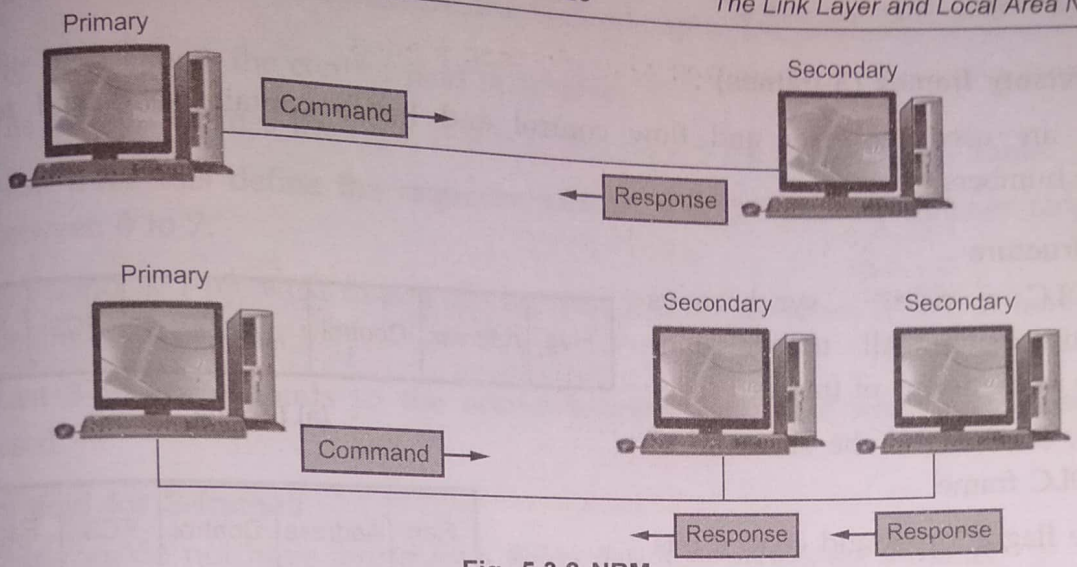


Fig. 5.3.2 NRM

- The NRM is used for both point-to-point and multipoint links.

2) Asynchronous Balanced Mode (ABM)

- In ABM, the configuration is balanced. The link is point to point and each station can functions as a primary and a secondary.
- Fig. 5.3.3 shows the ABM.

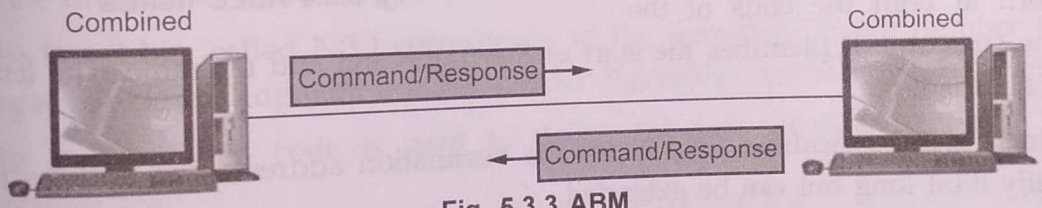


Fig. 5.3.3 ABM

- Either station can send data, control information or commands. This is typical in connections between two computers and in the X.25 interface standard.

5.3.2 Frames

In HDLC both data and control messages are carried in a standard format frame. Three classes of frame are used in HDLC.

1) Unnumbered frames (U-frames) :

These are used for functions such as link setup and disconnection. The name derives from the fact that they do not contain any acknowledgment information, which is contained in sequence numbers.

2) Information frames (I-frames) :

These carry the actual information or data and are normally referred to simply as I-frames. I-frames can be used to piggy back acknowledgment information relating to the flow of I-frames in the reverse direction when the link is being operated in ABM or ARM.

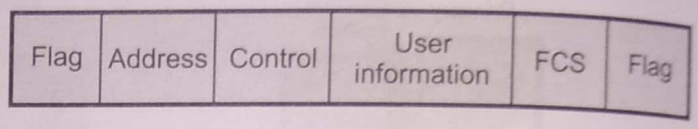
Scanned with CamScanner

3) Supervisory frames (S-frames) :

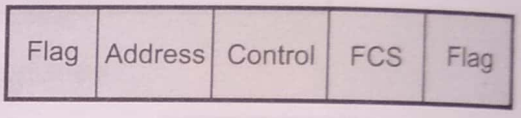
These are used for error and flow control and hence contain send and receive sequence numbers.

Frame structure

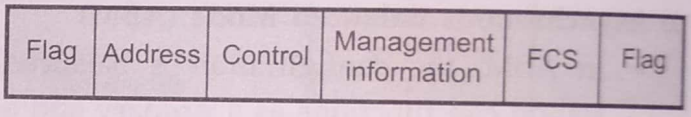
- HDLC uses synchronous transmission. All transmissions are in the forms of frames.
- Fig. 5.3.4 shows the structure of HDLC frame.
- The flag address and control bits before the information or data fields are known as a header. The FCS and flag fields following the data fields are referred as a trailer.



(a) I-frame



(b) S-frame



(c) U-frame

Fig. 5.3.4 HDLC frames

- **Flag fields :** It has a unique pattern at both the ends of the frame structure. It identifies the start of the frame and end of frame. The length of flag field is 8-bit.
- **Address fields :** Address field states the destination address. The address field is usually 8-bit long but can be extended.
- **Control fields :** Control fields contain frame numbers. Also it controls the acknowledgment of frames. Control field is 8 or 16 bits in length.
- **Information fields :** Data field contains the user data received from the network layer. It can be of variable length but in integral number of octets.
- **FCS (Frame Check Sequence) :** FCS is an error detecting code calculated from the remaining bits of the frame. FCS can be 16 bits or 32 bits long.

5.3.3 Control Field

Control field for I-frames

- I-frames are designed to carry user data from the network layer. This field also include flow and error control information.

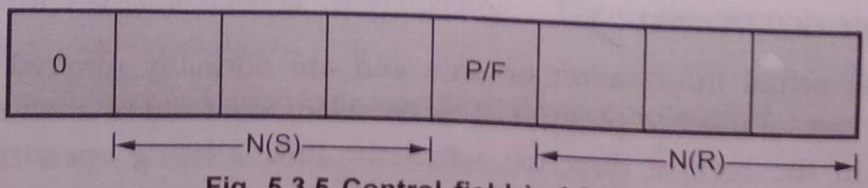


Fig. 5.3.5 Control field in I-frame

Scanned with CamScanner

- Fig. 5.3.5 shows the control field in I-frames.
- The first bit defines the type. If it is 0, this means the frame is an I-frame.
- Next three bits define the sequence number (NCS). Sequence number range is in between 0 to 7.
- P/F field is 1-bit with dual purpose. This field is set when it is 1. It may be poll or final.
- Last 3-bit corresponds to the acknowledgement number when piggy backing is used.

Control field for S-frames

- S-frames do not have information fields. Fig. 5.3.6 shows the S-frame.

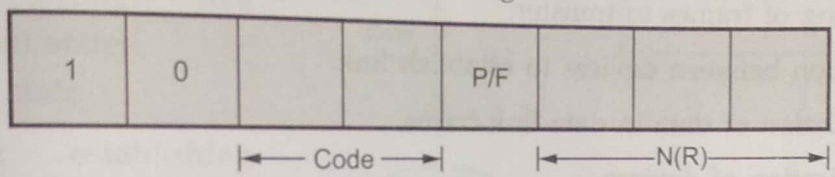


Fig. 5.3.6 S-frame

- If the first 2 bits of the control field is 10. This means the frame is an S-frame.
- The last 3 bits called N(R) corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame.
- The 2 bits called code is used to define the type of S-frame itself. Types of S-frames are :
  1) Receive Ready (RR)
  2) Receive Not Ready (RNR)
  3) Reject (REJ)
  4) Selective Reject (SREJ)

Code	S-frame type	Remarks
00	Receive ready	N(R) define ACK number
10	Receive not ready	N(R) define ACK number
01	Reject	N(R) define NAK number
11	Selective reject	N(R) define NAK number

Control Field for U-frames

- U-frames contain an information field, but one used for system management information, not user data.
- Fig. 5.3.7 shows U-frames.

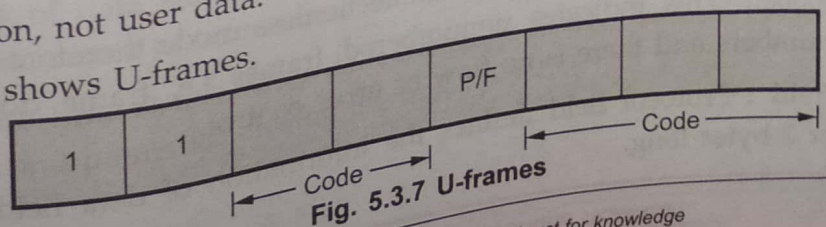


Fig. 5.3.7 U-frames

Scanned with CamScanner

- U-frames codes are divided into two sections a 2-bit prefix before a P/F and a 3-bit suffix after the P/F bit.

University Questions

1. *Explain HDLC.* **GTU : Dec.-11, Marks 7**
2. *Explain working principle of HDLC.* **GTU : Summer-13, Marks 7**

5.4 Point-to-Point Protocol

- PPP is the most commonly used protocol for point-to-point transfer of data. The services provided by PPP are
  1) Formatting of frames to transfer.
  2) Negotiation between devices to establish link.
  3) Encapsulation of data in data link frame.
  4) Authentication of devices.
- PPP can operate between point-to-point transmission link in full duplex mode. Also PPP can be used as a data link control to connect two routers.

5.4.1 Frame Format

- PPP frame format is similar to HDLC. Fig. 5.4.1 shows PPP frame format. It has seven fields.

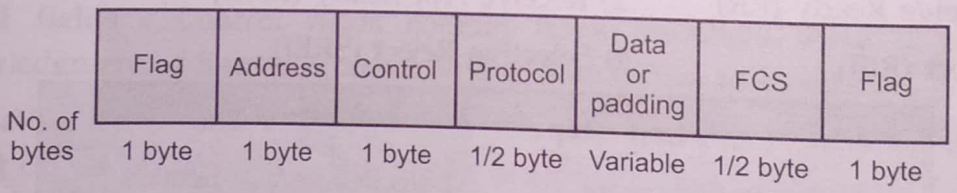


Fig. 5.4.1 PPP frame format

1) **Flag field :** The flag field identifies the boundaries of PPP frame i.e. each frame begins and ends with flag field. This field is 1 byte in length.
2) **Address field :** Address field indicates the address of destination. Address field is 1 byte (8-bits). When the address field contains "all 1's" i.e. 11111111, this indicates that all stations are to accept the frames (broadcast).
3) **Control field :** PPP normally runs in connectionless mode therefore control field is set to 11000000. This indicates unnumbered frames i.e. frame does not contain sequence numbers and there is no flow or error control.
4) **Protocol field :** Protocol field defines the information of data field. The protocol field is 1 or 2 bytes long.

Scanned with CamScanner

5) **Data field :** The data field contains the actual data to transmit. The length of this field is variable.

6) **Frame Check Sequence (FCS) :** The FCS field is 24 byte long and contains CRC code. It checks length of all fields in frame.

5.4.2 Transition States

- The transition state is used to indicate the phases through which PPP connection passes. Fig. 5.4.2 shows PPP transition states.
- The PPP connection passes through five important states.
  1) Idle state
  2) Link establishing state
  3) Authenticate state
  4) Exchange of data state
  5) Terminate link state.

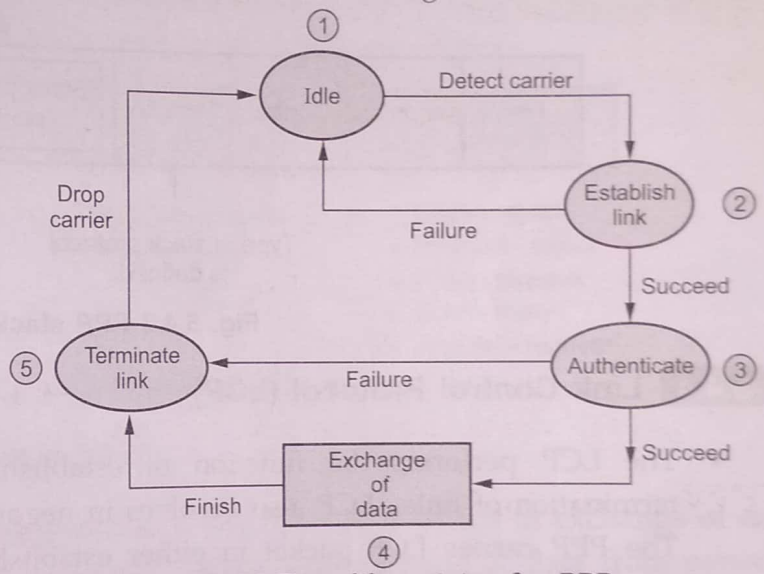


Fig. 5.4.2 Transition states for PPP

1) **Idle state :** In the idle state the link is not in use. The carrier is not activated in this state.

2) **Link establishing state :** When carrier is detected, one of the end points starts the transmission then connection enters into the link establishing state. Under this state there is negotiation between the devices. On successful negotiation, the connection enters into authenticate state otherwise it enters into idle state.

3) **Authenticate state :** The authenticate state is mutually decided by the stations. The stations sent several authentication packets. On successful authentication, the connection enters into exchange of data state otherwise to the terminate link state.

4) **Exchange of data state :** This state is also referred as networking state. In this state exchange of data started. The connection is terminated only after the any of the end points wants to terminate.

5) **Terminate link state :** After data exchange is over several packets are exchanged between and points for closing the link.

Scanned with CamScanner

5.4.3 PPP Stack

- PPP uses a stack of other protocols for establishing link and to authentications. Two major protocols are used in PPP stack. These protocols are -
  1) Link Control Protocol (LCP) 2) Network Control Protocol (NCP)
- During connection a PPP packet carry any of these protocols in its data field as shown in Fig. 5.4.3.

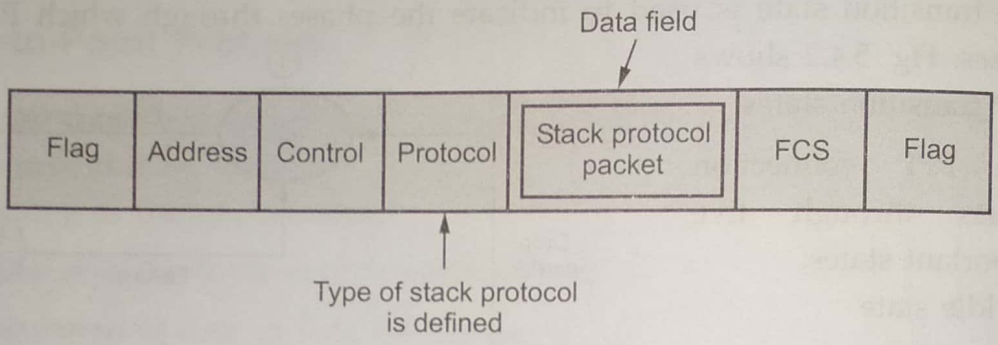


Fig. 5.4.3 PPP stack

5.4.4 Link Control Protocol (LCP)

- The LCP performs the function of establishing, maintaining, configuring and termination of links. LCP also involves in negotiating mechanism between stations. The PPP carries LCP packet in either establishing or terminating state i.e. when user data is not carried. Fig. 5.4.4 shows the frame format of LCP packet and how it is encapsuled in PPP frame.

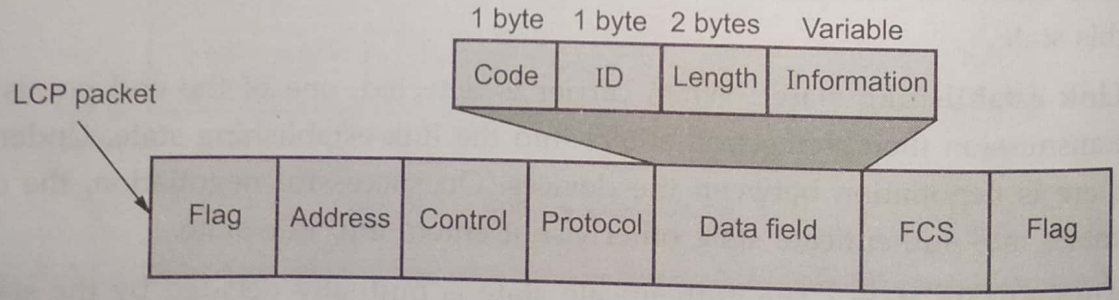


Fig. 5.4.4 LCP frame in PPP frame

1) **Code field :** Code field is 1 byte in length. The code field defines the type of LCP packet. There are three types of LCP packets - Configuration packets, link termination and link monitoring packets.

2) **ID :** ID field is 1 byte in length. ID field is used to match the request packet with its reply packet. The request end point inserts a value in this field which is copied in corresponding field in reply packet.

3) **Length :** The length field is 2 bytes. It defines the entire length of LCP packet.

Scanned with CamScanner

4) Information : This is a variable length field. Any additional information needed by LCP packet is stored in this field.

LCP packet types

- The LCP packets can be categorized according to their function. Fig. 5.4.5 shows types of LCP packets.

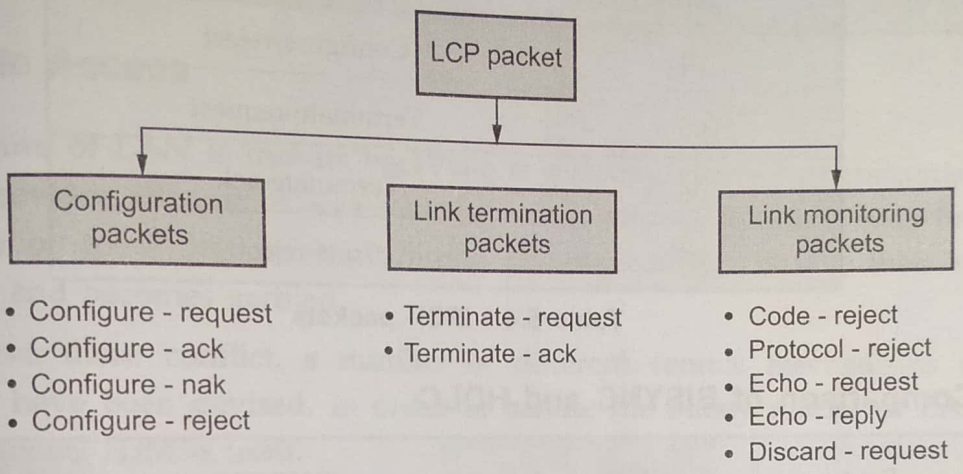


Fig. 5.4.5 LCP packet types

5.4.5 Network Control Protocol (NCP)

- The PPP uses Network Control Protocol (NCP) when it enters in exchange of data state. NCP is a set of protocols which allows encapsulation of data from network layer into PPP frame.

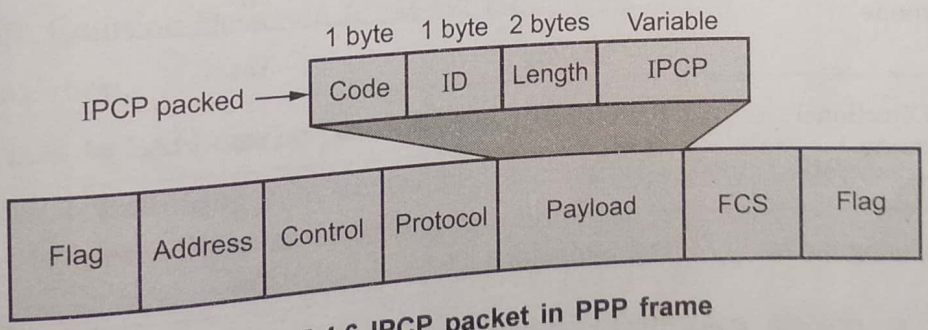


Fig. 5.4.6 IPCP packet in PPP frame

- PPP extends the negotiation not only in data link layer but in network layer also. The set of packets that establish and terminate a network layer connection for IP packets is called **Internetwork Protocol Control Protocol (IPCP).** The format of IPCP packet is shown in Fig. 5.4.6.
- The protocol field value for IPCP packet is $(8021)_H$. There exists seven types of IPCP packets, each having unique code value (1 byte). IPCP packets and their corresponding code values are shown in Table 5.4.1.

Scanned with CamScanner

Code value	IPCP packets
01	Configure-request
02	Configure-ack
03	Configure-nak
04	Configure-reject
05	Terminate-request
06	Terminate-ack
07	Code-reject

Table 5.4.1 IPCP packets

5.4.6 Comparison of BISYNC and HDLC

Sr. No.	Feature	Characteristic	BISYNC	HDLC
1.	Transmission	Serial	Yes	Yes
		Asynchronous	No	No
		Synchronous	Yes	Yes
2.	Communication mode	Asynchronous	No	Yes
		Synchronous	Yes	Yes
3.	Directional mode	TWA	Yes	Yes
		TWS	No	Yes
4.	Configuration	Point-to-point	Yes	Yes
		Point-to-multipoint	Yes	Yes
5.	Flow control	-	Stop and wait	Sliding windows
6.	Flow control	Content errors	LRC/CRC	CRC
7.	Error detection and correction	Flow integrity errors	ACK-0/ACK-1	Sequence number
8.	Code set	-	ASCII/EBCDIC/Transcde	Any
9.	Control Character	-	Many	None

Scanned with CamScanner

10.	Framing	Frame identifier	SYN SYN	Flag
		Frame delimiter	ETB/ETX	Flag
		Information field	Multiple bytes	Multiple bits
		Transparency	DLE stuffing	Zero stuffing

5.5 Multiple Access

- One feature of LAN is that its backbone is a shared channel or transmission link, which provides all user to access to the transmission facilities. It may be possible that two or more stations transmitting simultaneously, causing their signals to interfere and becomes garbled.
- To resolve these conflict, a number of different control mechanisms or access protocol have been devised. In order to handle the bursty nature of LAN, traffic asynchronous TDM is used.
- The asynchronous TDM mechanism is further divided into contention methods (random access) and deterministic methods (controlled methods).

Random access techniques are

i) ALOHA

ii) Carrier Sense Multiple-Access (CSMA)

iii) CSMA with Collision-Detection (CSMA/CD)

iv) Register insertion.

Controlled access to LAN can be performed in two types :

1) **Centralized technique :** In centralized technique master node decides which node is to access the channel at any one time.

e.g. Polling.

2) **Distributed technique :** In distributed technique each station is given an opportunity to transmit on the channel.

e.g. i) Token passing method ii) Slotted ring method.

- Fig. 5.5.1 illustrates the typical multiple access communications where a number of user stations share a transmission medium.
- This sharing techniques are used in wired communications, and networks based on radio communication.
- In wired communication multidrop cables are used in data networks to connected host computer.

Scanned with CamScanner

- The host computer broadcasts information to the users on the outbound line.
- The stations transmit information to the host using the inbound line. This system is illustrated in Fig. 5.5.2.

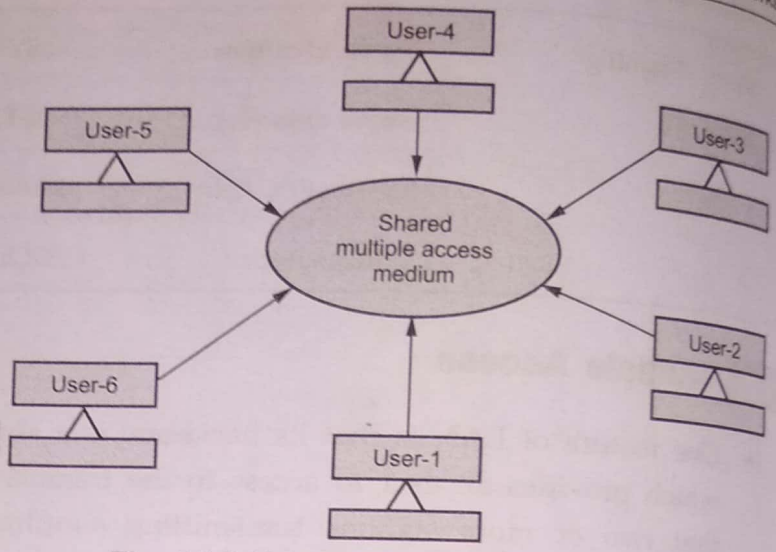


Fig. 5.5.1 Multiple access communication

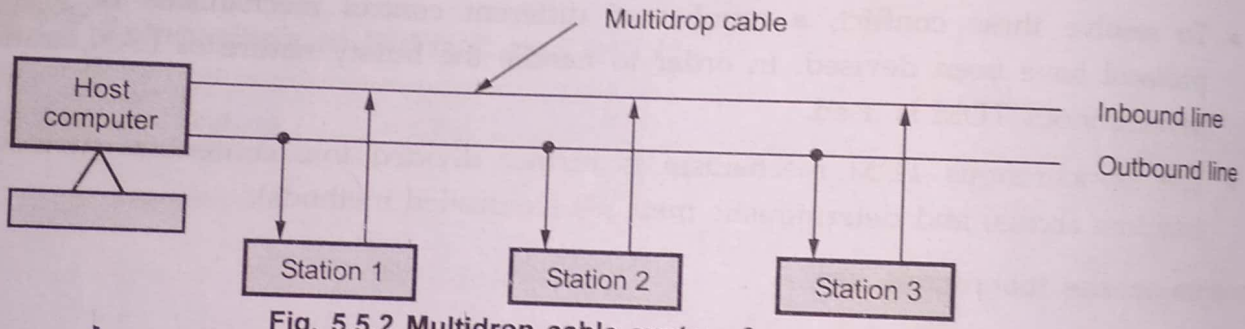


Fig. 5.5.2 Multidrop cable system for access control

- A Medium Access Control (MAC) protocol is developed for this system. Here the host computer issues polling messages to each stations, providing it with permission to transmit on the inbound line.

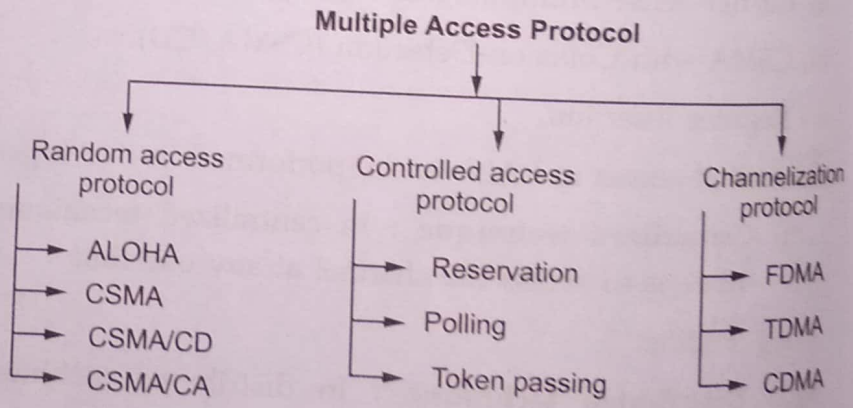


Fig. 5.5.3

- In radio communication several stations share two frequency bands, one for transmitting and one for receiving.
- In satellite communications each station is assigned a channel in an uplink frequency band that it uses to transmit to the satellite. The satellite sends back the signals on different frequency band called down link frequency band.

Scanned with CamScanner

5.6 Random Access

GTU : Dec.-10, June-11, May-12, Summer-13, 14, Winter-13,14

- Access to the medium from many entry points is called contention. It is controlled with a **contention protocol.**
- In a random access method, each station has the right to the medium without being controlled by other station. However if more than one station tries to send, there is an access conflict, i.e. collision and the frames will be either destroyed or modified.

5.6.1 ALOHA

Any time data transfer
- Collisions
- ACK

- The ALOHA protocol was developed at the university of Hawaii in the early 1970s. ALOHA was developed for packet radio networks. However, it is applicable to any shared transmission medium.
- In a system when multiple users try to send messages to other stations through a common broadcast channel random access or contention techniques are used.
- Random access means there is no definite or scheduled time for any station to transmit. This scheme is simplest possible and it is asynchronous. It is asynchronous because there is no co-ordination among users.
- The basic idea of ALOHA system is applicable to any system in which unco-ordinated users are competing for the use of a single shared channel.
- When a station send data, another station may attempt to do so at the same time. The data from the two station collide and become garbled. If two signals collided, so be it. Each station would simply wait a random time and try again.

5.6.1.1 Pure ALOHA

Type

- The original ALOHA protocol is called pure ALOHA. The idea is that each station sends a frame whenever it has a frame to send. Since there is only one channel to share, there is the possibility of collision between frames from different stations.
- Fig. 5.6.1 shows the frame collisions in pure ALOHA.
(See Fig. 5.6.1 on next page)
- The pure ALOHA protocol relies on acknowledgements from the receiver. When a user sends a frame, it expects the receiver to send an acknowledgement. If the acknowledgement does not arrive after a time out period, the station assumes that the frame has been destroyed and resends the frame.
- 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.

Scanned with CamScanner

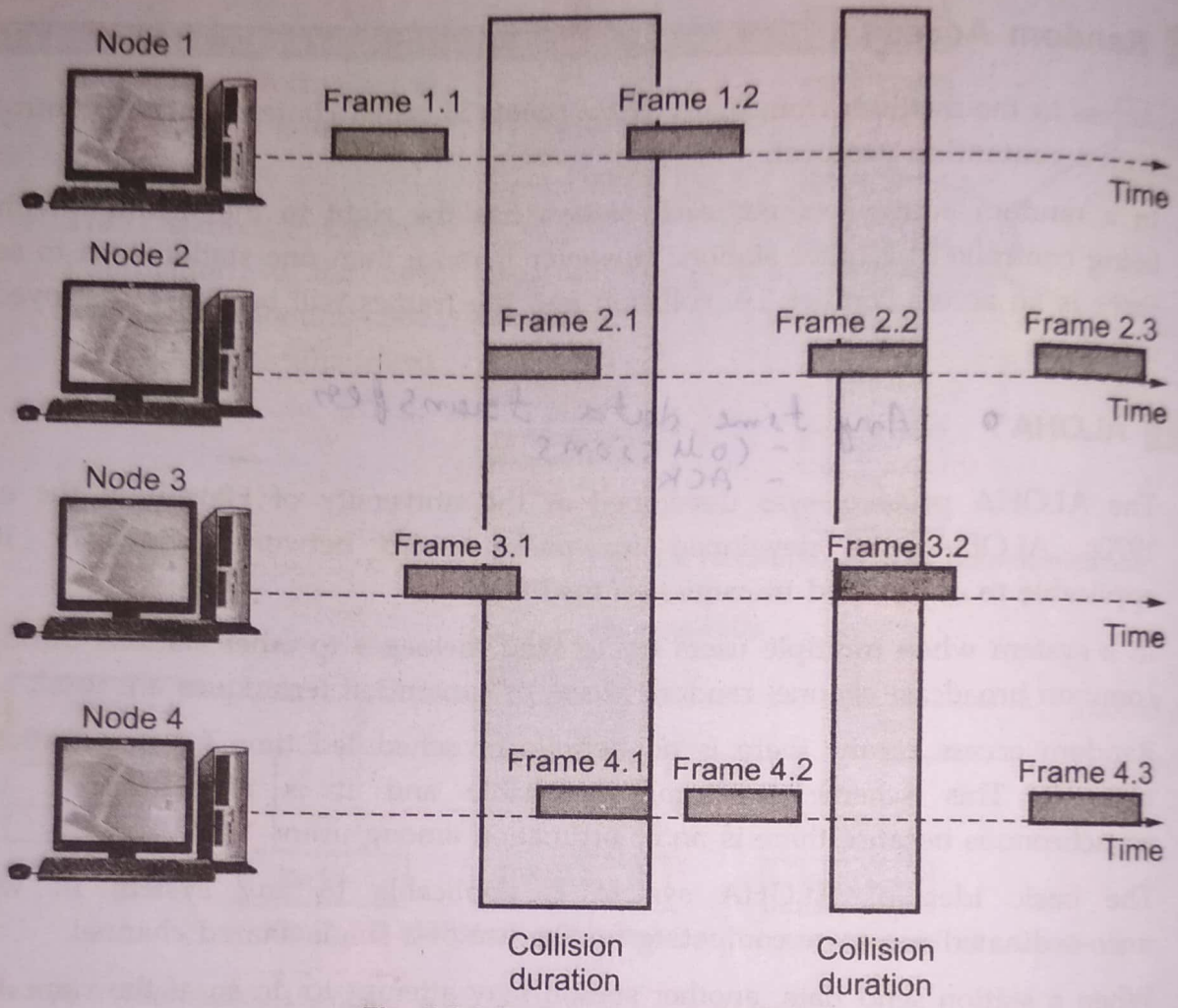


Fig. 5.6.1 Frames in pure ALOHA

- If all users try to resend their frames after the time-out, the frames will collide again. Pure ALOHA dictates that when the time-out period passes, each user waits a random amount of time before resending its frame. The randomness will help to avoid more collisions. This time is called as back-off time (T_B).
- Fig. 5.6.2 shows the procedure for pure ALOHA protocol.
(See Fig. 5.6.2 on next page)
- The time out period is equal to the maximum possible round trip propagation delay, which is twice the amount of time required to send a frame between the two most widely separated stations ($2 \times T_p$).
- Let all the packets have the same length. And each requires one time unit for transmission (t_p). Consider any user to send packet A at time t_o. If any other user B has generated a packet between time t_o and $t_o + t_p$, the end of packet B will collide with the beginning of packet A. Since in pure ALOHA packet, a station does not listen to the channel before transmitting, it has no way of knowing that above frame was already under way.

Scanned with CamScanner

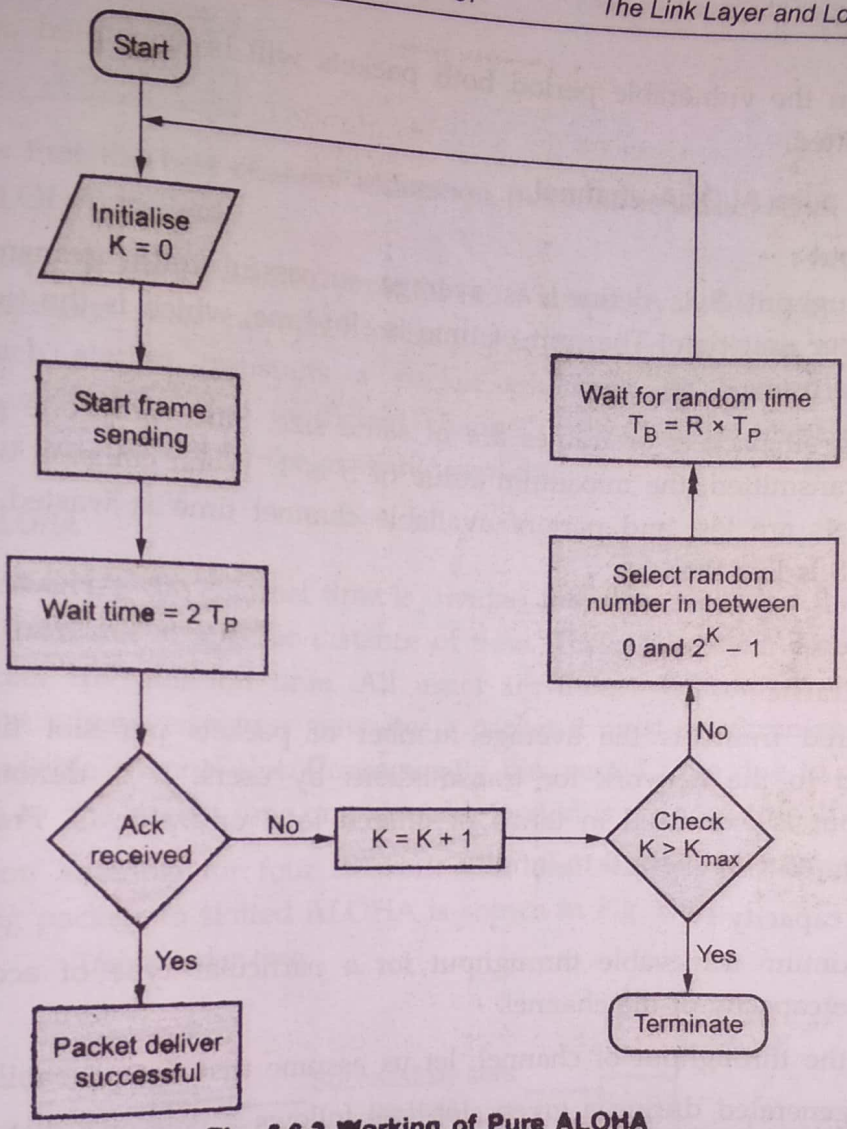


Fig. 5.6.2 Working of Pure ALOHA

- Fig. 5.6.3 shows vulnerable periods during which packets can collide.
- Similarly if another user wants to transmit between $(t_o + t_p)$ and $(t_o + 2t_p)$ i.e. packet C, the beginning of packet C will collide with the end of packet A. Thus if two packets overlap by even the smallest

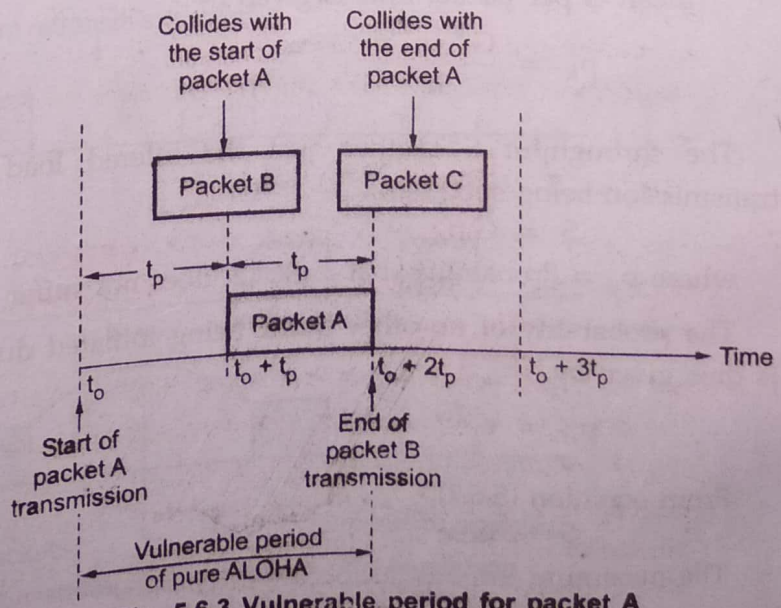


Fig. 5.6.3 Vulnerable period for packet A

Scanned with CamScanner

amount in the vulnerable period both packets will be corrupted and need to be retransmitted.

Throughput of pure ALOHA channel

1) Throughput :

- The throughput S is defined as average successful traffic transmitted between stations per unit time. The unit of time is slot-time, which is the time required to transmit a frame.
- Assuming all packets or frames are of same size. Since only one packet per slot can be transmitted, the maximum value of S is 1. When collision occurs, some of the packets are lost and part of available channel time is wasted. The resulting value of S is less than 1.

$$S = G \times e^{-2G}$$

2) Offered traffic :

- The offered traffic is the average number of packets per slot time which are presented to the network for transmissions by users. It is denoted by G. The throughput is expressed in terms of offered load or traffic G. Practically G can have any value between 0 to infinity.

3) Channel capacity :

- The maximum achievable throughput for a particular type of access scheme is called the capacity of the channel.
- To find the throughput of channel, let us assume that the probability (p_k) that k packets generated during a given slot-time follows a Poisson's distribution with a mean G per packet time is given by

$$p_k = \frac{G^k \cdot e^{-G}}{k!} \quad \ldots (5.6.1)$$

The throughput S is then just the offered load G times the probability of a transmission being successful.

$$\therefore \quad S = G p_o \quad \ldots (5.6.2)$$

where p_o = Probability that a packet does not suffer a collision

The probability of no other traffic being initiated during the entire vulnerable period is thus given by

$$p_o = e^{-2G} \quad \ldots (5.6.3)$$

From equation (5.6.2),

$$S = G \cdot e^{-2G}$$

The maximum throughput occurs at G = 0.5,

Scanned with CamScanner

i.e. $$S = \frac{1}{2e} = 0.184$$

- This means that the best channel utilization that can be achieved is around 18 % for pure ALOHA method.
- The advantage of pure ALOHA protocol is its simplicity, which can result in low cost user stations since no synchronization is required between stations in the system. Each station transmits a packet whenever its buffer has one. The disadvantage is somewhat inefficient channel utilization i.e. maximum channel utilization is only 18.4 % of the available capacity.

5.6.2 Slotted ALOHA

- In slotted ALOHA, the channel time is divided into time slots and the stations are allowed to transmit at specific instance of time. These time slots are exactly equal to the packet transmission time. All users are then synchronized to these time slots, so that whenever a user generates a packet it must synchronize exactly with the next possible channel slot. Consequently the wasted time due to collisions can be reduced to one packet time or vulnerable period is reduced to half.
- Transmission attempts for four network user and random retransmission delays for colliding packets in slotted ALOHA is shown in Fig. 5.6.4.

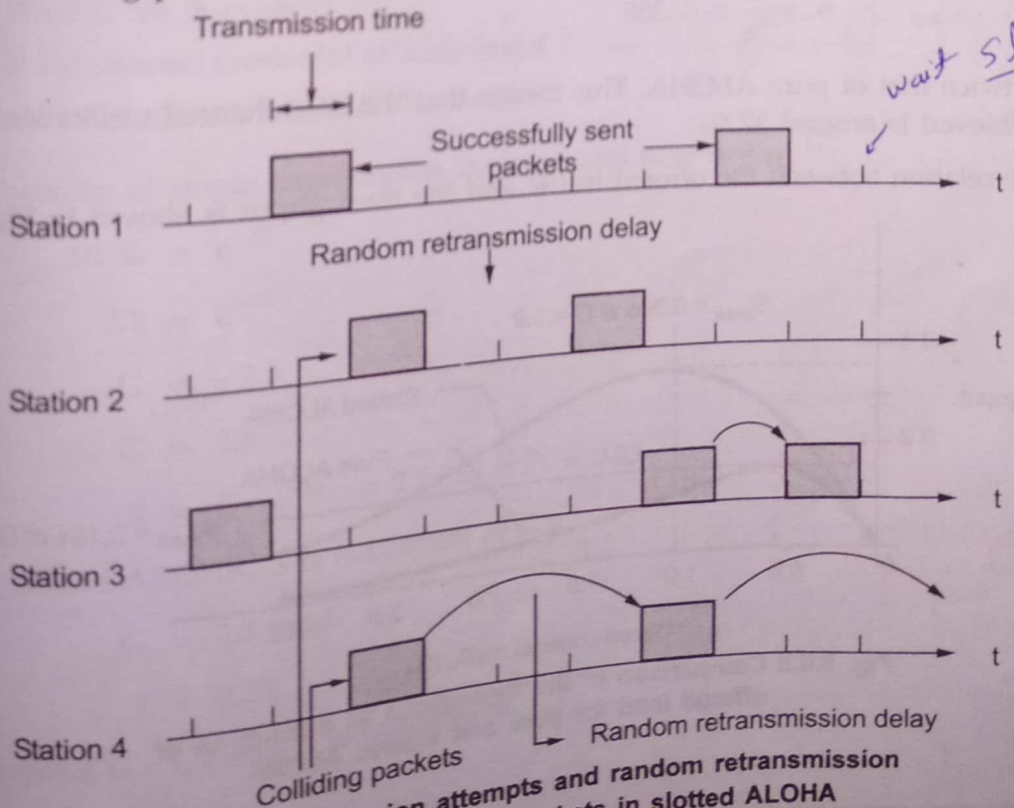


Fig. 5.6.4 Transmission attempts and random retransmission delays for colliding packets in slotted ALOHA

Scanned with CamScanner

Slotted ALOHA

Assumptions :

1. All frames are of same size.
2. Time is divided into equal sized slots, a slot equals the time to transmit one frame.
3. Nodes start to transmit frames only at beginning of slots.
4. Nodes are synchronized.
5. If two or more nodes transmit in a slot, all nodes detects collision before the slot ends.

Throughput of slotted ALOHA channel

- In slotted ALOHA, the packets arrive in a synchronized fashion. The probability of single transmission during a slot time is

$$P_o = e^{-G} \quad \ldots (5.6.4)$$

From equation (5.6.2)

$$S = G \cdot e^{-G} \quad \ldots (5.6.5)$$

The maximum throughput occurs at G = 1,

i.e. $$S = \frac{1}{e} = 0.368$$

which is twice that of pure ALOHA. This means that the best channel utilization that can be achieved is around 37 %.

- The relation between the offered traffic and the throughput is shown in Fig. 5.6.5.

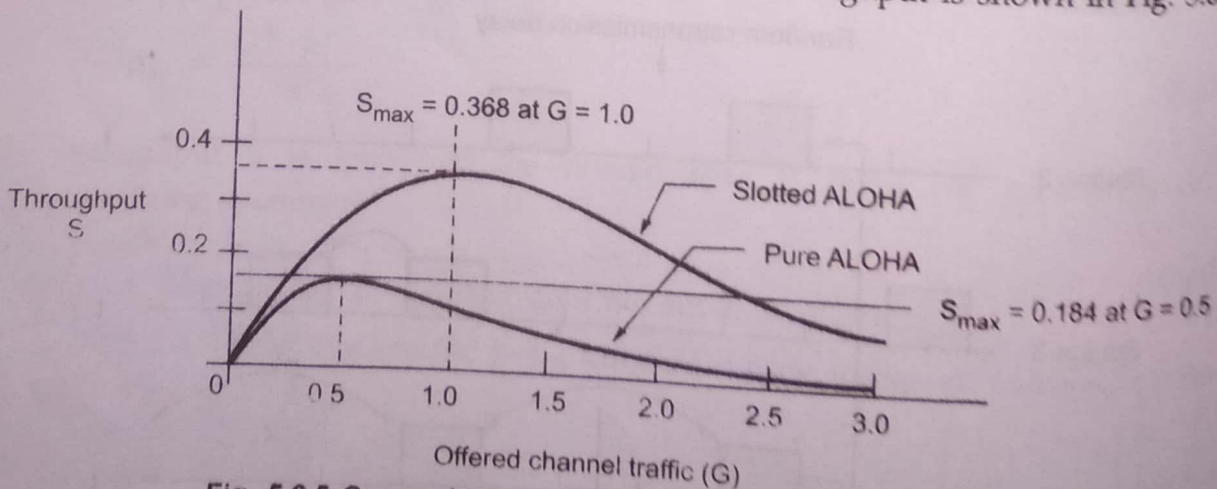


Fig. 5.6.5 Comparison of the throughput as a function of offered load for pure and slotted ALOHA

Scanned with CamScanner

Pros and Cons of slotted ALOHA

Pros

1. Single active node can continuously transmit at full rate of channel.
2. Highly decentralized, each node independently decides when to retransmit.
3. Simple to implement.

Cons

1. Collisions waste slots. 2. Idle slots.

Example 5.6.1 *Calculate the throughput S for a pure ALOHA network if the offered traffic G is 0.75.*

Solution : For pure ALOHA, throughput S is given by,

$$S = G \cdot e^{-2G}$$

$$\therefore \quad S = 0.75 * e^{-2 * 0.75} = 0.1673$$

or $S = 16.73\ \%$

Example 5.6.2 *A slotted ALOHA channel has an average 10 % of the slots idle.*

a) What is the offered traffic G ?

b) What is the throughput ?

c) Is the channel overloaded or underloaded ?

Solution : For slotted ALOHA channel,

a) Probability of single transmission during a slot time is $= e^{-G}$.

$$10\ \% = e^{-G}$$

$$0.1 = e^{-G}$$

$$\Rightarrow \quad -G = -2.3$$

$$\therefore \quad G = 2.3$$

b) $S = G * e^{-G} = 2.3 * e^{-(2.3)} = 0.23$

c) For slotted ALOHA, S is maximum at G = 1.

Here $G = 2.3$ and

$S = 0.23$

It is beyond G = 1, hence it is **overloaded**.

Scanned with CamScanner

5.6.1.3 Difference between Pure ALOHA and Slotted ALOHA

Sr. No.	Pure ALOHA	Slotted ALOHA
1.	Frames are transmitted at arbitrary time.	Time is divided up into discrete slot, the frame is sent at the start of a slot.
2.	Throughput (s) = $G \times e^{-2G}$	Throughput (s) = $G \times e^{-G}$
3.	Vulnerable time is 2 times the frame transmission time.	Vulnerable time is one half that of pure ALOHA.
4.	The maximum utilization is about 18.4 %.	The maximum utilization is about 36.8 %.
5.	Global time is not required.	It requires global time for synchronization, as it is divided up into discrete slot.
6.	Simple to implement.	Implementation is complex due to the synchronization of all nodes.
7.	Cannot used for satellite, due to very low utilization.	It is used in broadcast satellites.

5.6.2 Carrier Sense Multiple Access Protocol [CSMA]

— No ACK

- The low maximum throughput of the ALOHA schemes is due to the wastage of transmission bandwidth because of frame collisions. This wastage can be reduced by avoiding transmissions that are certain to cause collisions. By sensing the medium for the presence of a carrier signal from other stations, a stations can determine whether there is an ongoing transmission.
- CSMA requires that each station first listen to the medium before sending. CSMA can reduce the possibility of collision, but it cannot eliminate it.
- The possibility of collision still exists because of propagation delay. A station may sense the medium and find it idle, only because the first bit sent by another station has not yet been received.

Vulnerable Time

- The vulnerable time for CSMA is the propagation time (T_p). This is the time needed for a signal to propagate from one end of the medium to the other.
- Fig. 5.6.6 shows the vulnerable time for CSMA. Station A sends a frame at time t_1, which reaches the rightmost station D at time $t_1 + T_p$.

Scanned with CamScanner

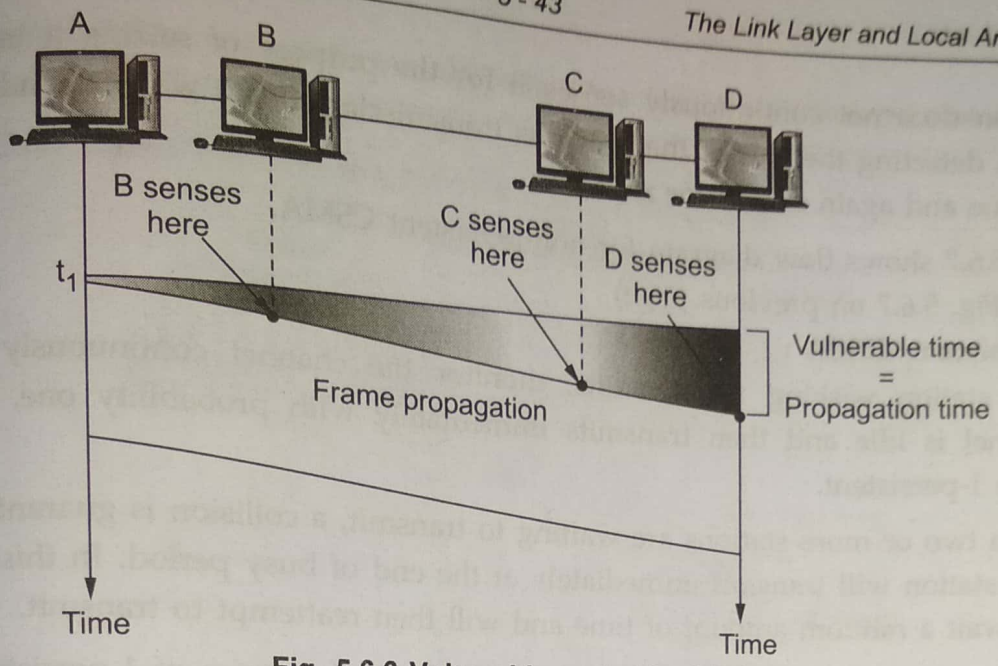


Fig. 5.6.6 Vulnerable period in CSMA

Persistence Methods

- These three protocols are
  1) Non-persistent CSMA.
  2) 1-persistent CSMA.
  3) p-persistent CSMA.

All three protocols are differing by the action to be taken by any station after sensing the readiness of the channel.

1) Non-persistent CSMA :

- In non-persistent CSMA, when a station having a packet (frame) to transmit and finds that the channel is busy, it backs off for a fixed interval of time. It then checks the channel again and if the channel is free then it transmits. The back-off delay is determined by the transmission time of a frame, propagation time and other system parameters. If the channel is already in use, the

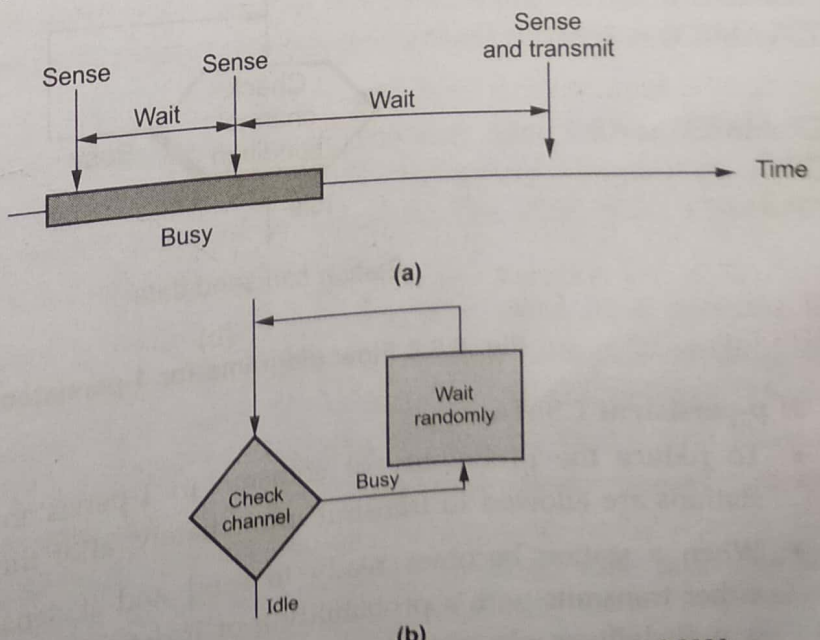


Fig. 5.6.7 Flow diagram for non-persistent CSMA

Scanned with CamScanner

station does not continuously senses it for the purpose of seizing it immediately upon detecting the end of the previous transmission. But it waits a random period of time and again checks for activity.

- Fig. 5.6.7 shows flow diagram for non-persistent CSMA.
(See Fig. 5.6.7 on previous page)

2) 1-persistent CSMA :

- Any station wishing to transmits, monitor the channel continuously until the channel is idle and then transmits immediately with probability one, hence the name 1-persistent.
- When two or more stations are waiting to transmit, a collision is guaranteed. Since each station will transmit immediately at the end of busy period. In this case each will wait a random amount of time and will then reattempt to transmit.
- As in the case with non-persistence CSMA, the performance of 1-persistent CSMA protocol depends on the channel delay time.
- Fig. 5.6.8 shows the flow diagrams for 1-persistent CSMA.

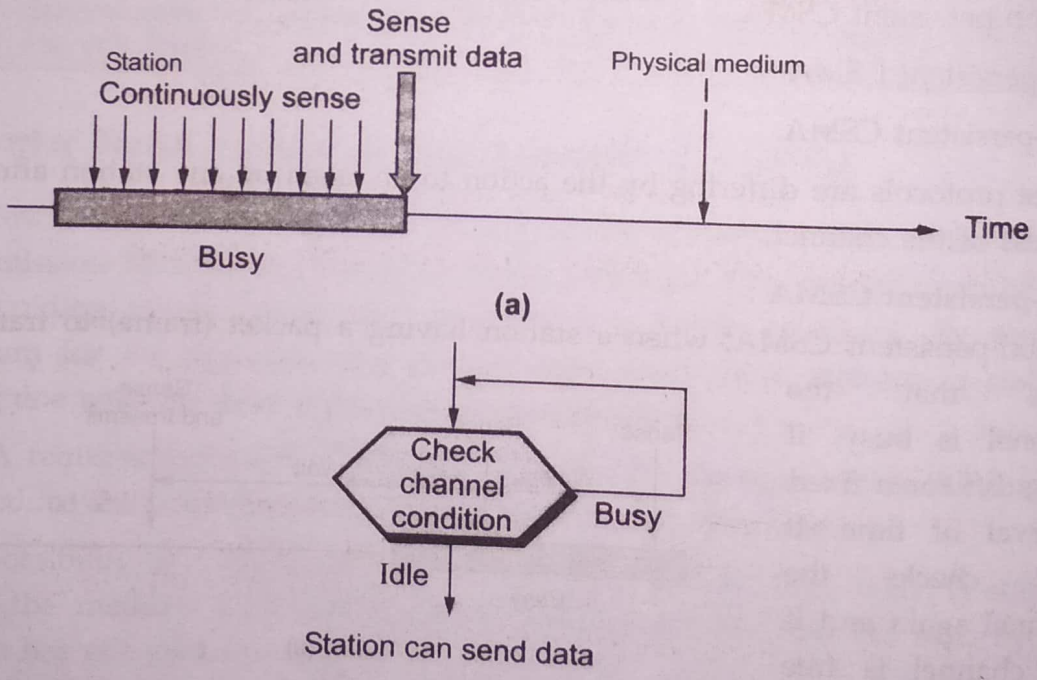


Fig. 5.6.8 Flow diagrams for 1-persistent CSMA

3) p-persistent CSMA :

- To reduce the probability of collision in 1-persistent CSMA, not all the waiting stations are allowed to transmit immediately, after the channel is idle.
- When a station becomes ready to send and it senses the channel to be idle, it either transmits with a probability p or it defers transmission by one time slot with a probability $q = 1 - p$. If the deferred slot is also idle, the station either transmits

Scanned with CamScanner

with probability p or defers again with a probability q. This process is repeated until either packets are transmitted or the channel becomes busy.

- Fig. 5.6.9 shows the flow diagram for p-persistent CSMA.

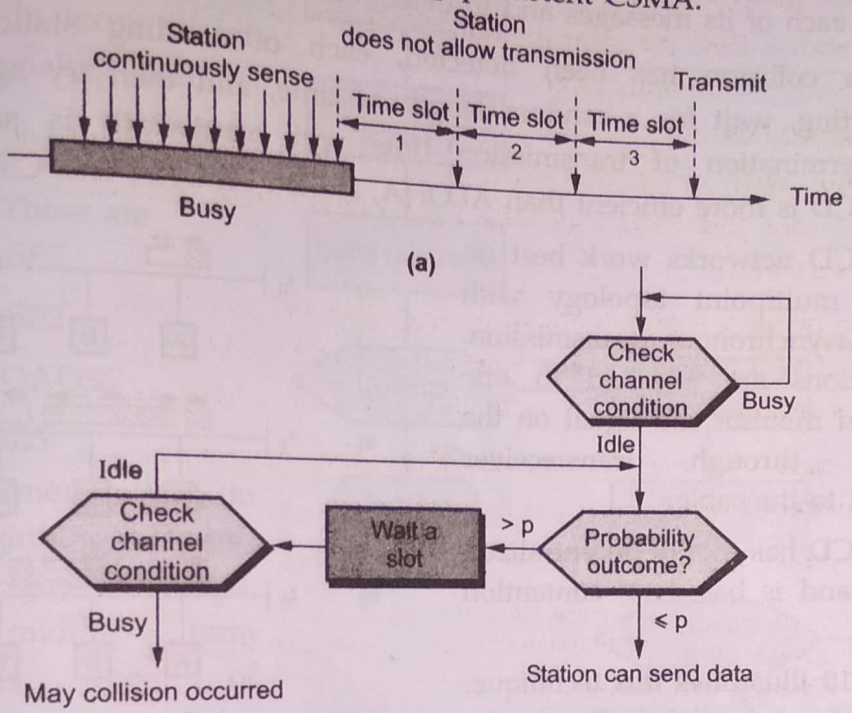


Fig. 5.6.9 Flow diagrams for p-persistent CSMA

5.6.3 Carrier Sense Multiple Access with Collision Detection

- In both CSMA and ALOHA schemes, collisions involve entire frame transmissions. If a station can determine educed by aborting the transmission when a collision is detected. The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) use this approach.
- CSMA/CD is the most commonly used protocol for LANs. CSMA/CD specifications were developed jointly by Digital Equipment Corporation (DEC), Intel and Xerox. This network is called as Ethernet. The IEEE 802.3 CSMA/CD standard for LAN is based on Ethernet specification.
- The basic protocol is that, a station with a message to send must monitor the channel to see if any other station is sending. If another station is sending, the second station must wait or defer, until the sending station has finished. Then it may send its message. If no station was sending at the time that it first listened, the station may send its message immediately. The term "carrier sense" indicates this "listening before transmitting" behaviour.
- If two or more stations have messages to send at the same time and they are separated by significant distances on the bus/channel, each may begin transmitting at roughly the same time without being aware of the other station.

Scanned with CamScanner

The signals from each station will superimpose on the channel and is garbled beyond the decoding ability of the receiving station. This is termed as "collision".

- A protocol is required for transmitting station to monitor the channel while sending each of its messages and to detect such "collisions".
- When a collision has been detected, each of sending stations must cease transmitting, wait for a random length of time, and then try again. Because of quick termination of transmission time and bandwidth is saved. Therefore CSMA/CD is more efficient than ALOHA, slotted ALOHA and CSMA.
- CSMA/CD networks work best on a bus, multipoint topology with bursty asynchronous transmission. All stations are attached to one path and monitor the signal on the channel through transreceiver attached to the cable.
- CSMA/CD has totally decentralized control and is based on contention access.
- Fig. 5.6.10 illustrates this technique. Station A and station D are the extreme ends of a bus structure.
  - i) Station A listens channel starts transmitting a packet addressing D.
  - ii) Station B and C are ready for transmission. B senses a transmission on channel so defers. C is unaware of transmission and begins its own transmission.
  - iii) Station A's transmission reaches C. C detects collision and ceases transmission. Sends jam signal.
  - iv) Effect of collision propagates back to A, A stops its transmission.
  - v) A sends jam signal.
  - vi) No station is transmitting but there are still signals on the bus.

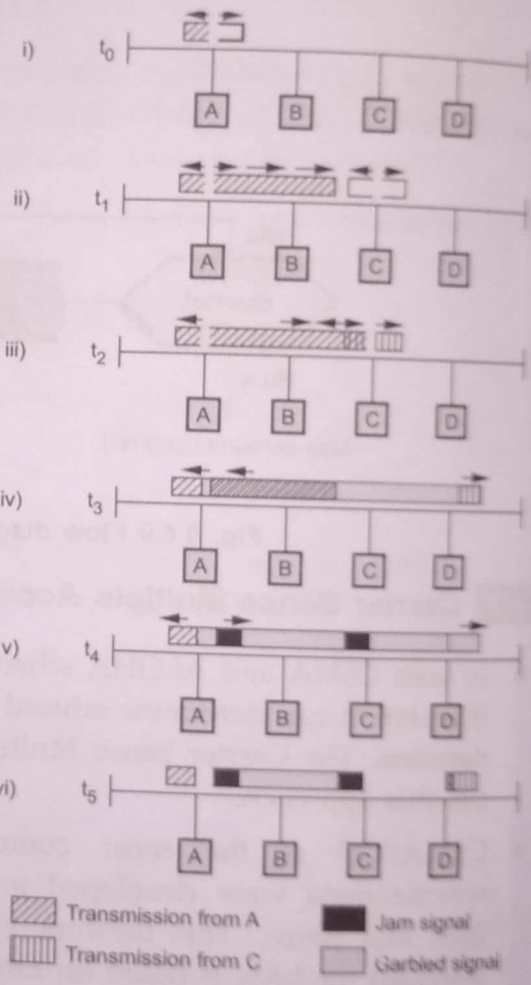


Fig. 5.6.10 CSMA/CD operation

Scanned with CamScanner

- CSMA/CD supports both baseband and broadband system. CSMA/CD offers four options in terms of bit rate, signaling method and maximum electrical cable segment length. These are
  1) 10BASE5
  2) 10BASE2
  3) 10BROAD36
  4) 1BASE5
- The numeric field in the beginning indicates the bit rate in Mbps, the middle term indicates type of signaling system i.e. baseband or broadband, the numeric field in the end indicates the electrical cable segment length in X 100 metres.

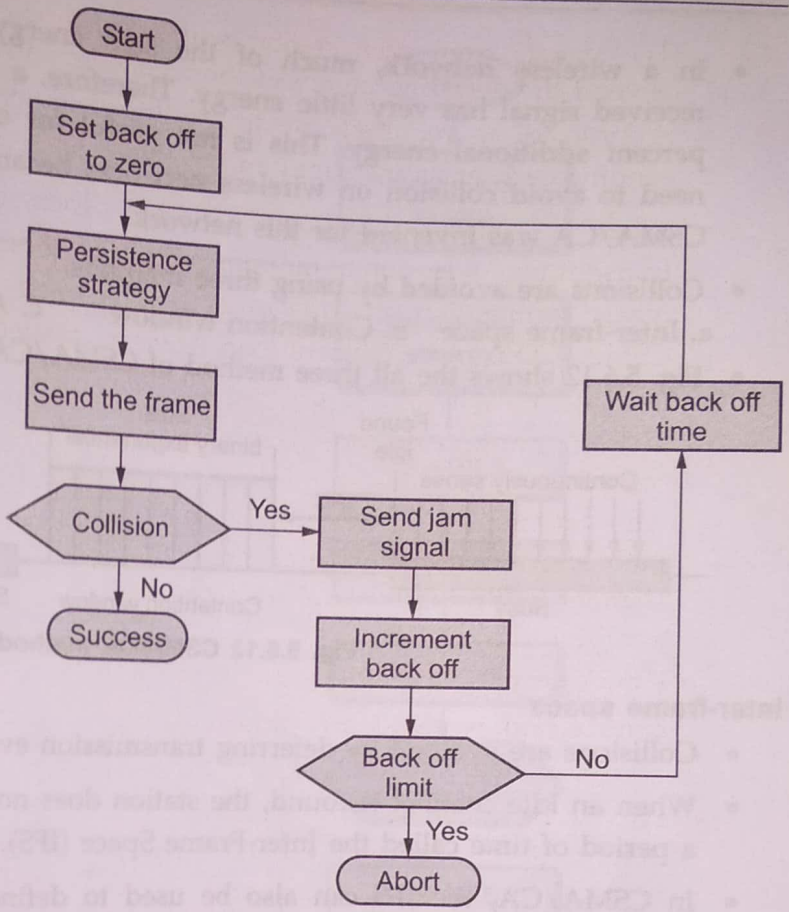


Fig. 5.6.11 Flowchart for CSMA/CD

- Manchester signal code is used at the baseband level of transmission. In broadband transmission, Differential Phase Shift Keying (DPSK) is used to convert the Manchester encoded signal into analogue form.
- Fig. 5.6.11 shows the flowchart for CSMA/CD procedure.

CSMA/CD throughput

- The throughput of CSMA/CD is greater than that of pure or slotted ALOHA.
- For 1-persistent method, the maximum throughput is around 50 % when G = 1.
- For non-persistent method, the maximum throughput can go upto 90 % when G is between 3 and 8.

5.6.4 Carrier Sense Multiple Access with Collision Avoidance

- Wireless networks cannot use CSMA/CD in the MAC sublayer, since this requires the ability to receive and transmit at the same time - hence the use of CSMA/CA.

Scanned with CamScanner

- In a wireless network, much of the sent energy is lost in transmission. The received signal has very little energy. Therefore, a collision may add only 5 to 10 percent additional energy. This is not useful for effective collision detection. We need to avoid collision on wireless networks because they cannot be detected. So CSMA/CA was invented for this network.
- Collisions are avoided by using three methods.
  a. Inter-frame space b. Contention window c. Acknowledgments
- Fig. 5.6.12 shows the all three method of CSMA/CA

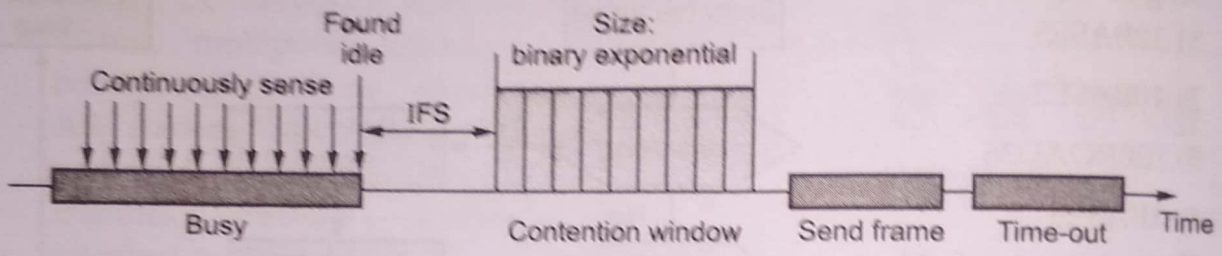


Fig. 5.6.12 CSMA/CA methods

Inter-frame space

- Collisions are avoided by deferring transmission even if the channel is found idle.
- When an idle channel is found, the station does not send immediately. It waits for a period of time called the Inter-Frame Space (IFS).
- In CSMA/CA, the IFS can also be used to define the priority of a station of a frame. A station that is assigned shorter IFS has a higher priority.

Contention window

- Contention windows are an amount of time divided into slots. A station that is ready to send chooses a random number of slots as its wait time.
- Station set one slot for the first time and then double each time the station cannot detect an idle channel after the IFS time.
- In this method, the station needs to sense the channel after each time slot
- If the station finds the channel busy, it does not restart the process; it just stops the timer and restarts it when the channel is sensed as idle.
- This method gives the priority to the station with the longest waiting time.

Acknowledgments

- The data may be corrupted during the transmission. The positive acknowledgment and the time out can help guarantee that the receiver has received the frame.
- Fig. 5.6.13 shows the flowchart for CSMA/CA.

Scanned with CamScanner

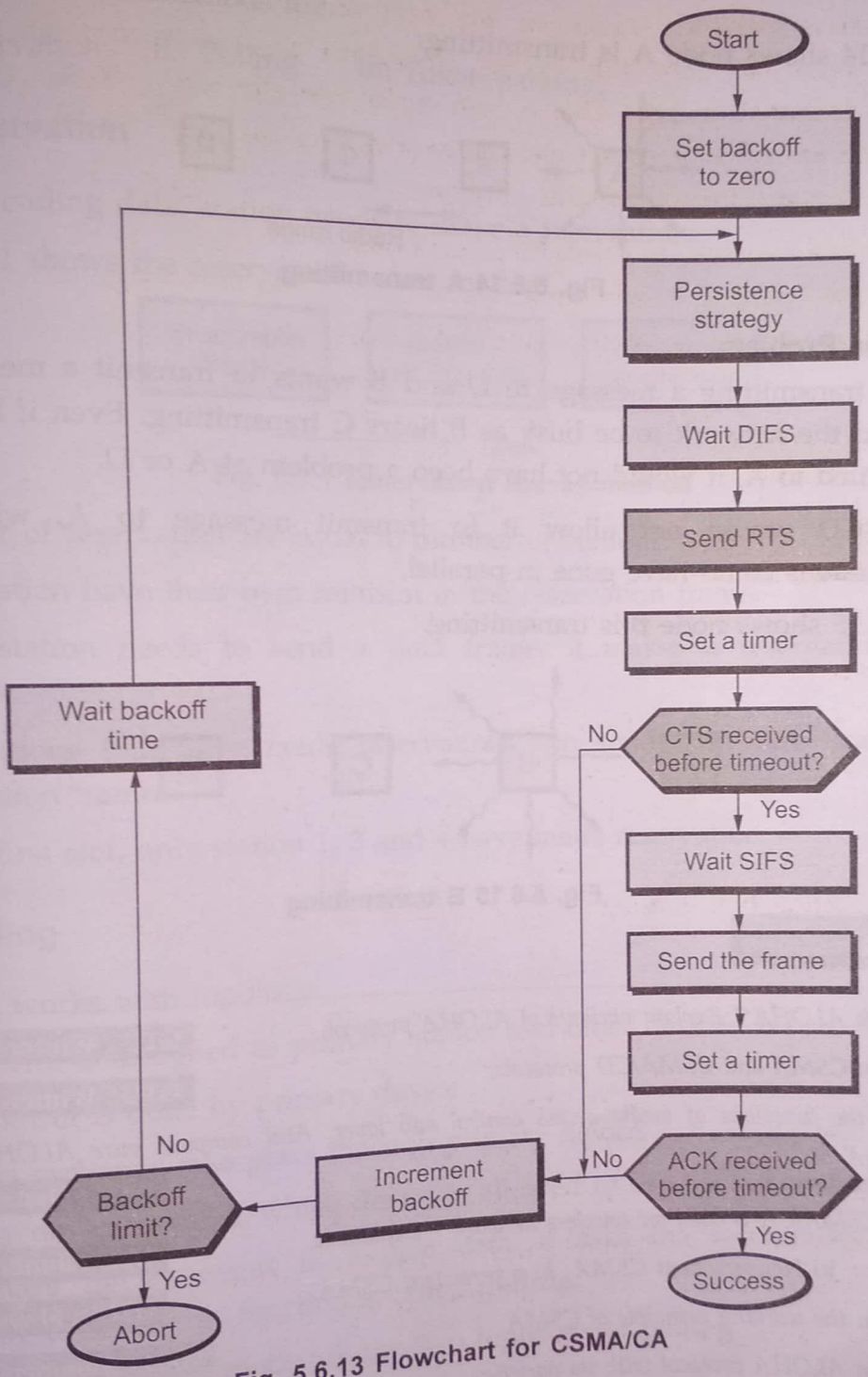


Fig. 5.6.13 Flowchart for CSMA/CA

Hidden Node Problem

- In the case of wireless network it is possible that A is sending a message to B, but C is out of its range and hence while "listening" on the network it will find the network to be free and might try to send packets to B at the same time as A. So, there will be a collision at B.
- The problem can be looked upon as if A and C are hidden from each other. Hence it is called the "hidden node problem".

Scanned with CamScanner

- Fig. 5.6.14 shows node A is transmitting.

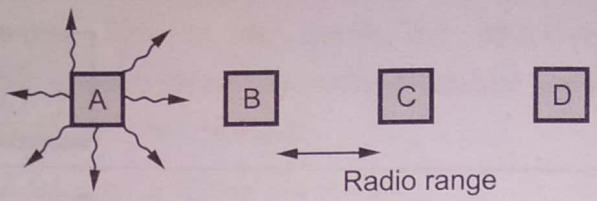


Fig. 5.6.14 A transmitting

Exposed Node Problem

- If C is transmitting a message to D and B wants to transmit a message to A, B will find the network to be busy as B hears C transmitting. Even if B would have transmitted to A, it would not have been a problem at A or D.
- CSMA/CD would not allow it to transmit message to A, while the two transmissions could have gone in parallel.
- Fig. 5.6.15 shows node B is transmitting.

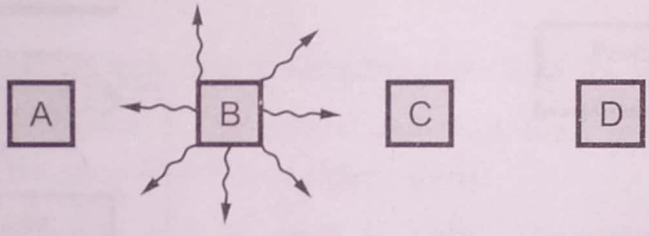


Fig. 5.6.15 B transmitting

University Questions

1. *What is ALOHA ? Explain variants of ALOHA protocol.* **GTU : Winter-14, Marks 7**
2. *Explain CSMA and CSMA/CD protocols.* **GTU : Winter-14, Marks 7**
3. *Write the functions of media access control sub layer. And compare pure ALOHA and slotted ALOHA.* **GTU : Dec.-10, Marks 7**
4. *Why CSMA/CA is used for wireless network ? Explain it.* **GTU : June-11, Marks 7**
5. *Define : a) Nonpersistent CSMA b) p-persistent CSMA.* **GTU : May-12, Marks 3**
6. *Explain the working principle of CSMA.* **GTU : Summer-13, Marks 7**
7. *Explain ALOHA protocol with its varieties.* **GTU : Winter-13, Marks 7**
8. *Explain : ALOHA* **GTU : Summer-14, Marks 4**
9. *What is channel allocation ? How CSMA helps to solve problem?* **GTU : Summer-14, Marks 7**

5.7 Controlled Access

1. In this, the stations consult one another to find which station has the right to send.
2. A station cannot send unless it has been authorized by other stations.

Scanned with CamScanner

3. Controlled access methods are :
   i. Reservation ii. Polling iii. Token passing.

5.7.1 Reservation

1. Before sending data, station needs to make a reservation.
   Fig. 5.7.1 shows the reservation access method.

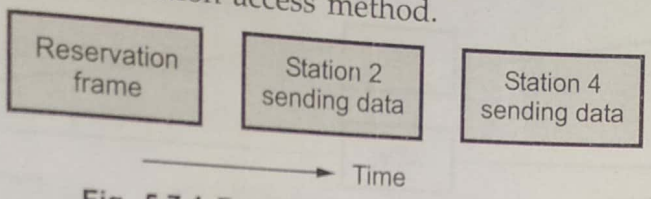


Fig. 5.7.1 Reservation access method

2. Number of reservation are equal to number of stations.
3. Each station have their own minislot in the reservation frame.
4. When station needs to send a data frame, it makes a reservation in its own minislot.
5. The stations that have made reservations can send their data frames after the reservation frame.
6. In the first slot, only station 1, 3 and 4 have made reservation.

5.7.2 Polling

1. Polling works with topology.
2. One device is designed as primary station and other devices are secondary station.
3. Link control is done by primary device.
4. All data exchange take place through primary device.
5. Primary device decides, which device is allowed to use the channel at a given time.
6. If primary device wants to receive data, it asks the secondaries if they have anything to send, this function is called **polling.**
7. Select mode and poll mode are the two functions of polling.
8. In polling, primary device receive the data.
9. In select mode, primary device sends data to secondary device.
   Fig. 5.7.2 shows the select mode.
10. Link is available if primary device is not sending or receiving any data.
11. Before sending data, the primary creates and transmits a select (SEL) frame.
12. SEL frame includes address of the intended secondary device.

Scanned with CamScanner

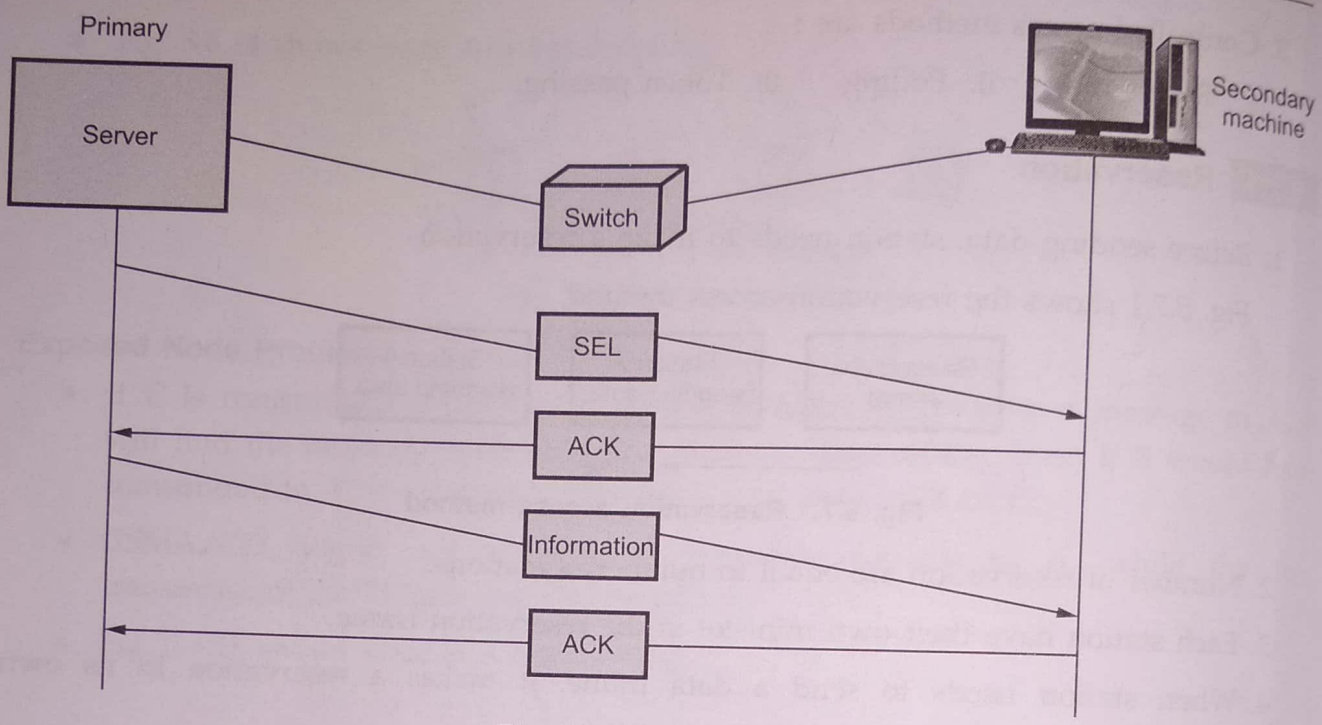


Fig. 5.7.2 Select mode

Fig. 5.7.3 shows the poll method.

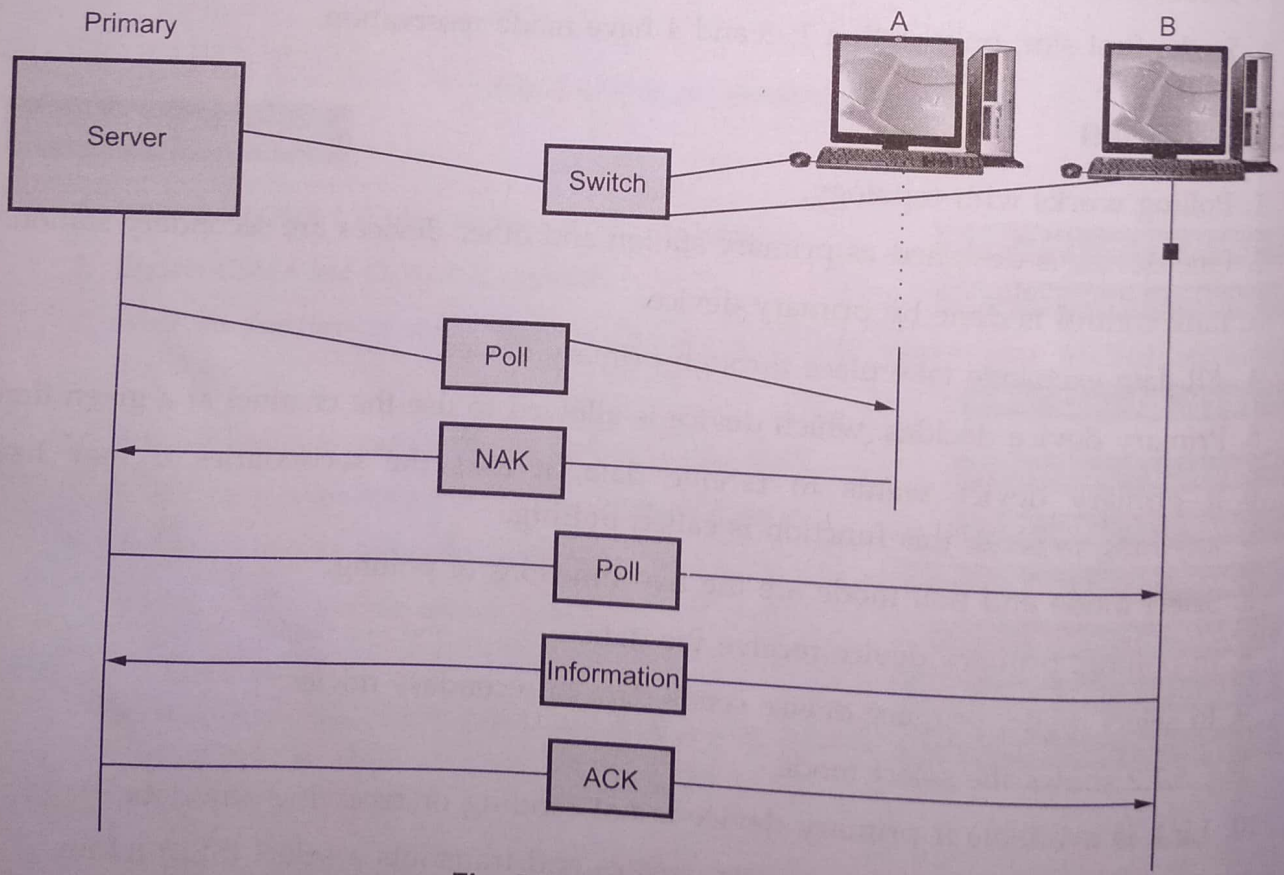


Fig. 5.7.3 Polling method

Scanned with CamScanner

5.7.3 Token Passing

1. A station is allowed to send data when it receives a token (special frame).
2. Ring topology is used for connecting devices.
3. Each station has a predecessor and a successor.
4. Frames are coming from predecessor and going to the successor.
5. Token is circulates around the ring.
6. The station captures the token if they want to send data.

Fig. 5.7.4 shows token passing network.

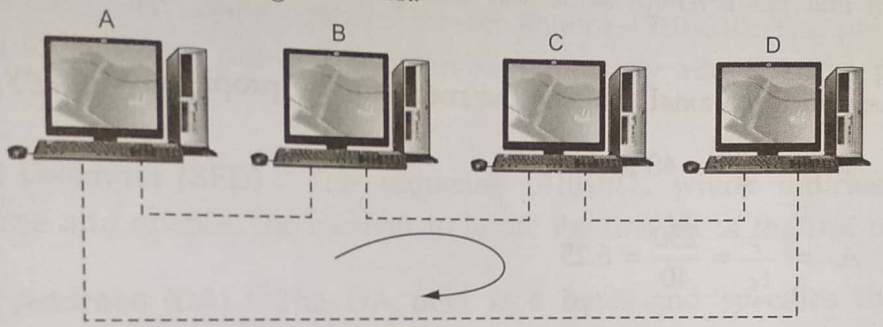


Fig. 5.7.4 Token passing network

7. Flowchart for token passing procedure is shown in Fig. 5.7.5.

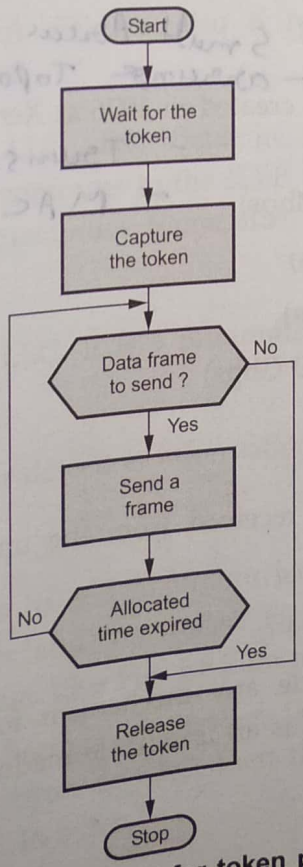


Fig. 5.7.5 Flowchart for token passing

Scanned with CamScanner

Example 5.7.1 *A CSMA/CD bus spans a distance of 1.5 km. If the data rate is 5 Mbps what is the minimum frame size ?*

Solution : Typical propagation speed in LAN cables = 200 m/μs

End to end propagation delay t_p,

$$t_p = \frac{1500}{200} = 7.5 \text{ μs}$$

Minimum frame size = $2 \times 7.5 \times 10^{-6} \times 5 \times 10^{6} = 75$ bits

Example 5.7.2 *Compute the maximum channel utilization for a MAN which uses CSMA mechanism and has a length of 50 km, and operates at 50 Mbps with a frame length of 2000 bits.*

Solution : Assuming co-axial cable as the medium, the propagation delay is 5 μs/km

$$\therefore \quad t_f = \frac{2000}{5} = 40 \text{ μs}$$

$$A = \frac{t_p}{t_f} = \frac{250}{40} = 6.25$$

$$U_{max} = \frac{1}{1+A} = 0.14$$

5.8 IEEE Standard 802.3

GTU : May-12

- The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center.
- Generations of Ethernet
  a) Standard Ethernet (10 Mbps)
  b) Fast Ethernet (100 Mbps)
  c) Gigabit Ethernet (1 Gbps)
  d) Ten-Gigabit Ethernet (10 Gbps)

5.8.1 MAC Sublayer

MAC sublayer frames data received from the upper layer and passes them to the physical layer.

5.8.1.1 Frame Format

- Ethernet does not provide any mechanism for acknowledging received frames, making it what is known as an unreliable medium.

Scanned with CamScanner

- The frame format of the MAC is shown in Fig. 5.8.1

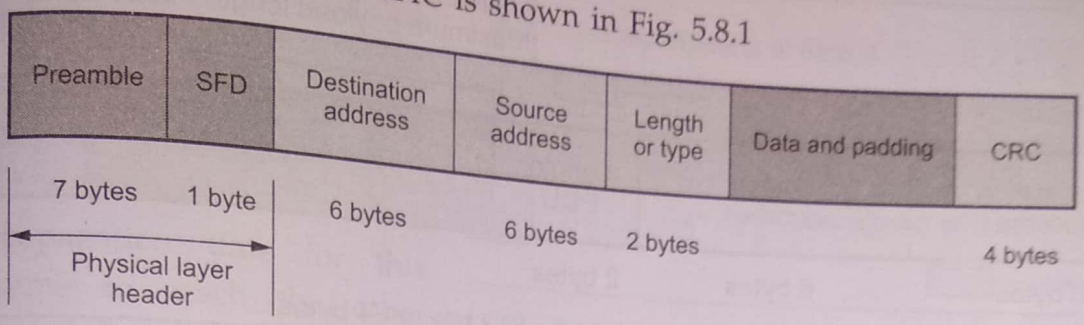


Fig. 5.8.1 802.3 Frame format

1. Preamble : A 7-byte pattern of alternating 0s and 1s used by the receiver to establish bit synchronization. Each frame contains the bit pattern 10101010. The pattern provides only an alert and a timing pulse. The preamble is actually added at the physical layer and is not part of the frame.

2. Start Frame Delimiter (SFD) : The sequence 10101011, which indicates the actual start of the frame and enables the receiver to locate the first bit of the rest of the frame.

3. Destination Address (DA) : The DA field is 6 bytes and specifies the station for which the frame is intended. It may be a unique physical address, a group address or a global address.

4. Source Address (SA) : The SA field is also 6 bytes and contains the physical address of the sender of the packet.

5. Length or Type : Length of LLC data field in octets, or Ethernet Type field, depending on whether the frame conforms to the IEEE 802.3 standard or earlier Ethernet specification. In either case, the maximum frame size, excluding preamble and SFD, is 1518 bytes.

6. Data : Data unit supplied by LLC. It is a minimum of 46 bytes and a maximum of 1500 bytes.

7. CRC : This field contains error detection information.

5.8.1.2 Frame Length

- An Ethernet frame needs to have a minimum length of 512 bits or 64 bytes. Part of this length is the header and the trailer. Fig. 5.8.2 shows the minimum and maximum length of the frame. (See Fig. 5.8.2 on next page)
- If we count 18 bytes of header and trailer i.e. 6 bytes of SA + 6 bytes of DA + 2 bytes of length + 4 bytes of CRC, then the minimum length of data from the upper layer is 64 – 18 = 46 bytes.

Scanned with CamScanner

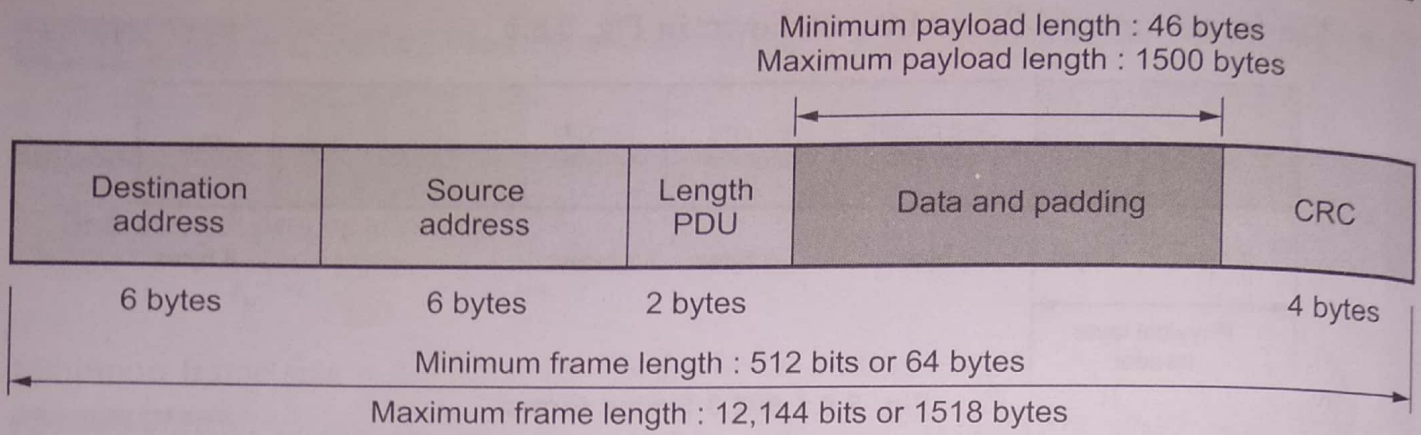


Fig. 5.8.2 Minimum and maximum lengths

- If the upper layer packet is less than 46 bytes, padding is added to make up the difference.
- The standard defines the maximum length of a frame (without preamble and SFD) as 1518 bytes. If we subtract the 18 bytes of header and trailer, the maximum length of the payload is 1500 bytes.

Addresses

- A source address is always a unicast address, i.e. the frame comes from only one station. The destination address can be unicast, multicast or broadcast.
- If the least significant bit of the first byte in a destination address is 0, the address is unicast; otherwise, it is multicast.

Access Method : CSMA/CD

- Standard Ethernet uses 1-persistent CSMA/CD.
  1. Slot time = Round trip time + Time required to send the jam sequence
  2. Maximum length = Propagation speed $\times \frac{\text{Slot time}}{2}$

Minimum frame size

- While a data field of 0 bytes is sometimes useful, it causes a problem. When a transceiver detects a collision, it truncates the current frame, which means that stray bits and pieces of frames appear on the cable all the time.
- Ethernet requires that valid frames must be atleast 64 bytes long, from destination address to checksum, including both.
- Reason for having a minimum length frame is to prevent a station from completing the transmission of a short frame before the first bit has even reached the far end of the cable, where it may collide with another frame.

Scanned with CamScanner

- Fig. 5.8.3 shows collision detection.
- At time 0, station A, at one end of the network sends off a frame. Let us call the propagation time for this frame to reach the other end T. Just before the frame gets to the other end (i.e. at time T – ε), the most distant station B, starts transmitting.
- When B detects that it is receiving more power than it is putting out, it knows that a collision has occurred, so it abort its transmission and generates a 48-bit noise burst to warn all other stations. In other words, it jams the either to make sure the sender does not miss the collision.
- Loss of bandwidth is 936×10^3 bytes/sec because we only transmit 64 bytes. The available bandwidth is 100 Mbps.

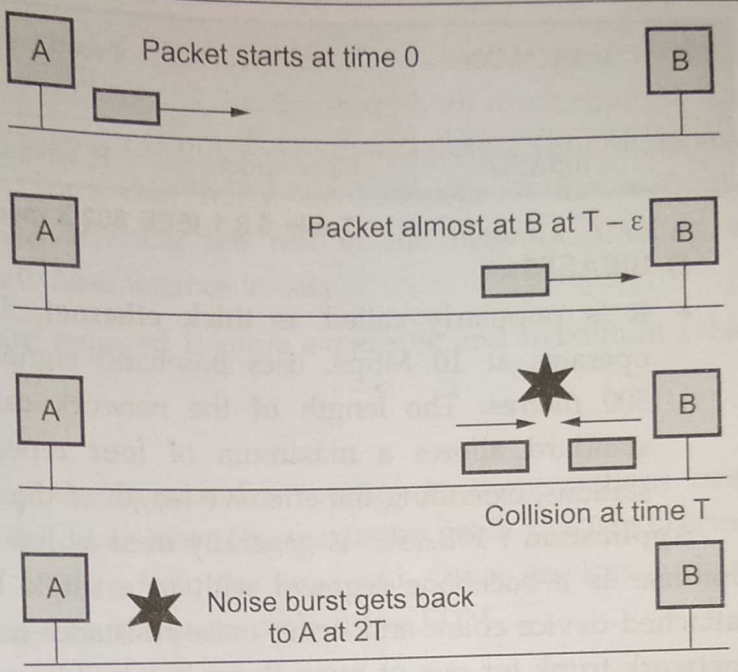


Fig. 5.8.3 Collision detection can takes as long as 2T

5.8.1.3 Ethernet Specifications

- CSMA/CD offers various options in terms of transmission medium, signalling technique, data rate and maximum electrical cable segment length.
- Table 5.8.1 summarizes these options defined for the IEEE 802.3 medium.

Sr. No.	Medium options	Transmission medium	Signaling technique	Data rate (Mbps)	Maximum segment length (m)
1.	10BASE5	Coaxial cable (50 ohm)	Baseband (Manchester)	10	500
2.	10BASE2	Coaxial cable (50 ohm)	Baseband (Manchester)	10	185
3.	1BASE5	Unshielded twisted pair	Baseband (Manchester)	1	250
4.	10BASET	Unshielded twisted pair	Baseband (Manchester)	10	100

Scanned with CamScanner

5.	10BROAD36	Co-axial cable (75 ohm)	Broad band (DPSK)	10	3600
6.	10BASEF	Fiber optics	Baseband	10	2000

Table 5.8.1 IEEE 802.3 medium options

1) 10BASE5 :

- It is popularly called as **thick ethernet**. The notation 10BASE5 means that it operates at 10 Mbps, uses baseband signaling and can support segment upto 500 metres. The length of the network can be extended using repeaters. The standard allows a maximum of four repeaters in the path between any two stations, extending the effective length of the network to 2.5 km.

Application : 10BASE5 is generally used as low cost alternative for fiber optic media for use as a backbone segment with in a single building. Its extended length, higher attached device count and better noise resistance make 10BASE5 well suited for use as a network trunk for one or more floors in a building. However the high cost of connecting each device makes 10BASE5 too expensive for most LAN installations a single break or bad connection in the cable can bring the entire network down.

2) 10BASE2 :

- It is popularly called as **chepearnet** or thin ethernet. It uses thin co-axial cable. The thinner cable results in significantly cheaper cost, at the penalty of fewer stations and shorter length. Therefore 10BASE2 is limited to a maximum of 30 network devices per unrepeated network segment with a minimum distance of 0.5 m. And segment length is reduced to 185 metres.

Application : For small budget conscious installations, 10BASE2 is the most economical topology such as UNIX work stations.

The disadvantages of 10BASE2 is that any break in the cable or poor connection will bring the entire network down and repeaters are required if more than 30 devices are connected to the network or the cable length exceeds 185 m.

3) 1BASE5 :

- It is also known as **star LAN**. It specifies operation at 1 Mbps, using a passive star topology.

Application : This options is substantially lower in cost than either of coaxial cable options. This options could be appropriate for a departmental-level LAN.

4) 10BASET :

- 10BASET is 10 MHz ethernet running over UTP cable. It also uses passive star topology. The maximum cable segment allowed is 100 - 150 metres. There is no minimum distance requirements between devices, such devices cannot be connected serially but in star wired. Maximum 1024 stations can be connected to network.

Scanned with CamScanner

Application : 10BASET is the most flexible topology for LAN's and is generally the best choice for network installations. 10BASET hubs or multi-hub concentraters, are typically installed in a central locations to the user community. The signalling technology is very reliable even in somewhat noisy environments it automatically shutdown the offending parts without affecting the rest of the network. Cabling is cheaper and requires less skill to install. Maintenance is easy.

The disadvantages are the hardware required is more expensive and maximum cable run from hub is 100-150 metre.

5) 10BROAD36 :

- It is a 10 Mbps broadband option. It provides support to more stations over greater distances than the baseband versions. The maximum cable run is restricted to 3600 m in two segments of 1800 m from the head end. Other services such as TV or voice can also be integrated on the same cable using FDM.

6) 10BASEF :

- 10BASEF is 10 Mbps running over fiber optic cabling. The maximum cable length depends on signaling technology and medium used but can go upto 2 km unrepeated segment. It is star wired so there is no minimum distance requirement between devices.

Application : 10BASEF is the only recommended topologies for inter-building links. However they need not be limited to this role, it can also run to desktop. It has excellent noise immunity.

The disadvantage is, it is very expensive due to the cost of connectors and terminators.

5.8.1.4 Manchester Encoding

- In order to transport digital bits of data across carrier waves, encoding techniques have been developed each with their own merits and demerits.
- Digital signal is a sequence of discrete, discontinuous voltage pulses. Data is represented in binary. These binary data is transmitted by encoding each data bit into signal elements.

Desirable Properties of Encoding Techniques

1. **Synchronization capability :** The ability to stay synchronized or to get re-synchronized.
2. Error detection capability.
3. **Immunity to noise :** Ability to separate noise from the transmitted signal.

Scanned with CamScanner

1) Manchester Encoding

- In Manchester encoding each bit period has both the high and low voltage values. If the data is a 1, the first half of the bit time period is sent at the positive level, and the second half of the period is at the negative level. For data bit of 0, first a negative signal and then a positive signal. There is a transition which can be used for a synchronization. Sometimes this method is called self clocking encoding method. Fig. 5.8.4 shows Manchester encoding waveform for the 8-bit data stream 11000101.

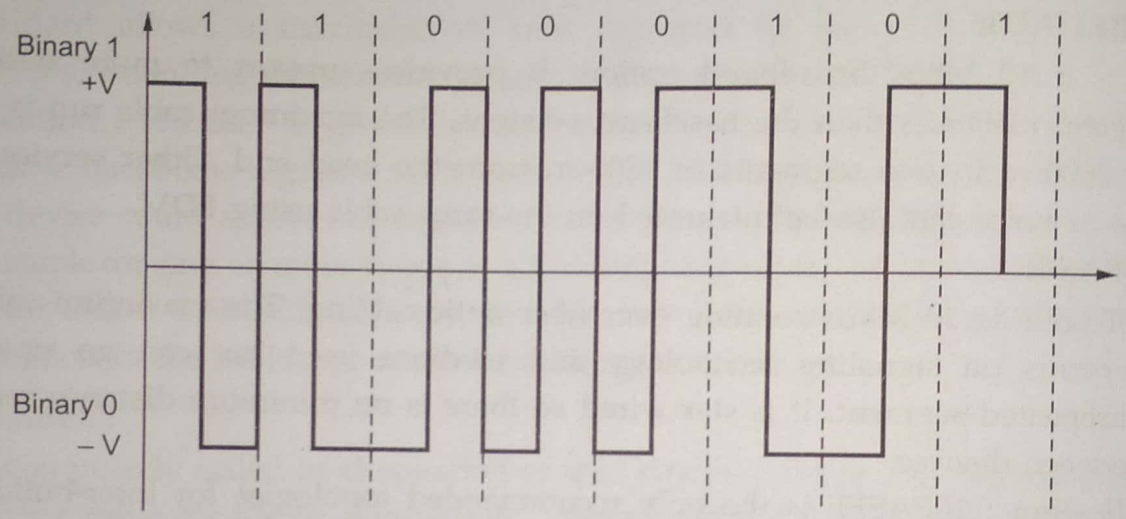


Fig. 5.8.4 Manchester encoding

2) Differential Manchester

- In differential Manchester encoding, a binary 0 is marked by a transition at the beginning of an interval, whereas a 1 is marked by the absence of a transition. In this encoding method, detecting changes is often more reliable, especially when there is a noise in the channel. Fig. 5.8.5 shows the differential Manchester encoding for 8-bit data stream 101011100.

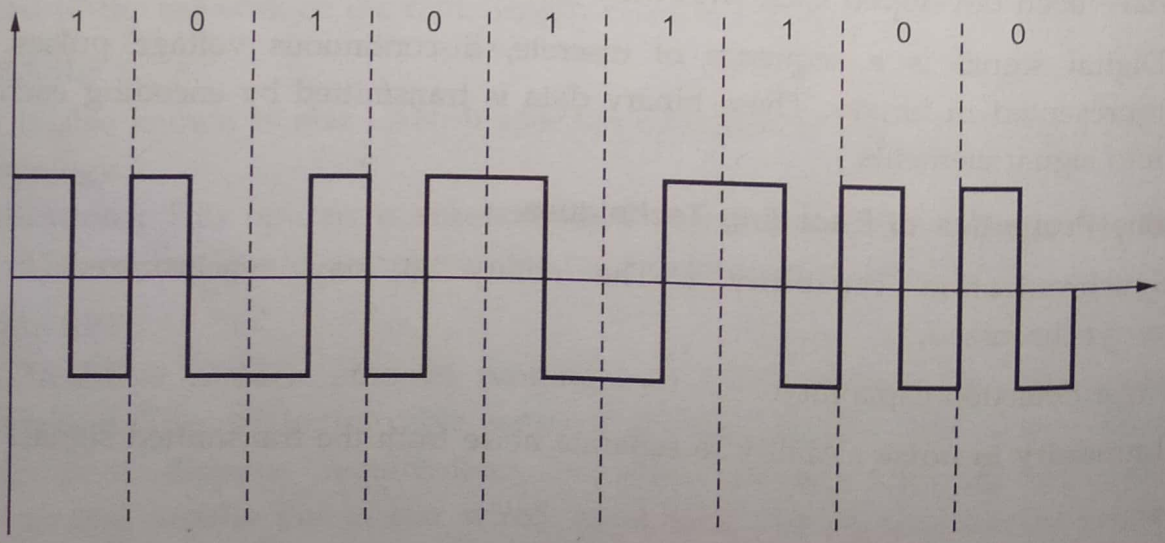


Fig. 5.8.5 Differential Manchester

Scanned with CamScanner

5.8.1.5 Binary Exponential Backoff Algorithm

- After a collision, time is divided into discrete slots whose length is equal to the worst-case round-trip propagation time on the either (2τ).
- **After first collision :** Each station was either 0 or 1 slot times before trying again. If two station collide and each one picks the same random number, they will collide again.
- **After second collision :** Each one picks either 0, 1, 2 or 3 at random and waits that number of slot times.
- **After third collision :** If a third collision occurs, then the next the number of slots to wait is chosen at random from the interval 0 to $2^3 - 1$.
- **After i^{th} collision :** A random number between 0 and $2^e - 1$ is chosen, and that number of slots is skipped.
- This algorithm, called **binary exponential backoff** was chosen to dynamically adapt to the number of stations trying to send.

5.8.1.6 Ethernet Performance

$$\text{Channel efficiency} = \frac{P}{P + 2\tau / A} \quad \ldots (5.8.1)$$

Rewrite the above formula in terms of the frame length (F), the network bandwidth (B), the cable length (L) and the speed of signal propagation (c) for the optimal case of e contention slots per frame. With P = F/B, then equation (5.8.1) becomes

$$\text{Channel efficiency} = \frac{1}{1 + 2\,B\,Le / cF} \quad \ldots (5.8.2)$$

University Question — **GTU : May-12, Marks 3**

1. *Explain switched ethernet.*

5.9 Bridged Ethernet

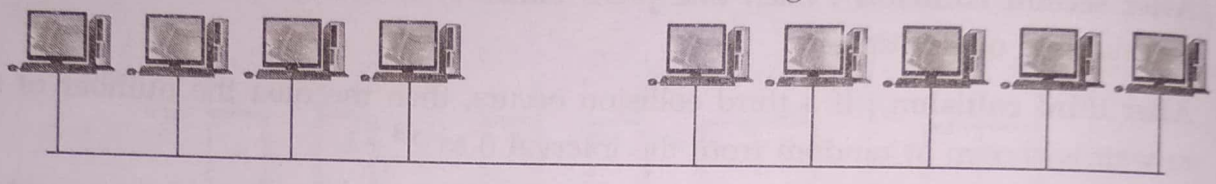
- Bridges have two effects on an Ethernet LAN.
  1. Raising the bandwidth
  2. Separating collision domains

Ethernet as IEEE802.3 standard
Support up to 100 MBPS
ethernet to more speed

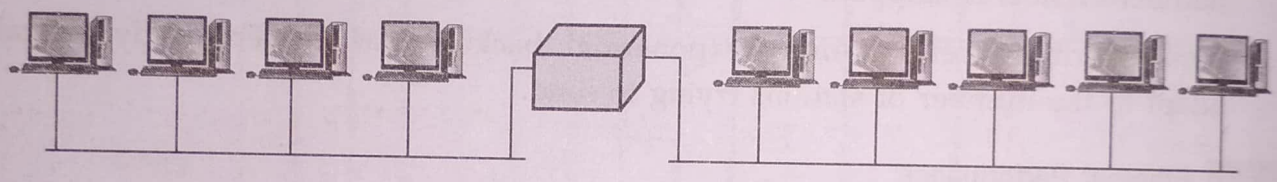
Scanned with CamScanner

1. Raising the bandwidth

- Without using the bridge in the ethernet network, the total capacity is shared among all stations with a frame to send; the stations share the bandwidth of the network.
- If only one station has frames to send, it benefits from the total capacity. But if more than one station needs to use the network, the capacity is shared.
- Fig. 5.9.1 shows a network with and without bridge.



(a) Without bridge



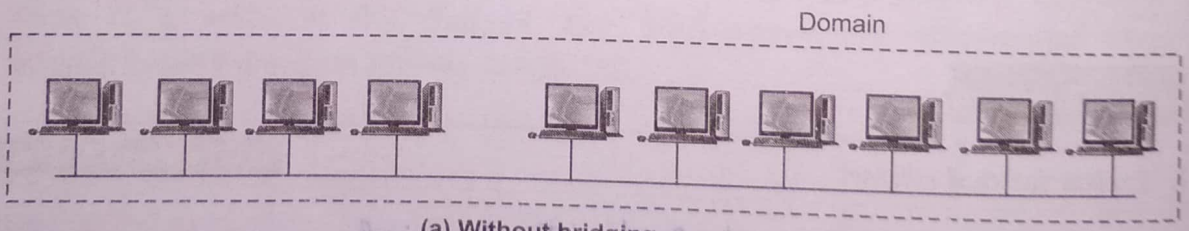
(b) With bridge

Fig. 5.9.1 Network with and without bridge

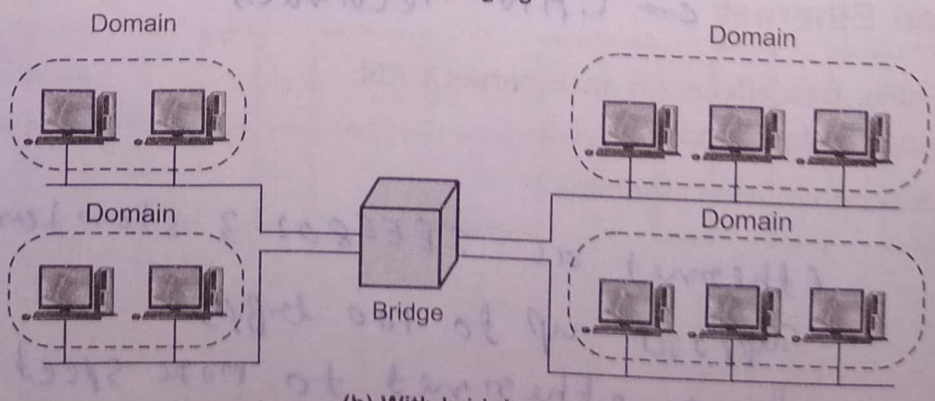
- A bridge divides the network into two or more networks. Bandwidth-wise, each network is independent.

2. Separating collision domains

- Bridge separates the collision domain. Fig. 5.9.2 shows the collision domains for an unbridged and a bridged network.



(a) Without bridging



(b) With bridging

Fig. 5.9.2 Collision domain

Scanned with CamScanner

- The collision domain becomes much smaller and the probability of collision is reduced tremendously.

Switched Ethernet

- The idea of a bridged LAN can be extended to a switched LAN. Fig. 5.9.3 shows the switched ethernet.
- A layer-2 switch is an N-port bridge with additional sophistication that allows faster handling of the packets.

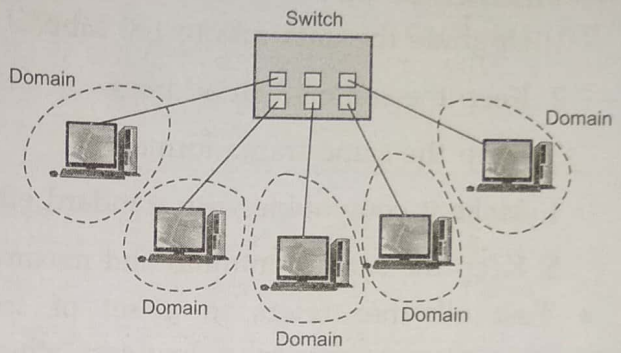


Fig. 5.9.3 Switched ethernet

Full Duplex Ethernet

- 10Base5 and 10Base2 supports the half duplex communication. A station can either send or receive, but may not do both at the same time.
- Fig. 5.9.4 shows full duplex switched Ethernet.

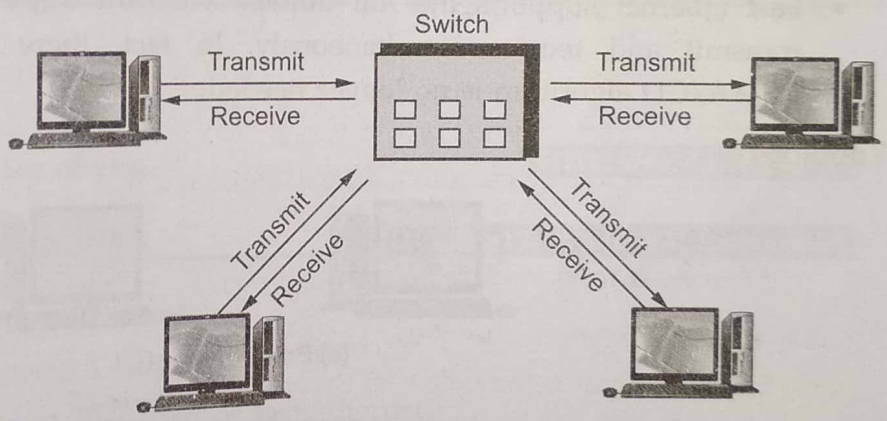


Fig. 5.9.4 Full duplex switched ethernet

- The full duplex mode increases the capacity of each domain from 10 to 20 Mbps. Full duplex ethernet uses two lines : one to transmit and one to receive.
- There is no need of CSMA/CD method. In full duplex ethernet, each station is connected to the switch via two separate links. Each station or switch can send and receive independently without worrying about collision.
- Each link is a point to point dedicated path between the station and the switch. The carrier sensing and collision detection functionalities of the MAC sublayer can be turned off.

MAC Control Layer

There is no explicit flow control or error control to inform the sender that the frame has arrived at the destination without error. When the receiver receives the frame, it does not send any positive or negative acknowledgment.

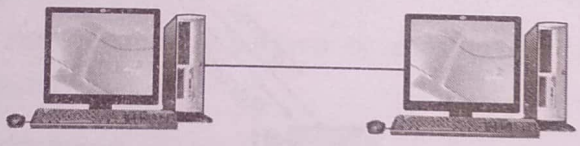
Scanned with CamScanner

5.10 Fast Ethernet

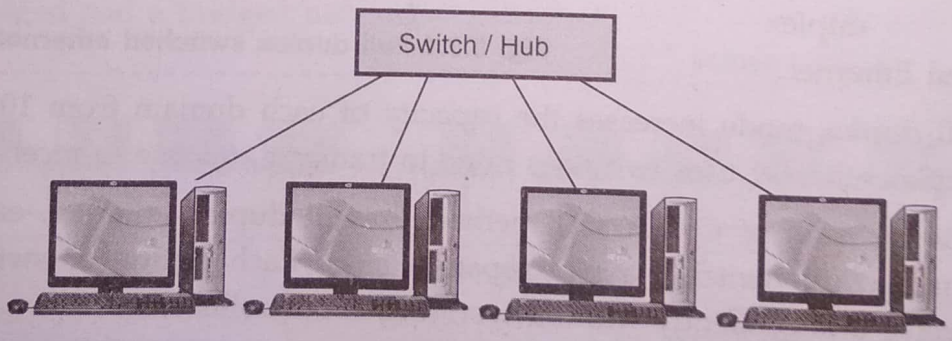
GTU : Dec.-11

- Fast ethernet is backward compatible with standard ethernet. The goals of fast ethernet can be :
  1. Upgrade the data rate to 100 Mbps.
  2. Keep the same 48-bit address.
  3. Keep the same frame format.
  4. Make it compatible with standard ethernet.
  5. Keep the same minimum and maximum frame length.
- Fast ethernet refers to a set of specifications developed by the IEEE 802.3 committee to provide a low cost, ethernet compatible LAN operating at 100 Mbps. A traditional ethernet is half duplex : A station can either transmit or receive a frame, but it cannot do both simultaneously.
- Fast ethernet supports the full duplex with full duplex operation, a station can transmit and receive simultaneously. In fact, there is no collisions and the CSMA/CD algorithm is no longer needed.

Topology



(a) Point-to-point



(b) Star

Fig. 5.10.1 Fast ethernet topology

- Fast ethernet is designed to connect two or more stations together. If there are only two stations, they can be connected point-to-point. Three or more stations need to be connected in a star topology with a hub or a switch at the center. It is shown in the Fig. 5.10.1.

Scanned with CamScanner

Summary sheet of fast ethernet

Parameters	100BASE-TX		100BASE-FX	100BASE-T4
Transmission medium	STP	Cat 5 UTP	Fiber	Cat 3, 4, 5 UTP
Number of wires	4	4	2	4
Data rate	100 Mbps	100 Mbps	100 Mbps	100 Mbps
Maximum segment length	100 m	100 m	100 m	100 m
Network span	200 m	200 m	400 m	200 m
Line coding	MLT-3	MLT-3	4B5B	8B/6T/NRZ

University Question

1. *Compare : Ethernet and fast ethernet.* **GTU : Dec.-11, Marks 5**

5.11 Gigabit Ethernet

GTU : May-12, Winter-13, Summer-14

- Goals of gigabit ethernet
  1. Upgrade the data rate to 1 Gbps.
  2. Make it compatible with standard or fast ethernet.
  3. Use the same 48-bit address.
  4. Use the same frame format.
  5. Keep the same minimum and maximum frame lengths.
- It support the two different modes of operations.
  i) Full duplex ii) Half duplex.
- In full duplex mode, there is a central switch connected to all computers or other switches. Each switch has buffers for each input port in which data are stored until they are transmitted. There is no collisions in this mode. This means that CSMA/CD is not used.
- Gigabit ethernet can also be used in half duplex mode. In this case, a switch can be replaced by a hub, which acts as the common cable in which a collision might occur. The half duplex approach uses CSMA/CD. For shared medium hub operation, there are two enhancements to the basic CSMA/CD scheme.

Extension : It defines the minimum length of a frame as 512 bytes.

Scanned with CamScanner

2. **Frame bursting :** It allows for multiple short frames to be transmitted consecutively, up to a limit, without relinquishing control for CSMA/CD between frames.

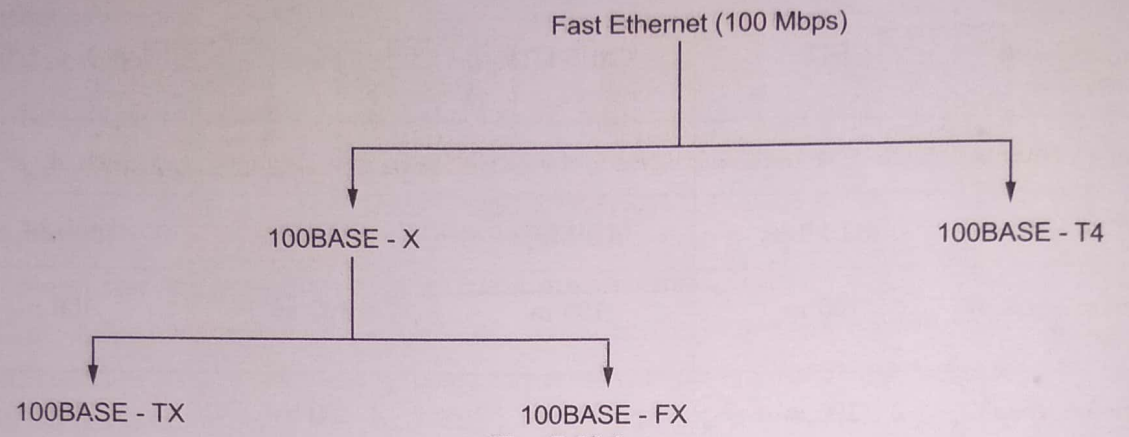


Fig. 5.11.1

Transmission Media

Summary sheet of gigabit ethernet

Parameters	1000Base-SX	100Base-LX	100Base-CX	1000Base-T
Transmission medium	Fiber short wave	Fiber long wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum segment length	550 m	500 m	25 m	100 m
Line coding	NRZ	NRZ	NRZ	4D-PAM5

Topology

- Gigabit ethernet is designed to connect two or more stations. If there are only two stations, thsey can be connected point-to-point. Fig. 5.11.2 shows the point-to-point connection.

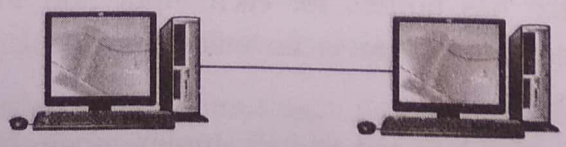


Fig. 5.11.2 Point-to-point

- Three or more stations need to be connected in a star topology with a hub or a switch at the center. This is shown in Fig. 5.11.3.

Scanned with CamScanner

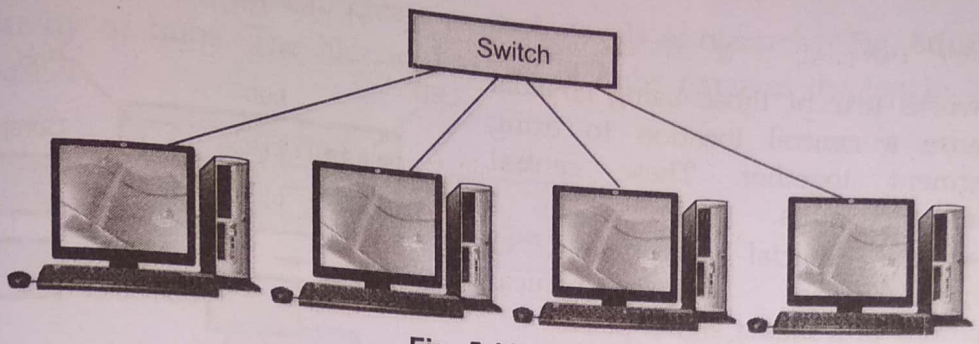


Fig. 5.11.3 Star

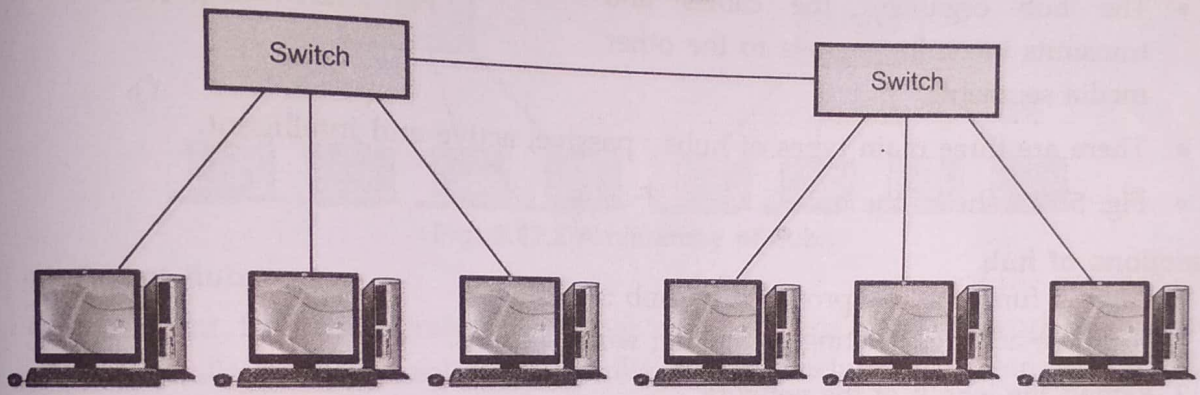


Fig. 5.11.4 Two stars

University Questions

1. *Which two different modes of operation are supported by gigabit ethernet ?* **GTU : May-12, Marks 4**
2. *Explain how congestion control is achieved in TCP ?* **GTU : Winter-13, Marks 7**
3. *What is ethernet ? Explain fast ethernet and gigabit ethernet.* **GTU : Summer-14, Marks 7**

5.12 Switching and Bridging

GTU : Dec.-10, 11, June-11, May-12, Summer-13,14, Winter-14

- Connecting devices are divided into five different types based on the layer in which they operate in a network.
- The device which operates below the physical layer such as passive hub.
- A repeater or an active hub operates at the physical layer.
- A bridge or a two layer switch operates at the physical and data link layers.
- A router or layer three switch operates at the physical, data link and network layers.
- Those which can operate at all five layers i.e. a gateway.

Scanned with CamScanner

5.12.1 Hubs

All networks (except those using co-axial cable) require a central location to bring media segment together. These central locations are called **hubs**.

- Hubs are special repeaters that overcome the electromechanical limitations of a media signal path.
- The hub organizes the cables and transmits incoming signals to the other media segments.

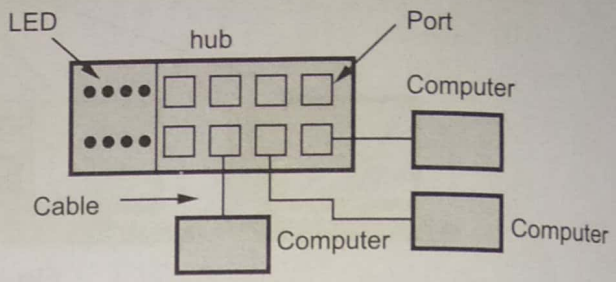


Fig. 5.12.1 Hub (8 port)

- There are three main types of hubs : passive, active and intelligent.
- Fig. 5.12.1 shows the hub.

Functions of hub

Following functions are provided by hub :

1. Facilitate adding, deleting or moving work stations.
2. Extend the length of the network.
3. Provide flexibility by supporting multiple interfaces. For example, ethernet, token ring, FDDI.
4. It offers fault tolerance feature.
5. Provide centralize management services.

1) Passive hubs

- A passive hub is just a connector. A passive hub simply combines the signals of network segments. There is no signal regeneration. This type of a hub is a part of the transmission media; its location in the Internet model is below the physical layer.
- The hub is the collision point. A passive hub reduces by half the maximum cabling distances permitted. With passive hub, each computer receives the signal sent from all the other computers connected the hub.

2) Active hub

- An active hub is actually a multiport repeater. An active hub is that regenerates or amplifies the signals.
- By using active hubs the distance between devices can be increased. It is normally used to create connections between stations in a physical star topology. An active hub is expensive than passive hub.
- One of the disadvantages of an active hub is that they amplify noise along with the signal.

Scanned with CamScanner

- Hubs can also be used to create multiple levels of hierarchy. Fig. 5.12.2 shows the hierarchy of hubs. The hierarchical use of hubs removes the length limitation of 10 Base-T.

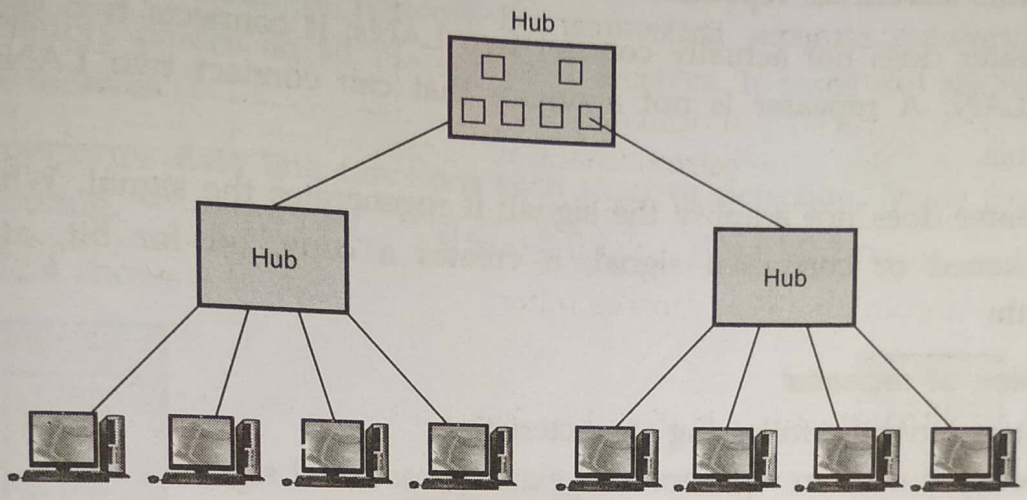


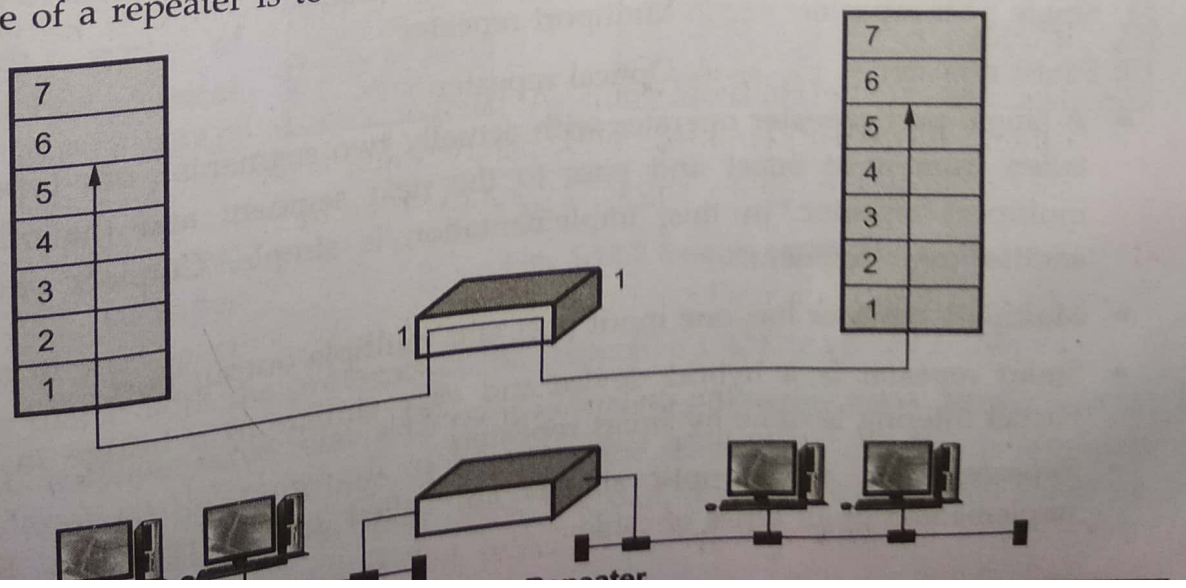
Fig. 5.12.2 A hierarchy of hubs

3) Intelligent hub

- Intelligent hub regenerates the signal and performs some network management and intelligent path selection. Intelligent hub includes switching hubs. Many switching hubs can choose that alternative path which, will be the quickest and send the signal that way.
- Advantages of this hub is all transmission media segment permanently connecting to hub because each segment will be used only when a signal is sent to a device using that segment.

5.12.2 Repeaters

- Repeater is an electronic device. It operates only in the physical layer. The basic purpose of a repeater is to extend the distance of LAN.



Scanned with CamScanner

- A repeater receives a signal and before it becomes too weak or corrupted, regenerates the original bit pattern. The repeater then sends the refreshed signal.
- Fig. 5.12.3 shows the repeater.
- A repeater does not actually connect two LANs; it connects two segments of the same LAN. A repeater is not a device that can connect two LANs of different protocols.
- A repeater does not amplify the signal; it regenerates the signal. When it receives a weakened or corrupted signal, it creates a copy, bit for bit, at the original strength.

Characteristics of repeater

- Repeaters have the following characteristics.

1. Repeaters are used to regenerate an existing baseband signal.
2. Repeater will pass a broadcast.
3. Repeater is used primarily in a co-axial bus topology.
4. Repeater operates at physical layer of OSI model.
5. Segments connected by a repeater must use the same media access control method.
6. Repeater does not filter packets.
7. Repeater can pass traffic between different types of media (e.g. coaxial to FOC) provided appropriate interfaces exist.
8. Repeater does not accelerate or change the signal. It simply regenerates it.
9. Segments connected by a repeater must have the same network address.

Types of repeaters are as follows :

1. Single port repeater
2. Multiport repeater
3. Smart repeater
4. Optical repeater

- A single port repeater operates with actually two segments : one type has a signal taken from it to boost and pass to the next segment and the other type is a multiport repeater. In this, implementation is simple. Connect one segment to another cable segment.
- Multiport repeater has one input port and multiple output port.
- Smart repeater is a hybrid device and very similar to a bridge in functionality. Packet filtering is done by smart repeaters.
- Repeaters that repeat optic signals are called optical repeaters. Repeaters are implemented in all types of cable.

Scanned with CamScanner

5.12.3 Bridges

- A bridge operates in both the physical and the data link layer. A bridge extends the maximum distance of network by connecting separate network segment. A bridge simply passes on all the signals it receives. It reads the address of all the signal it receives.
- Bridge performs data link functions such as error detection, frame formatting and frame routing. -Solve traffic problem
- Fig. 5.12.4 shows a bridge.

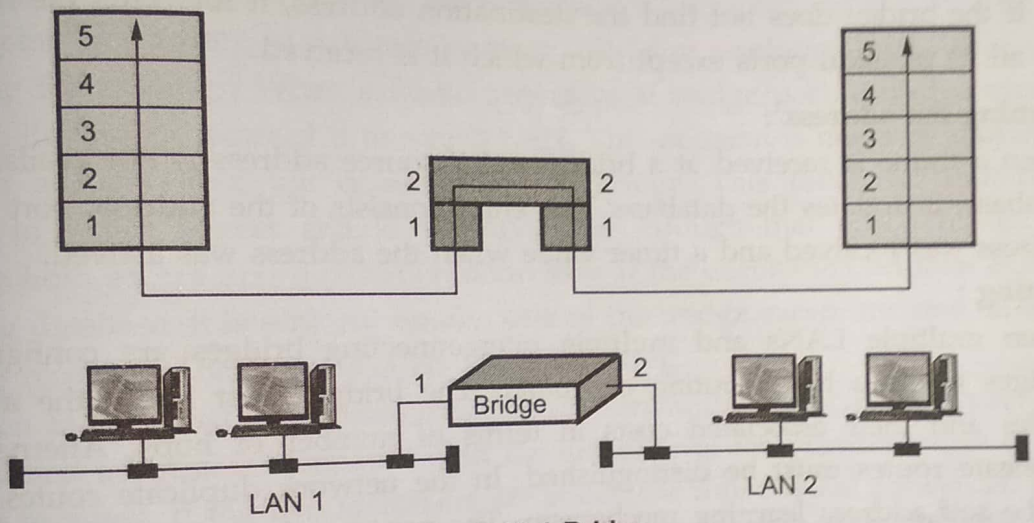


Fig. 5.12.4 Bridge

5.12.3.1 Bridge Architecture

- Fig. 5.12.5 shows the layered architecture of two-port bridge.
- At each port of bridge, it has physical layer and MAC sublayer.
- The physical layer and MAC sublayer protocols at each port of bridge match with the protocols of the respective LAN.

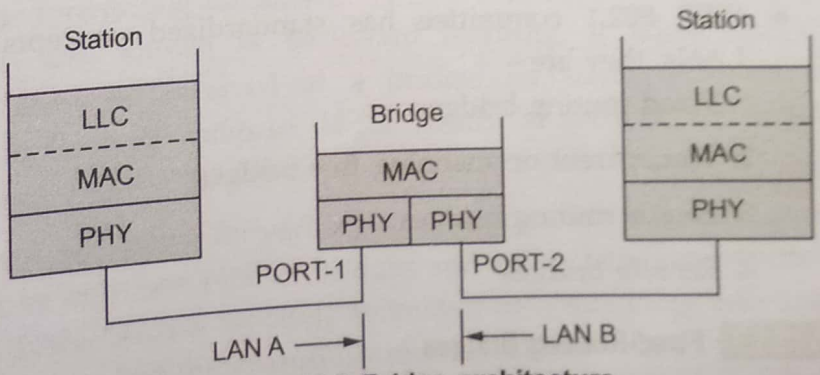


Fig. 5.12.5 Bridge architecture

- The MAC sublayer have relay and routing function between them. When a MAC frame is received by the bridge, it examines the destination address, it reformats the frame as required by the other LAN. The data fields of MAC frame are of no interest to a bridge.

Scanned with CamScanner

5.12.3.2 Functions of Bridge

- A bridge performs following basic functions.

1. Frame filtering and forwarding :

- When the bridge receives a frame at any of its ports, it takes any of one actions.
  i) If the destination address is available on the same port by which it is received, the bridge discards the frame.
  ii) If the destination address is on different physical port, it forwards the frame onto that port.
  iii) If the bridge does not find the destination address, it forwards the frame over all its physical ports except from which it is received.

2. Learning the address :

- When a frame is received at a bridge and if source address is not available in the database, it updates the database. This entry consists of the address, port on which address was received and a timer value when the address was arrived.

3. Routing :

- When multiple LANs and multiple interconnecting bridges are configured, the bridges need to have routing capability. The bridge must know the alternative routes and their associated costs in terms of number of hops. Alternative and duplicate routes must be distinguished. In the network duplicate routes interfere in the self address learning mechanism. The process of deciding which frames to forward and where is called **bridge routing.**

802 Bridges :

- IEEE 802.1 committee has standardized concepts for interconnecting 802-type LANs, they are -
  1. Fixed-routing bridges
  2. Transparent or spanning tree bridges
  3. Source routing bridges
  4. Remote bridges

5.12.3.3 Fixed-Routing Bridges

- Fixed-routing bridges maintain a routing table (routing directory) which contains the information of addresses of each stations and the LANs.
- Each bridge has a routing table for LAN to which it is connected.
- When a frame arrives on that LAN the bridge finds the destination address in the appropriate routing table.
- The table entry specifies to which LAN, the bridge should forward the frames. If alternate routes are available between two LANs then typically the route with

Scanned with CamScanner

least number of hops is selected. These type of bridges are called fixed-routing because the content of the table does not change.

- Fixed-routing is the simplest and most commonly adopted strategy. It is most suited for small LANs, which are relatively stable.

5.12.3.4 Transparent Bridges or Spanning Tree Bridges

- In transparent bridge mechanism, bridges automatically develop a routing table and update table in response to changing topology.
- The algorithm consists of three mechanisms :
  i) Frame forwarding ii) Address learning iii) Loop resolution.

i) Frame forwarding : When a frame arrives on a bridge port, a bridge must decide whether to discard or forward it to which LAN. This decision is made by looking up the destination address in a big database in the bridge. This database contains station addresses to which frames should be forwarded through that port. This information pertains to both source and destination addresses. If the destination address is not in the forwarding database, it is sent out on all ports of the bridge except the one on which the frame was received. This process is known as **flooding**.

- When the destination address exists in the forwarding database, the port identifier of stored address is compared with the identifier of the port on which the address was received. If the two identifiers are equal, the frame is not forwarded since it is addressed to a station on the same LAN in which it originated. When the port identifiers are different, the frame is forwarded to the bridge port associated with the address as listed in the forwarding database.

ii) Address learning : It is also known as **backward learning**, it takes care of destination address. When a frame is received at a bridge, its source address is compared with the addresses in forward database. If the source address is not found there, the bridge makes a new entry to the data base.

iii) Spanning tree algorithm : For frame forwarding and address learning processes to operate properly, there must be only one path of bridges and LANs between any two segments in the entire bridged LAN. Such a topology is known as a **spanning** tree and the methodology for setting it up is called the **spanning tree algorithm**.

- Before learning spanning algorithm, the following concepts are needed.

1) Root bridge : Each bridge has a unique identifier. The bridge with the lowest identifier is called the root bridge.

2) Root path cost : Each port of a bridge has an associated cost parameter which is the cost of transmitting a frame through the particular port. When a frame transverse a path through several bridges, the path cost is the sum of all the intervening port cost parameters. Root path cost is the minimum path cost from a bridge to the root bridge.

Scanned with CamScanner

3) Root port : Each bridge determines its port through which if a frame is transmitted, it will reach the root bridge incurring the root path cost. This port of the bridge is called the root port.

4) Designated bridge and designated port : If a LAN has several bridges connecting it to the root bridge, one of the bridge is called the designated bridge and all the frames from the LAN are transmitted through designated bridge. The corresponding port of the bridge is called the designated port.

5) Construction of spanning tree : For spanning tree algorithm to work properly, each bridge must have unique identifier. Each port within a bridge must have distinct identity.

- First a root bridge is selected. Then each bridge selects a port through which the least cost path to the root bridge is found. This is called a root port.
- Then a specific designated bridge is selected for each LAN. Lastly each bridge puts its root port and all bridge ports to LANs for which it is designated into a forwarding state. The other bridge ports are said to be in a blocked state.
- When the network is in operation, the spanning tree algorithm exchanges status information between bridges via messages called **Bridge Protocol Data Units** (BPDUs).
- Each BPDU contains the following information.
  i) The identifier of this bridge and the port of this bridge.
  ii) The identifier of the bridge that this bridge considers to be the root.
  iii) The root path cost for this bridge.
- Consider the configuration shown in Fig. 5.12.6. Each bridge has a bridge identifier and each port has a port cost as shown.

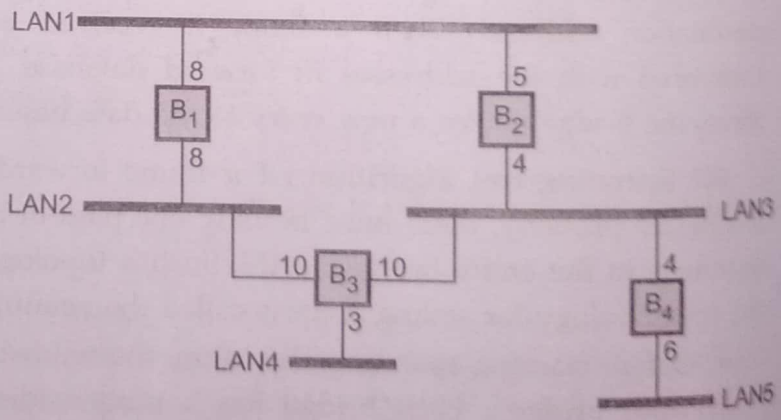


Fig. 5.12.6 Multiple interconnected LANs

- All bridges consider themselves to be the root bridge by broadcasting a BPDU on each of its LANs. On any LAN, only one claimant will have the lowest bridge identifier and will maintain this belief. The others will accept this fact by comparing the bridge identifier in the BPDUs they receive.
- Thus bridge B_1 will be identified as the root by bridges B_2 and B_3. They will identify the root port as well. Bridge B_4, will consider bridge B_2 as the root.

Scanned with CamScanner

- Bridge B_2 and B_3 release BPDUs indicating the root bridge identifier and the path cost to the root. These BPDUs are released only through port other than the root port. Thus, the BPDU released by bridge B_2 in LAN3 will indicate root bridge identifier as B_1 and root path cost as 5.
- Similarly, the BPDU released by bridge B_3 in LAN3 and LAN4 will indicate the root bridge identifier as B_1 and the root path cost as 10. When bridge B_4 receives these BPDUs, it will realize that the root identifier is B_1 and the root is accessible through bridge B_2 at a lower path cost of 5. Therefore, bridge B_2 is the designated bridge for LAN3 and the port of bridge B_2 connected to LAN3 is chosen as the designated port for transmission of frames to the root.

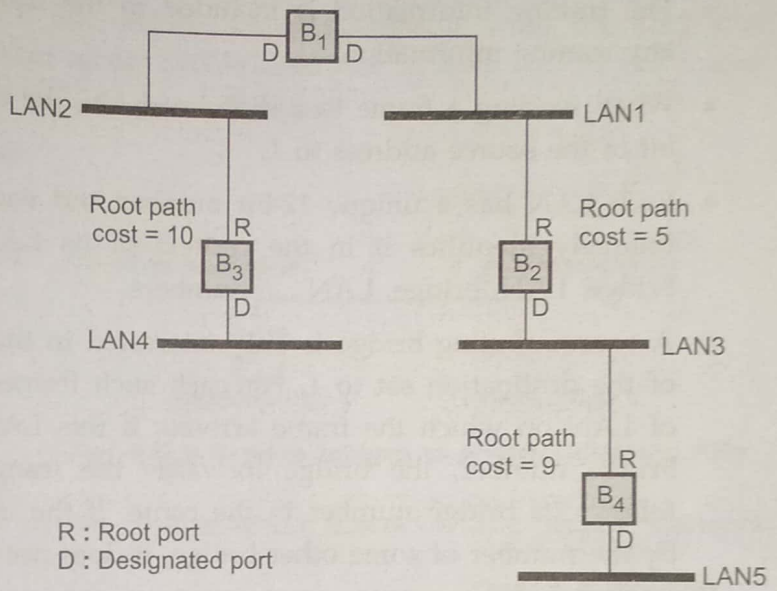


Fig. 5.12.7 Spanning tree configuration

- The port of bridge B_3 connected to LAN3 is put in the blocked state. Bridge B_4 will further propagate this information to other bridges connected to LAN5. It will indicate the root identifier as B_1 and the root path cost is 9. This process continues and finally we have the spanning tree with no loops. The results of this activity are shown in Fig. 5.12.7

Advantages of transparent bridge

1. Easy to use.
2. Just install the bridge, no software changes are needed in hosts.

Disadvantages of transparent bridge

1. Does not support multipath routing.
2. The path between any two hosts may not be the optimal path.
3. Broadcast and multicast frames must be flooded in all cases.

Scanned with CamScanner

5.12.3.5 Source Routing Bridges

- In spanning tree routing algorithm, the bridge uses the MAC destination address of a frame to direct it. The route is decided by the bridges.
- In source routing, each station on the extended LAN is expected to know the route over which the frame is to send.
- The routing information is included in the frames. The bridges do not maintain any routing information.
- When sending a frame to a different LAN, the source machine sets the high order bit of the source address to 1.
- Each LAN has a unique 12-bit number and each bridge has a 4-bit number that uniquely identifies it in the context of its LANs. A route is then sequence of bridge, LAN, bridge, LAN ..... numbers.
- A source routing bridge is only interested in those frames with the high order bit of the destination set to 1. For each such frames, it scans looking for the number of LAN on which the frame arrived. If this LAN number is followed by its own bridge number, the bridge forwards the frame onto the LAN whose number follows its bridge number in the route. If the incoming LAN number is followed by the number of some other bridge, it does not forward the frame.

Discovery frame :

- If the destination address is not known, the source issues a broadcast frame asking where it is ? This broadcast frame is discovery frame. These frames will thus travel through all possible paths between the source and destination stations.
- Along the way, each frame records the route it takes upon reaching the destination, the bridges record their identity in it, so that the original sender can see the exact route taken and ultimately choose the best route.

Advantages of source routing bridge

1. Uses the optimal route.
2. Better use of resources.
3. Also can make use of multiple path to same destination.

Disadvantages of source routing bridge

1. Not transparent to hosts.
2. Each host must detect bridge failure on its own.
3. Installing new bridges non-trivial.

Scanned with CamScanner

5.12.3.6 Remote Bridges

- A common use of bridges is to connect two or more distant LANs. But if two LANs are located at greater distance, these LANs are to be interconnected. So the complete system acts like one large LAN. For this, one bridge may not serve the purpose because of the distance limitation of the LANs or the non-availability of a high speed transmission medium.
- One alternative is taken as a full duplex leased connection from the telephone network and connect the LANs using two bridges one at each end of the leased connection. These bridges are called **remote bridges**.
- Fig. 5.12.8 shows remote bridge.
- The bridges establish a data link connection between two or more LANs through leased circuit and carry out the bridge operation.

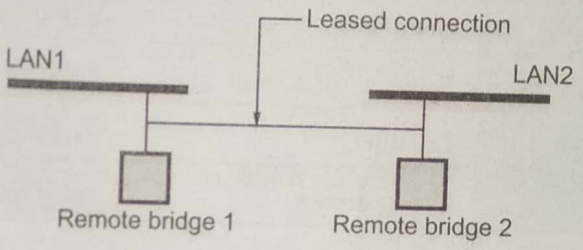


Fig. 5.12.8 Remote bridges to connect distant LANs

- On one port of the bridge, HDLC protocol takes care of the transmission errors of the leased network connection.
- HDLC protocols implement the MAC frame transport service.
- The MAC frame is encapsulated in the information field of an HDLC frame using a header and trialer at the transmitting end. A new MAC header and trailer then be generated at the destination bridge.

Features of bridge at a glance

A) Bridge can do following :

1. Filter traffic by reading packet address.
2. Link dissimilar network.
3. Link segments of a network together.

B) Bridge cannot do following :

1. Determine the most efficient path to transmit data.
2. Traffic management function.

C) Benefits provided by a bridge :

1. Expand the length of an existing network.
2. Increase the number of workstations on the network.
3. Reduce traffic congestion.

Scanned with CamScanner

4. Provide a connection to a dissimilar network.
5. Move data across a intermediate network with a dissimilar protocol.

5.12.3.7 Comparison between Transparent Bridge and Source Routing Bridge

Sr. No.	Parameter	Transparent bridge	Source routing bridge
1.	Orientation	Connectionless	Connection-oriented
2.	Configuration	Automatic	Manual
3.	Routing	Suboptimal	Optimal
4.	Complexity	In the bridge	In the hosts
5.	Failures	Handled by the bridges	Handled by the host
6.	Locating	Backward learning	Discovery frame
7.	Transparency	Fully transparent	Not transparent

5.12.4 Switch

- Switches operate at the Data Link layer (layer 2) of the OSI model. It can interpret address information.
- Switches resemble bridges and can be considered as multi-port bridges. By having multi-ports, can better use limited bandwidth and prove more cost-effective than bridge.
- Switches divide a network into several isolated channels. Packets sending from 1 channel will not go to another if not specify. Each channel has its own capacity and need not be shared with other channels

Advantages of switches

1. Switches divide a network into several isolated channels or collision domains.
2. Reduce the possibility of collision.
   i. Collision only occurs when two devices try to get access to one channel.
   ii. Can be solved by buffering one of them for later access.
3. Each channel has its own network capacity.
   i. Suitable for real-time applications e.g. video conferencing.

Limitations of switches

1. Although contains buffers to accommodate bursts of traffic, can become overwhelmed by heavy traffic.

Scanned with CamScanner

2. Device cannot detect collision when buffer full.
3. CSMA/CD scheme will not work since the data channels are isolated, not the case as in Ethernet.
4. Some higher level protocols do not detect error e.g. UDP.

1) Layer 2 switch

- Layer 2 switch performs at the physical and data link layer.
- It is a **bridge** with many ports and a design that allows better performance.
- Layer 2 switch operates using physical network addresses, identify individual devices. Most hardware devices are permanently assigned this number during the manufacturing process.
- Switches operating at Layer 2 are very fast because they are just soring physical addresses, but they usually aren't very smart.
- They don't look at the data packet very closely to learn anything more about where it's headed.

2) Layer 3 switch

- Layer 3 switches use network or IP addresses that identify locations on the network. They read network addresses more closely than Layer 2 switches.
- They identify network locations as well as the physical device. A location can be a LAN workstation, a location in a computer memory or even a different packet of data travelling through a network.
- Switches operating at Layer 3 are smarter than Layer 2 devices and incorporate routing functions to actively calculate the best way to send a packet to its destination.
- But although they are smarter, they may not be as fast as their algorithms, fabric and processor don't support high speeds.

3) Layer 4 switch

- Layer 4 of the OSI model co-ordinates communications between systems. Layer 4 switches are capable of identifying which application protocols (HTTP, SNTP, FTP) are included with each packet and they use this information to hand off the packet to the appropriate higher-layer software.
- Layer 4 switches make packet forwarding decisions based not only on the MAC address and IP address, but also on the application to which a packet belongs. Because Layer 4 devices enable you to establish priorities for network traffic based on application, you can assign a high priority to packets belonging to vital

Scanned with CamScanner

in-house applications such as peoplesoft, with different forwarding rules for low priority packets generic HTTP-based Internet traffic.

- Layer 4 switches also provide an effective wire-speed security shield for your network because any company or industry-specific protocols can be confined to only authorized switched ports or users. This security feature is often reinforced with traffic filtering and forwarding features.

5.12.5 Routers

- A router is a three layer device that routes packets based on their logical addresses. Router connects two or more networks. It consists of combination of the hardware and software. Fig. 5.12.9 shows the router.

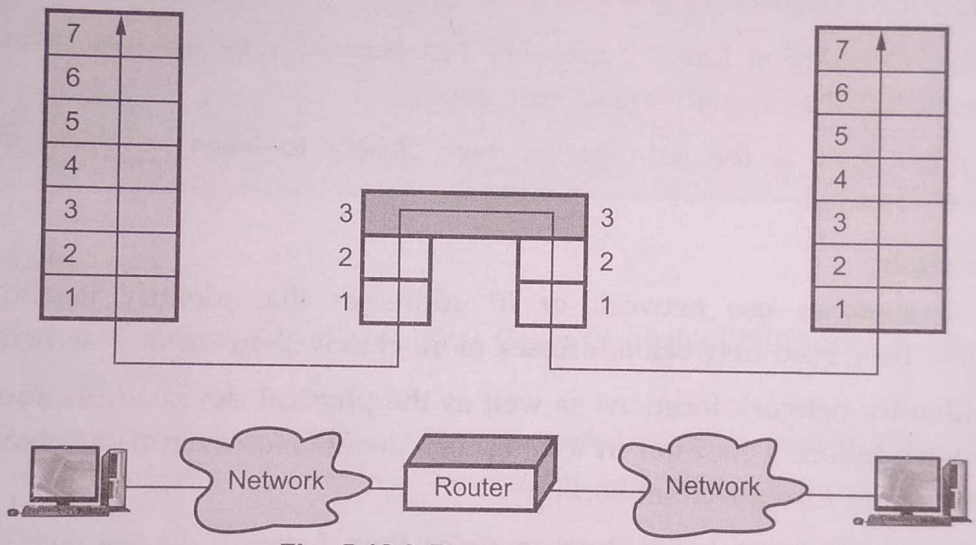


Fig. 5.12.9 Router in OSI model

- A router normally connects LANs and WANs in the Internet and has a routing table that is used for making decisions about the route. Routers connect dissimilar networks together and have access to information from physical, data link and network layer.
- The key feature of a router is to determine the shortest path to destination. A router forwards packet by examining protocol address at network layer, look up the address in the routing table, then forward the packet to the next hop.
- Router uses one or more routing algorithms to calculate the best path through an internetwork.

5.12.6 Gateways

- Gateway connects two independent networks. A gateway is protocol converter.
- It operates in all seven layers of the OSI model.

Scanned with CamScanner

- A gateway can accept a packet formatted for one protocol (e.g. TCP/IP) and convert it to a packet formatted for another protocol (e.g. Apple Talk) before forwarding it.
- The gateway must adjust the data rate, size and data format. Gateway is generally software installed within a router.
- Fig. 5.12.10 shows the gateway.

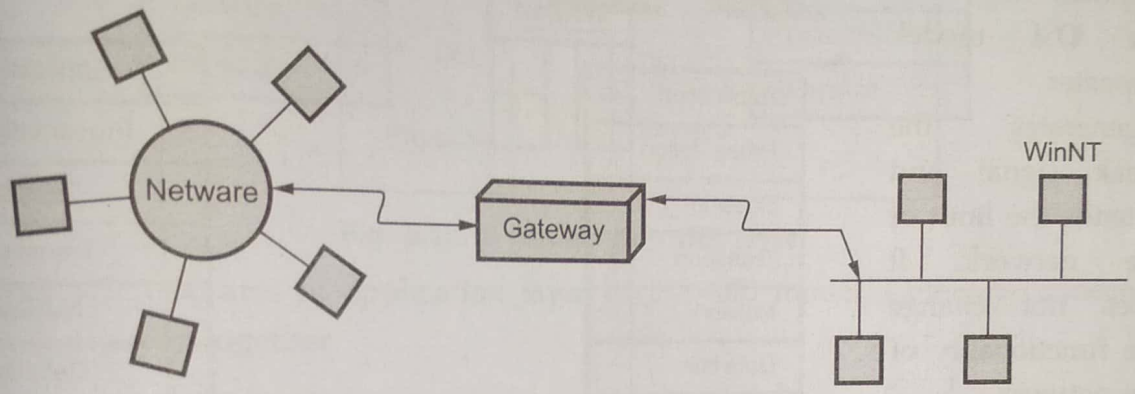


Fig. 5.12.10 Gateway

5.12.7 Network Interface Card (NIC)

- The Network Interface Cards (NICs) are also called as Network Adapter. The primary function of NIC is to allow the computer to communicate on the network. It supports transmitting, receiving and controlling traffic with other computers on the network. NIC operates at physical layer of OSI model.
- When NIC transmits data on network, it converts data from parallel to serial then encodes and compresses it.
- When NIC receives data, it translates the electrical signal into equivalent binary data bits which can be accepted by computer.
- NIC can be installed in computer expansion bus via ISA or PCI slot. ISA slot is 8 or 16-bit and PCI slot is 32 bits. Now a days, NIC cards are in-built in the motherboard itself.
- A NIC is specific to a particular type of LAN architecture. For example, fiber-optic, token ring and ethernet. NIC has one or more external parts with which network cable can be attached. Most NICs support both 10 Base T and 100 Base TX.
- MAC address is hard coded onto the card by manufacturer. This MAC address is globally unique and is of 48 bits. The MAC address provides a way to distinguish one NIC from other NIC. These MAC addresses are also called physical addresses.

Scanned with CamScanner

5.12.8 Difference between Repeater, Bridge, Router and Gateway

- **Repeater :** Repeater is an electronic device. It operates on only the physical layer of the OSI model. Repeater regenerates the weak signal and extends the limit of the network. It does not change the functionality of the network.

Bridge : It operates at the data link layer. It connects two networks together.

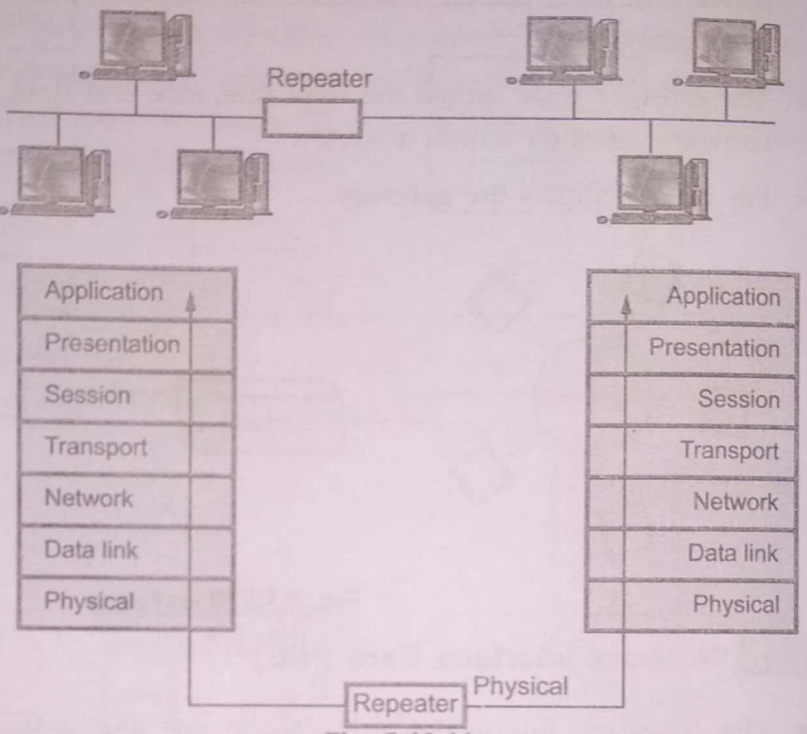


Fig. 5.12.11

Router : Router operates at the third layer i.e. network layer of the ISO-OSI model. It connects more than two different types of networks. Router determines the short path in between source and destination for data transmission.

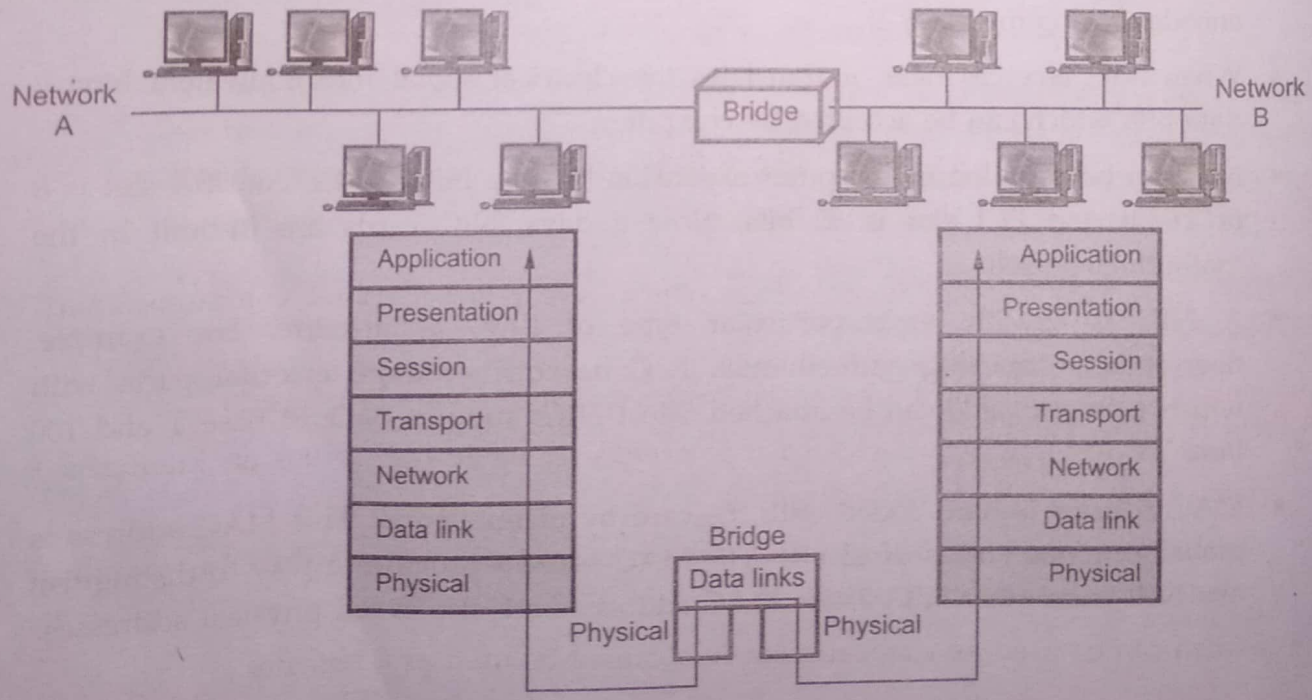


Fig. 5.12.12 Bridge with OSI layer

Scanned with CamScanner

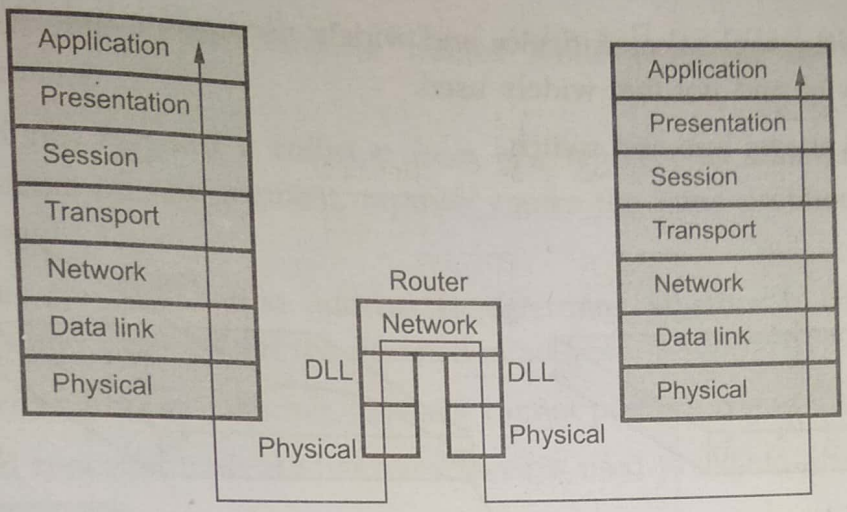


Fig. 5.12.13 Router with OSI layer

Gateway : It operates at application layer of ISO-OSI model. Gateway connects two dissimilar networks together.

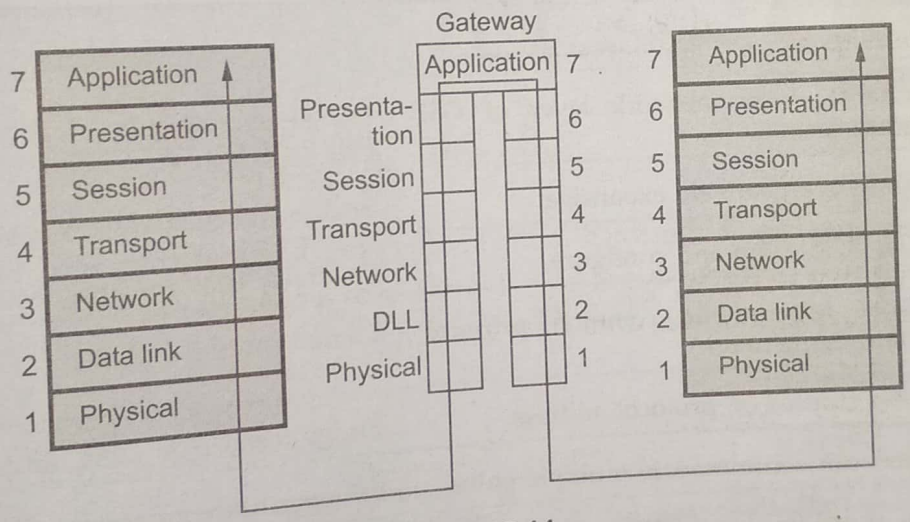


Fig. 5.12.14

5.12.9 Comparison of Hub and Switch

- Hub is a broadcasting device while switch is a point to point communication device.
- Hub operates at physical layer while switch operates at data link layer.
- Switch is an intelligent device so it is expensive, while hub is not an intelligent device so it is comparatively cheap.
- Switch uses switching table to find out the correct destination while hub simply broadcasts the incoming packet.
- Switch can be used as a repeater but hub cannot be used as a repeater.

Scanned with CamScanner

- Switch is very sophisticated device and widely used while hub is an ordinary old type of device and not that widely used.
- Fig. 5.12.15 shows hub and switch.

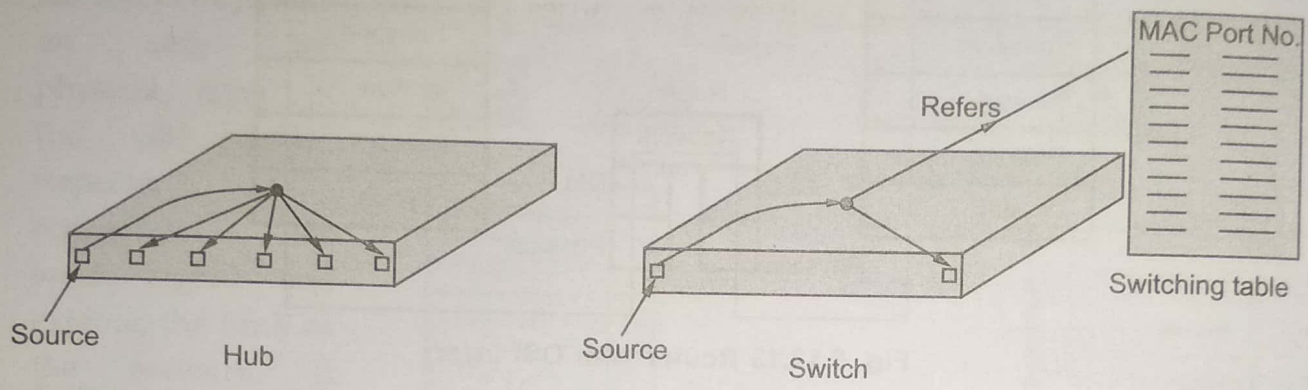


Fig. 5.12.15 Hub and switch

5.12.10 Comparison between Router and Bridge

Sr. No.	Router	Bridge
1.	Router operates network layer of OSI model.	Bridge operates the data link layer of OSI model.
2.	Routers are relatively expansive.	Bridges are relatively inexpansive.
3.	Difficult to setup and configure.	Relatively easy to configure.
4.	Cannot route some common protocol (within the network).	Cannot route the packet.
5.	Router focuses on protocol address.	Bridge focuses on MAC address.
6.	Router can accommodate multiple paths.	Bridge can accommodate single path.
7.	Can route packets to reduce network bottlenecks.	Filter packets faster than routers.
8.	Routers are good solution for joining remote network.	Bridges are good for segment network.
9.	Join two different networks.	Extends the existing network.
10.	Routers are both hardware and software device.	Bridges are both hardware and software device.

5.12.11 Difference between Bridge and Repeater

1. Bridge operates the data link layer while repeater operates at physical layer of OSI model.

Scanned with CamScanner

2. Bridge understands the complete frames while repeaters do not understand complete frames.
3. Bridge will not forward a collision from one segment to another. With repeater, collision occurs on one segment, repeater causes the same problem to occur on all other segment.
4. Bridge uses the destination address to determine whether to forward a frame. Repeater cannot understand the destination address.
5. Bridge performs frame filtering. Repeater cannot perform frame filtering.
6. Bridge and repeater, both are hardware devices used to extend a LAN.

University Questions

1. *Explain the usage of following devices. Also discuss the OSI layer at which they are used.*
   1. Bridge 2. Repeater 3. Hub and Switch 4. Router 5. Gateway
   GTU : Winter-14, Marks 7
2. *Write note on : Internet Control Message Protocol (ICMP), bridge and switches.*
   GTU : Dec.-10, Marks 7
3. *Compare transparent bridge and source routing bridge.*
   GTU : June-11, Marks 7
4. *What is gateway ? How it works ?*
   GTU : Dec.-11, Marks 4
5. *Differentiate router and switch.*
   GTU : Dec.-11, Marks 4
6. *What is the similarity and difference between switch and bridge ? Also explain cut through switched.*
   GTU : May-12, Marks 3
   GTU : May-12, Marks 4
7. *Explain spanning tree bridges*
   GTU : May-12, Marks 3
8. *Explain remote bridges.*
   GTU : Summer-13, Marks 7
9. *Explain the working principle of bridges.*
   GTU : Summer-14, Marks 3
10. *Explain router and its use.*

❑❑❑

Scanned with CamScanner