4AID4-07: Data Communication and Computer Networks

Credit: 3 **Max. Marks: 100(IA:30, ETE:70)**

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

SN	Contents	Hours
1	**Introduction:** Objective, scope and outcome of the course.	1
2	Introductory Concepts: Network hardware, Network software, topologies, Protocols and standards, OSI model, TCP model, TCP/IP model, Physical Layer: Digital and Analog Signals, Periodic Analog Signals, Signal Transmission, Limitations of Data Rate, Digital Data Transmission, Performance Measures, Line Coding, Digital Modulation, Media and Digital Transmission System	7
3	Data Link Layer: Error Detection and Correction, Types of Errors, Two dimensional parity check, Detection verses correction, Block Coding, Linear Block Coding, Cyclic Codes, Checksum, Standardized Polynomial Code, Error Correction Methods, Forward Error Correction, Protocols: Stop and wait, Go-back-N ARQ, Selective Repeat ARQ, Sliding window, Piggy backing, Pure ALOHA, Slotted ALOHA, CSMA/CD, CSMA/CA	9
4	Network Layer: Design issues, Routing algorithms: IPV4, IPV6, Address mapping: ARQ, RARQ, Congestion control, Unicast, Multicast, Broadcast routing protocols, Quality of Service, Internetworking	8
5	Transport Layer: Transport service, Elements of transport protocols, User Datagram Protocol, Transmission Control Protocol, Quality of service, Leaky Bucket and Token Bucket algorithm	8
6	Application Layer**:** WWW, DNS, Multimedia, Electronic mail, FTP, HTTP, SMTP, Introduction to network security	7
	Total	40

DATA COMMUNICATION AND COMPUTER NETWORKS

IV C.S.

1 INTRODUCTORY CONCEPT

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 ***What is physical address?*** *[R.T.U. 2019]*

Ans. Physical Address : In computing, physical address refers to a memory address or the location of a memory cell in the main memory. It is used by both hardware and software for accessing data. Software, however, does not use physical addresses directly; instead, it accesses memory using a virtual address. A hardware component known as the memory management unit (MMU) is responsible for translating a virtual address to a physical address.

In networking, physical address refers to a computer's MAC address, which is a unique identifier associated with a network adapter that is used for identifying a computer in a network.

In computing, a physical address (also real address or binary address) is a memory address that is represented in the form of a binary number on the address bus circuitry in order to enable the data bus to access a particular storage cell of main memory or a register of memory mapped I/O device.

Q.2 ***Write any two differences between UPD and TCP.*** *[R.T.U. 2019]*

Ans. Transmission Control Protocol (TCP) : TCP is a reliable connection-oriented protocol that allows a byte stream originating on one machine to be delivered without errors on any other machine in the internet. It fragments the incoming byte stream into discrete message and passes each one to the internet layer. At the destination, the receiving TCP process reassembles the received messages into the output stream. TCP also handles flow control to make sure a fast sender cannot swap a slow receiver with more messages than it can handle.

User Datagram Protocol (UDP) : UDP is an unreliable, connectionless protocol for application that do not want TCP's sequencing or flow control and wish to provide their own. It is also widely used for one-shot client-server type, request-reply queries and applications in which prompt delivery is more important than accurate delivery, such as transmitting speech or video. The relation of IP, TCP and UDP is shown in fig.

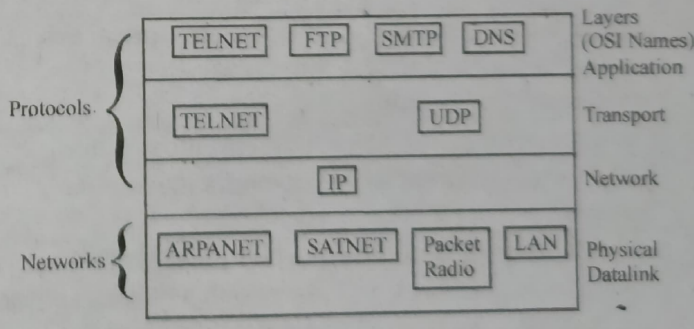


Fig. : Protocols and networks in the TCP/IP model initially

Q.3 ***Define framing and the reason for its need.*** *[R.T.U. 2019]*

Ans. Framing : The data link layer divides the stream of bits received from the network layer into manageable data units called frames.

Reason for its Need : The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption. A glitch in line during the transmission will corrupt some frames. Only the lost frames and not the entire set of data needs to be retransmitted.

Q.4 ***List the seven layers of the OSI model.***

[R.T.U. 2019]

Ans. OSI/ISO Model : Seven layers are shown in fig.

7	Application Layer
6	Presentation Layer
5	Session Layer
4	Transport Layer
3	Network Layer
2	Data Link Layer
1	Physical Layer

Fig. : OSI layers

Q.5 ***What is difference between analog and digital signals.*** *[R.T.U. 2019]*

Ans. Difference between Analog and Digital Signal

Analog Signals	Digital Signals
Continuous Signals	Discrete Signals
Represented by sine waves	Represented by square waves
Human voice, natural sound, analog electronic devices are few examples.	Computer, optical drives, and other electronic devices.
Continuous range of values	Discontinuous values
Records sound waves as they are	Converts into a binary waveform.
Only Be used in analog devices.	Suited for digital electronics like computer, mobiles, and more.

Q.6 ***Write advantages of Digital Modulation.***

Ans. (i) Digital Modulation gives improved reliability due to channel coding.

(ii) Digital Modulation are cheaper compare to analog communication system.

Q.7 ***What does network software mean?***

Ans. Network software is an extremely broad term for a range of software aimed at the design and implementation of modern networks. In using network software, the size and scope of a network plays a key role in decision making. Companies or other parties can choose specific network access tools for setup and installation. Other network software resources help administrators and security personal to monitor a network to protect it against a range of attacks.

Q.8 ***What are the functions of network software (Write any 2).***

Ans.(i) Helps to setup and install computer network.
(ii) Enable users to have access to network resources in a Seamless manner.

Q.9 ***Write the different components of network hardware.***

Ans. The basic computer hardware components that are needed to setup a network are as follows:

(i) Network cables
(ii) Routers
(iii) Repeaters
(iv) Hubs
(v) Switches
(vi) Bridges
(vii) Gateways
(viii) Network interface card etc.

PART-B

Q.10 ***Explain the services provided by the TCP.***

[R.T.U. 2019]

Ans. Transport Layer Services : The Transport Layer is a higher level layer on the Open Systems Interconnect model than the Data Link Layer. The Transport Layer is responsible for checking that data packets created in the Sessions Layer are error free. The Transport Layer ensures that the protocols operated at this layer provide reliable end-to-end flow and error control. The most common protocol that you would see at this layer is TCP/IP.

- **Types of service :** The transport layer also decides the type of service that should be provided to the session layer. The service may be perfectly reliable, or may be reliable within certain tolerances or may not be reliable at all. The message may or may not be received in the order in which it was sent. The decision regarding the type of service to be provided is taken at the time when the connection is established.
- **Error Control :** If reliable service is provided then error detection and error recovery opeations are also performed. It provides error control mechanism on end to end basis.
- **Flow Control :** A fast host cannot keep pace with a slow one. Hence, this is a mechanism to regulate the flow of information.
- **Connection Establishment/Release :** The transport layer also established and releases the connection across the network. This requires some sort of naming mechanism so that a process on one machine can indicate with whom it wants to communicate.

Q.11 ***What is line coding? Explain its characteristics.***

[R.T.U. 2019]

OR

Explain the types of line coding in brief.

[R.T.U. 2011, 10]

Ans. Line Coding : The digital data can be transmitted by various transmission or line codes such as on-off, polar bipolar and so on. This is called *line coding.*

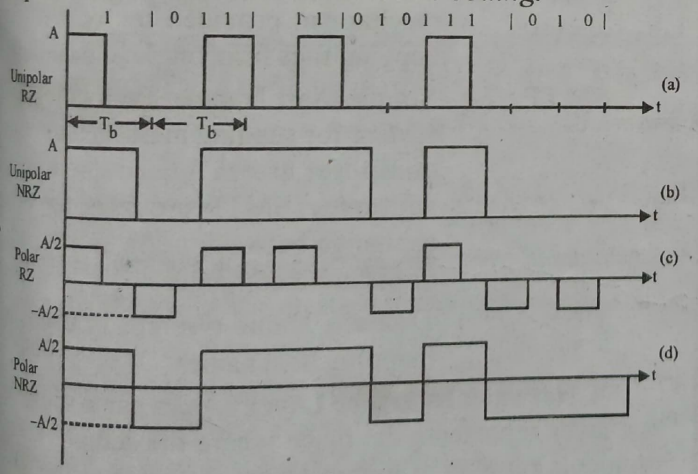


Fig.

A line code must have the following properties :

(1) Transmission bandwidth must be as small as possible.

(2) Power efficiency also as small as possible.

(3) Error detection and correction capability .

(4) Favourable power spectral density.

(5) Adequate timing content must be possible to extracted timing or clock information from the signal.

(6) It must be possible to transmit a digital signal correctly regardless of the pattern of 1's and 0's.

Various PAM format for line codes

Some important PAM formats are as under :

(1) Non-return to zero (NRZ) and return to zero (RZ) unipolar formats.

(2) NRZ and RZ polar format.

(3) Non-return to zero bipolar format.

(4) Manchester format.

(5) Polar quaternary NRZ format.

(1) Unipolar RZ and NRZ

In this, the waveform has single polarity. The waveform can have +5 or +12 volts when high. The waveform is simple on-off. In the unipolar RZ format the waveform has zero value when symbol '0' is transmitted and waveform has 'A' volt when '1' is transmitted. In RZ, the 'A' volts is present for $T_b/2$ period if symbol '1' is transmitted and for remaining $T_b/2$, waveform returns to zero value *i.e.* for unipolar RZ form, we have,

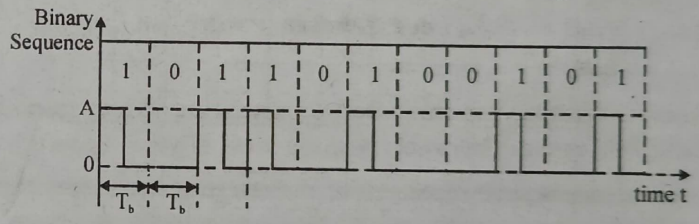


Fig.

If symbol '1' is transmitted then

$$x(t) = \begin{cases} A \text{ for } 0 \le t \le T_b/2 \text{(half interval)} \\ 0 \text{ for } T_b/2 \le t < T_b \text{(half interval)} \end{cases}$$

and if symbol then

$x(t) = 0$ for $0 \le t < T_b$ (complete interval)

Thus for unipolar NPZ format.

If symbol '1' is transmitted we have

$x(t) = A$ for $0 \le t < T_b$ (complete interval)

If symbol '0' is transmitted we have

$x(t) = 0$ for $0 \le t < T_b$ (complete interval)

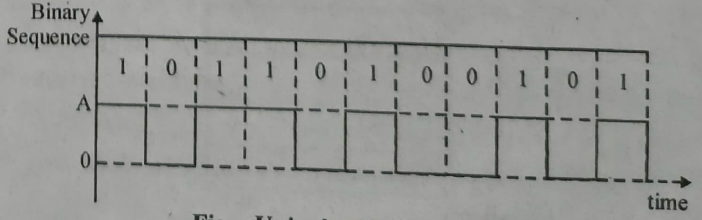


Fig. : Unipolar NRZ Format

DCCN. 4

(2) Polar NRZ and RZ

For polar RZ format

If symbol '1' is transmitted then

$$x(t) = \begin{cases} +A/2 \text{ for } 0 \le t < T_b/2 \\ 0 \text{ for } T_b/2 \le t < T_b \end{cases}$$

and

If symbol '0' is transmitted, then

$$x(t) = \begin{cases} -A/2 \text{ for } 0 \le t < T_b/2 \\ 0 \text{ for } T_b/2 \le t \le T_b \end{cases}$$

for Polar NRZ

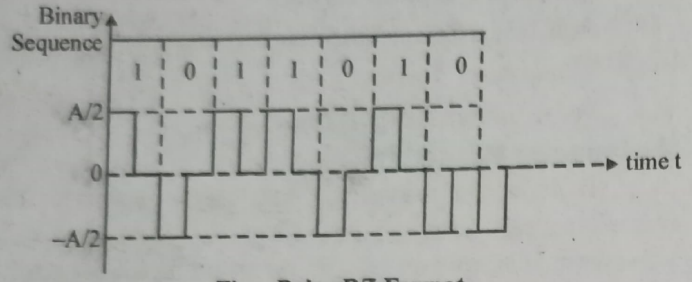
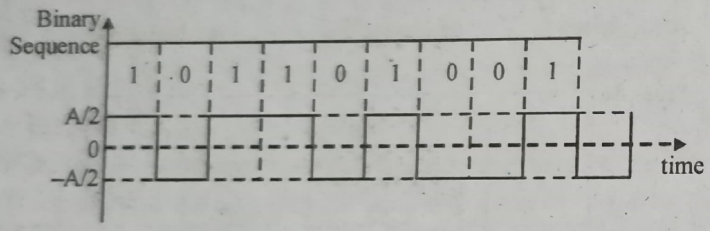


Fig. : Polar RZ Format

If symbol '1' is transmitted, then

$$x(t) = \frac{-A}{2} \text{ for } 0 \le t < T_b$$



Q.12 Explain TCP/IP model with suitable diagram.

[R.T.U. 2019]

Ans. TCP/IP Model : TCP/IP protocol suit, used in the internet, was developed prior to OSI model. Therefore, the layers in the Transmission Control Protocol/ Internet-working Protocol (TCP/IP) do not match exactly with those of OSI model.

TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality, but they are not necessarily interdependent.

Application
Transport
Network
Data Link
Physical

Fig. : TCP/IP reference model

1. Internet Layer : Its job is to permit hosts to inject packets into any network and have them travel independently to the destination (potentially on a different network). They may arrive in a different order than they were sent, in which case it is the job of higher layers to rearrange them, if inorder delivery is desired.

The internet layer defines an official packets format and protocol called IP (Internet Protocol). The job of the internet is to deliver IP packets where they are supposed to go. TCP/IP Internet layer is similar in functionality to the OSI network layer.

2. Transport Layer : The layer above the internet layer is transport layer. It is designed to allow peer entities on the source and destination hosts to carry on a conversation, just as in transport layer of OSI.

Two end to end transport protocols are defined here :

- **Transmission Control Protocol (TCP) :** *Refer to Q.2.*
- **User Datagram Protocol (UDP) :** *Refer to Q.2.*

3. Application Layer : On the top of transport layer is the application layer. It contains higher level protocols.

TELNET : Telnet is an abbreviation for Terminal Network. TELNET enables the establishment of a connection to a remote system in such a way that the local terminal appears to be a terminal at the remote system TELNET is a general purpose client-server application program.

FTP : File Transfer Protocol is the standard mechanism provided by TCP/IP for copying files from one host to another.

SMTP : Simple Mail Transfer Protocol. It is a system for sending messages to other computer users based on e-mail addresses. SMTP provides for mail exchange between users on the same or different computer.

DNS : Domain Name System is used for mapping host names

4. Network Interface Layer : As its name suggests, this layer represents the place where the actual TCP/IP protocols running at higher layers interface to the local network. This layer is somewhat "controversial" in that some people don't even consider it a "legitimate" part of TCP/IP. This is usually because none of the core IP protocols run at this layer. Despite this, the network interface layer is part of the architecture. It is equivalent to the data link layer (layer two) in the OSI Reference Model and is also sometimes called the *link layer.*

On many TCP/IP networks, there is no TCP/IP protocol running at all on this because it is simply not needed. For example, if we run TCP/IP over an Ethernet, then Ethernet handles layer two (and layer one) functions. However, the TCP/IP standards do define protocols for TCP/IP networks that do not have their own layer two implementation. These protocols, the Serial Line Internet Protocol (SLIP) and the Point-to-Point Protocol (PPP), serve to fill the gap between the network layer and the physical layer. They are commonly used to facilitate TCP/IP over direct serial line connections (such as dial-up telephone networking) and other technologies that operate directly at the physical layer.

Data Link Layer Services

Unacknowledged connectionless service :

- No acks, no connection
- Error recoverry up to higher layers
- For lowerror-rate links or vice traffice

Acknowledge connectionless service :

- Acks improve reliability
- For unrealiable channels. E.g.: Wireless Systems

Acknowledge connection-oriented service :

- Equivalent of reliable bit-stream
- Connection establishment
- Packets delivered In-Order
- Connection Release
- Inter-Router Traffic

Transport Layer services : *Refer to Q.10.*

Q.13 ***Differentiate between Analog and Digital transmission.*** *[R.T.U. 2017]*

Ans. Difference between Analog and Digital Transmission : Analog transmission is a method of conveying voice, data, image, signal, or video information. It uses a continuous signal varying in amplitude, phase, or another property that is in proportion to a specific characteristic of a variable. Analog transmission could mean that the transmission is a transfer of an analog source signal which uses an analog modulation method (or a variance of one or more properties of high frequency periodic waveform, also known as a carrier signal). FM and AM are examples of such a modulation. The transmission could also use no modulation at all. It is most notably an information signal that is constantly varying.

Data transmission (also known as digital transmission or digital communications) is a literal transfer of data over a point to point (or point to multipoint) transmission medium such as copper wires, optical fibres, wireless communications media, or storage media. The data that is to be transferred is often represented as an electro-magnetic signal (such as a microwave). Digital transmission transfers messages discretely. These messages are represented by a sequence of pulses via a line code. However, these messages can also be represented by a limited set of wave forms that always vary. Either way, they are represented using a digital modulation method.

Analog transmission is capable of being conveyed in a no fewer than four ways: through a twisted pair or coax cable, through a fibre optic cable, through the air, or through water. There are, however, only two basic types of analog transmission. The first is known as amplitude modulation (or AM). This is a technique used in electronic communication and works by alternating the strength of a transmitted signal in relation to the information that is being sent. The second is known as frequency modulation (or FM). This type of communication conveys information over a carrier wave, just as AM transmission. However, FM communication alternates the frequency of the transmitted signal.

Data that is transmitted via digital transmission may be digital messages that have origins for a data source (a computer or a keyboard, for example). However, this transmitted data may also be from an analog signal (a phone call or a video signal, for example). It may then be digitized into a bit stream using pulse code modulation (or PCM) or even more advanced source coding schemes. The coding of the data is carried out using codec equipment.

Q.14 ***(a) The loss in a cable is usually defined in decibels per kilometre. If the signal at the beginning of a cable with -0.3 dB/km has a power of 2mW, what is the power of the signal at 5km?***

(b) A digital signal has eight levels. How many bits are needed per level.

(c) Explain NRZ-L, NRZ-I and RZ line encoding. *[R.T.U. 2015]*

Ans.(a) The relation between decibel change and electrical power is a logarithmic one and is given by the following expression:

$$\Delta dB = 10*\log(P_{out}/P_{in})$$

$$P_{out} = P_{in} * 10^{(\Delta dB/10)}$$

The change in decibels that would take place over 5 kms = –0.3 dB/km * 5 km = –1.5 dB

Plugging values $\Delta dB = -1.5$ and $P_{in} = 2mW$ in the expression, we get

$P_{out} = 2\,mW * 0.707945784$

$P_{out} = 1.415891568\,mW$

Ans.(b) In general, 2^n levels require n bits. So 8 levels will require 3 bits ($2^3 = 8$). Thus, each signal level is represented by 3 bits. For example, the fifth level will be represented by 101.

Ans.(c) Non Return to Zero (NRZ) : NRZ codes share the property that voltage level is constant during a bit interval. High level voltage = bit 1 and Low level voltage = bit 0. A problem arises when there is a long sequence of 0's or 1's and the volatage level is maintained at the same value for a long time. This creates a problem on the recieving end because now, the clock synchronization is lost due to lack of any transitions and hence, it is difficult to determine the exact number of 0s or 1s in this sequence.

The two variations are as follows:

1. **NRZ-Level (NRZ-L):** In NRZ-L encoding, the polarity of the signal changes only when the incoming signal changes from a 1 to a 0 or from a 0 to a 1. NRZ-L method looks just like the NRZ method, except for the first input one data bit. This is because NRZ does not consider the first data bit to be a polarity change, where NRZ-L does.
2. **NRZ-Inverted (NRZ-I):** Transition at the beginning of bit interval = bit 1 and no transition at beginning of bit interval = bit 0 or viceversa. This technique is known as differential encoding.

NRZ-I has an advantage over NRZ-L. Consider the situation when two data wires are wrongly connected in each other's place. In NRZ-L all bit sequences will get reversed (Because voltage levels get swapped). Whereas in NAZ-I since bits are recognized by transition the bits will be correctly interpreted. A disadvantage in NRZ codes is that a string of 0's or 1's will prevent synchronization of transmitter clock with receiver clock and a separate clock line need to be provided.

In positive-logic NRZ, the low state is represented by the more negative or less positive voltage , and the high state is represented by the less negative or more positive voltage.

Examples are:

Logic 0 = +0.5 volts
Logic 1 = +5.0 volts
Logic 0 = –3.0 volts
Logic 1 = 0.0 volts

In negative-logic NRZ, the low state is represented by the more positive or less negative voltage, and the high state is represented by the less positive or more negative voltage.

Examples are:

Logic 0 = +5.0 volts
Logic 1 = +0.5 volts
Logic 0 = 0.0 volts
Logic 1 = –3.0 volts

Return-to-Zero (RZ): It refers to a form of digital data transmission in which the binary low and high states, represented by numerals 0 and 1, are transmitted by voltage pulses having certain characteristics. The signal state is determined by the voltage during the first half of each data binary digit. The signal returns to a resting state (called zero) during the second half of each bit. The resting state is usually zero volts, although it does not have to be.

In positive-logic RZ, the low state is represented by the more negative or less positive voltage, and the high state is represented by the less negative or more positive voltage.

Examples are:

Logic 0 = 0 volts for 1 bit
Logic 1 = + 5 volts for 1/2 bit, then 0 volts for 1/2 bit
Logic 0 = – 4 volts for 1/2 bit, then 0 volts for 1/2 bit
Logic 1 = 0 volts for 1 bit

In negative-logic RZ, the low state is represented by the more positive or less negative voltage, and the high state is represented by the less positive or more negative voltage.

Examples are:

Logic 0 = +5 volts for 1/2 bit, then 0 volts for 1/2 bit
Logic 1 = 0 volts for 1 bit
Logic 0 = 0 volts for 1 bit
Logic 1 = – 4 volts for 1/2 bit, then 0 volts for 1/2 bit.

Q.15 ***What is Network Hardware and how does it work?***

Ans. Network hardware is the individual components of a network system that are responsible for transmitting data and facilitating the operations of a computer network. Although a network contains many hardware components, there are several basic categories that make up the complete operations of a network system. Here are some of the different categories and how they contribute as a whole to the overall functioning of a network system.

Basic network infrastructure is connected by components that fall under several categories of different types of network hardware.

Network Router: A network router is a hardware device that is connected to multiple channels for different networks through an interface that is situated on each

network. The router is usually located within the layers of a network that determine the path for the transfer of data with the router acting as a processing unit for information packets. The router duplicates information packets for use during transmission from one network to another. The router uses a specific protocol or set of rules to determine which information packets are to be routed to certain interfaces within the network. Different types of routers perform different functions depending upon the requirements of the network system.

Network Interface Card : Network interface cards are used to connect each computer to the network so they can communicate with the network router to receive information packets. The interface cards determine the infrastructure of a local area network (LAN) and allow all of the computers to connect to the network. There are many different types of network interface cards that perform different functions within the network which include Ethernet cards and wireless network interface cards.

Network Switches : Network switches work similar to routers because they both copy information from one area of the network to the other. However, network switches contain multiple ports for copying frames of information from one port to the other. Like routers, switches operate within the layers of a network and evaluate every frame before determining the port in which the frame should be copied. Network switches are more sophisticated then their predecessor the network hub, which copied all frames to all ports instead of determining individual destinations. This required more bandwidth than what is required with network switches.

Network Bridge : A network bridge divides traffic on a local area network by separating the LAN into several different segments. It is also responsible for filtering data by determining the data destination or discarding unnecessary data. Network bridges operate within the layers of the network and also control the data that crosses the boundaries from one local area network to the other.

PART-C

Q.16 ***Explain any two functions of each layer in the OSI model.*** *[R.T.U. 2019]*

Ans. OSI/ISO Model : *Refer to Q.4.*

1. Physical Layer : The physical layer coordinates the functions required to transmit a bit stream over a physical medium.

- **Line Configuration :** The physical layer is concerned with the connection of device to the medium. In a multipoint configuration a link is shared between several devices. In a point to point configuration, two devices are connected together through a dedicated link.
- **Interface and Media :** Physical layer defines the type of transmission medium. It defines the characteristics of the interface between the devices and transmission media.
- **Bit Representation :** Physical layer data consists of a stream of bits without any interpretation. Before encoding bits must be encoded into signals, optical or electrical. The physical layer defines the encoding.
- **Transmission Rate :** Physical layer defines the duration of bit, which is how long it lasts and the number of bits sent each second is also defined by the physical layer.
- **Transmission Mode :** Physical layer defines the direction and mode of transmission between two devices : full duplex, half duplex or simplex.
- **Network Topology :** Physical layer also defines devices which are connected to each other to make network. Possible topologies are star, bus, ring, tree and mesh.

2. Data Link Layer : The data link layer transforms the physical layer, a row transmission facility to a reliable link and is responsible for node to node delivery.

Basic features of data link layer are the following :

- **Framing :** *Refer to Q.3.*
- **Physical Addressing :** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the physical address of the sender and/ or receiver of the frame.
- **Flow Control :** If the rate at which the data are absorbed by the receiver is less than the rate of data produced in the sender, the data link layer imposes a flow control mechanism to prevent overwhelming the receiver.
- **Error Control :** Data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to prevent duplication of frames. Error control is normally achieved through a trailer added to the end of frame.

- **Access Control :** When two or more devices are connected to same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

3. Network Layer : Network layer is responsible for source to destination delivery of a packet possibly across multiple networks.

- **Logical Addressing :** The physical addressing implemented by the data link layer handles the addressing problems locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that among other thing, includes the logical addressing of the sender and receiver.
- **Routing :** When independent networks or links are connected together to create an internet work or a large network, the connecting devices route the packet to their final destination. One of the function of this layer is to provide this mechanism.

4. Transport Layer : Transport layer is responsible for source to destination (end to end) delivery of entire message.

- **Service Point Addressing :** Computers often run several programs at the same time. For this reason, source to destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to the specific process (running program) on the other. The transport layer header therefore must include a type of address called service point address or port address. The network layer gets each packet to correct computer, the transport layers get each packet to correct process.
- **Segmentation and Reassembly :** A message is divided into transmittable segments, each segment containing a sequence number. These layers enable the transport layer to reassemble the message correctly upon arriving at destination and to identify and replace packet that were lost in transmission.
- **Flow Control :** The transport layer also controls the flow of the packets to prevent the source from sending packets faster than the destination can handle.
- **Error Control :** Connection oriented transport layer is responsible for error control between end to end link. Error control is usually done through retransmission.
- **Connection Control :** Transport layer provides both services connection oriented and connectionless. A connection oriented transport layer takes care of acknowledgement, error recovery and flow control whereas connection less transport layer does not provide all these.

5. Session Layer : The service provided by the first three layers are not sufficient for some processes. The session layer is the network dialog controller. It establishes, maintains and synchronize the interaction between communicating system.

- **Dialog Control :** Session layer allows the communication between two processes to take place either in half-duplex or full duplex. For example, dialog between two servers may be full duplex and between client and server may be half duplex.
- **Synchronization :** Check points are added into stream of data bits known as synchronization points. For example if a system is sending a file of 1000 kb, it is advisable to insert check-points after every 100 kb to ensure that each 100 kb is received and acknowledged independently. In this case if a 10 kb crash happens during the transmission of data lying at position 420-430 kb, retransmission begins at 401 kb, data from initial to 400 kb is need not be retransmitted.
- **Token Management :** Session layer provides a token management service. For some protocols it is essential that both sides do not attempt the same operation at the same time. To manage these activities, the session layer provides token that can be exchanged only the side holding the token may perform the critical operation.

6. Presentation Layer : Functions of presentation layer are given below :

- **Encryption :** For communication of sensitive information a system must be able to assure privacy. The task of presentation layer is to encrypt transmission that must be secure. Encryption is one to one transformation of a message into an encrypted form.
- **Compression :** An important task of presentation layer is data compression. Such data compression eliminates some of the redundancy in the information to be transmitted, thereby reducing the number of bits to be transmitted.

- **Translation :** The process in two systems are usually exchanging information in the form of characteristics, numbers and so on. The information should be changed to bit streams before being transmitted. Because different computer use different encoding systems, the presentation layer is responsible for inter-operatability between these different encoding methods. The presentation layer at sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver dependent format.

7. Application Layer : The application layer provides frequently needed communication services such as file transfer, terminal emulation, remote login and directory service. These services are used by the user applications. For instance an e-mail program uses a file transfer service that delivers a file to a list of addresses and inform the sender if some address cannot be reached. Such applications run on the top of application layer.

Specific services of application layer are :

- **Electronic Mail Services :** This service facilitates user for e-mail, storage, receiving and forwarding services.
- **Directory Services :** This service facilities access for global information about various objects and services using distributed database source.
- **File Transfer, Access and Management :** This application allows user to access files from a remote computer, to manage files and to retrieve files from remote computer.
- **Network Virtual Terminal (NVT) :** It is software version of a physical terminal and allows a user to log on to a remote host.

Q.17 ***Draw and explain TCP/IP reference model in computer network communication.*** *[R.T.U. 2017]*

OR

Draw the following reference models used in computer communication:

(i) OSI/ISO Model

(ii) TCP/IP Model *[R.T.U. 2016]*

Ans. (i) OSI/ISO Model : *Refer to Q.16.*

(ii) TCP/IP Model : *Refer to Q.12.*

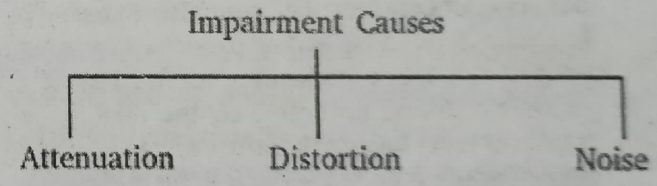
Q.18 ***What are lossless and lossy channels? Also explain transmission impairments in detail.*** *[R.T.U. 2017]*

Ans. Lossless and Lossy Channels : The Lossless channel is capable of reconstituting the original form of the data. The quality of the data is not compromised. This channel allows a signal to restore its original form. This type of channel is used for transmitting texts.

The Lossy channel eliminates some amount of data that is not noticeable. This technique does not allow a signal to restore in its original form but significantly reduces the size. The lossy channel is beneficial if the quality of the signal is not your priority. It slightly degrades the quality of the data but is convenient when one wants to send or store the data. This type of channel is used for organic data like audio signals and images.

Transmission Impairments : In communication system, analog signals travel through transmission media, which tends to deteriorate the quality of analog signal. This imperfection causes signal impairment. This means that received signal is not same as the signal that was send.

Causes of impairment



(a) Attenuation : It means loss of energy. The strength of signal decreases with increasing distance which causes loss of energy in overcoming resistance of medium. This is also known as attenuated signal. Amplifiers are used to amplify the attenuated signal which gives the original signal back.

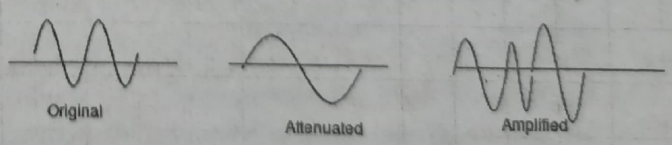


Fig.

Attenuation is measured in decibels(dB). It measures the relative strengths of two signals or one signal at two different point.

Attenuation(dB) = $10\log_{10}(P_2/P_1)$

where, P_1 is power at sending end and P_2 is power at receiving end.

(b) Distortion : It means change in the shape of signal. This is generally seen in composite signals with different frequencies. Each frequency component has its own propagation speed travelling through a medium. Every component arrive at different time which leads to delay distortion. Therefore, they have different phases at receiver end from what they had at senders end.

(c) Noise : The random or unwanted signal that mixes up with the original signal is called noise. There are several types of noise such as induced noise, crosstalk noise, thermal noise and impulse noise which may corrupt the signal.

(i) **Thermal Noise:** The thermal noise is due to thermal agitation of electrons in a conductor. It is distributed across the entire spectrum and that is why it is also known as white noise (as the frequency encompass over a broad range of frequencies).

(ii) **Intermodulation Noise:** When more than one signal share a single transmission medium, intermodulation noise is generated. For example, two signals f_1 and f_2 will generate signals of frequencies $(f_1 + f_2)$ and $(f_1 - f_2)$, which may interfere with the signals of the same frequencies sent by the transmitter. Intermodulation noise is introduced due to nonlinearity present in any part of the communication system.

(iii) **Cross Talk:** Cross talk is a result of bunching several conductors together in a single cable. Signal carrying wires generate electromagnetic radiation, which is induced on other conductors because of close proximity of the conductors. While using telephone, it is a common experience to hear conversation of other people in the background. This is known as cross talk.

(iv) **Impulse Noise:** Impulse noise is irregular pulses or noise spikes of short duration generated by phenomena like lightning, spark due to loose contact in electric circuits, etc. Impulse noise is a primary source of bit-errors in digital data communication. This kind of noise introduces burst errors.

Q.19 ***(a) Explain the layered architecture of OSI Model. What is the significance of OSI model? What is the significance of XDR(External Data Representation) at presentation layer.***

(b) Calculate the channel capacity of a telephone line having bandwidth 3000Hz in following cases:

(i) SNR = 3162

(ii) Noise is so strong that the signal is faint, means SNR is almost zero. ***[R.T.U. 2015]***

Ans.(a) Layered Architecture of OSI Model : *Refer to Q.4 and Q.16.*

Significance of OSI Model:

- It prevents changes in one layer from affecting other layers.
- It describes what functions occur at each layer of the model that encourages industry standardization.
- Dividing network communication process in smaller component makes software development, design, and troubleshooting easier.
- Standardization of network components allows multiple-vendor development.
- It allows different types of network hardware and software to communicate.
- Dividing network in layers make network administrators life easier. They can troubleshoot issue more quickly and effectually by looking in layer that is causing issue rather than finding it entire network. It also makes learning easier.

Significance of XDR :

There are many advantages in using a data-description language such as XDR versus using diagrams. Languages are more formal than diagrams and lead to less ambiguous descriptions of data. Languages are also easier to understand and allow one to think of other issues instead of the low-level details of bit-encoding. Also, there is a close analogy between the types of XDR and a high-level language such as C or Pascal. This makes the implementation of XDR encoding and decoding modules an easier task. Finally, the language specification itself is an ASCII string that can be passed from machine to machine to perform on-the-fly data interpretation. Apart from the above, XDR is faster in usage as compared to the alternative ASN technology.

Ans.(b) Shannon's Channel Capacity Criteria for Noisy Channels :

Given a communication channel with bandwidth of B Hz and a signal-to-noise ratio (SNR) of S/N, where S is the signal power and N is the noise power. Shannon's formula for the maximum channel capacity of such a channel is

$$C = B \log_2(1+S/N)$$

(i) Here, B = 3000 Hz

SNR = S/N = 3162

So, $C = 3000 * \log_2(1+3162)$

$= 3000 * 11.6270778405$

$= 34881.2335215$ Hz

(ii) Here, B = 3000 Hz

SNR = S/N = 0

So, $C = 3000 * \log_2(1+0) = 0$ Hz

Q.20 ***Draw the following reference models used in computer communication :***

(i) OSI/ISO Model

(ii) TCP/IP Model

Also give the key difference in both above models.

[R.T.U. 2014]

Ans. (i) Layers in OSI Model : *Refer to Q.4.*

(ii) TCP/IP Protocol : *Refer to Q.12.*

Comparison of OSI Reference Model and TCP/IP Reference Model

Following are some major differences between OSI Reference Model and TCP/IP Reference Model, with diagrammatic comparison below :

S. No.	OSI (Open System Interconnection)	TCP/IP (Transmission Control Protocol/ Internet Protocol)
1.	OSI provides layer functioning and also defines functions of all the layers.	TCP/IP model is more based on protocols and protocols are not flexible with other layers.
2.	In OSI model the transport layer guarantees the delivery of packets.	In TCP/IP model the transport layer does not guarantees delivery of packets.
3.	Follows horizontal approach.	Follows vertical approach.
4.	OSI model has a separate presentation layer.	TCP/IP does not have a separate presentation layer.
5.	OSI is a general model.	TCP/IP model cannot be used in any other application.
6.	Network layer of OSI model provide both connection oriented and connectionless service.	The Network layer in TCP/IP model provides connectionless service.
7.	OSI model has a problem of fitting the protocols in the model.	TCP/IP model does not fit any protocol.
8.	Protocols are hidden in OSI model and are easily replaced as the technology changes.	In TCP/IP replacing protocol is not easy.
9.	OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them.	In TCP/IP it is not clearly separated its services, interfaces and protocols.
10.	It has seven layers.	It has four layers.

Q.21 ***(a) Suppose a spectrum of a channel is between 3MHz and 4MHz and signal to noise ratio is 24 dB, compute how many signaling levels are required to achieve the reachable data rates. Also calculate the channel capacity.*** ***[R.T.U. 2014]***

(b) What are various transmission impairments? Explain in brief. ***[R.T.U. 2014]***

OR

What is transmission impairements? Explain.

[R.T.U. 2013, Raj. Univ. 2007]

Ans. (a) The spectrum of a communications channel is 3 MHz to 4 MHz. Signal-to-noise ratio is :

$SNR_{dB} = 24$ dB

(i) Find the channel capacity (in the presence of noise) :

$B = (4 - 3)$ MHz $= 1$ MHz

By definition,

$$SNR_{dB} = 10.\log_{10}(SNR)$$

Thus $24 = 10.\log_{10}(SNR)$

$2.4 = \log_{10}(SNR)$

Apply the inverse function for the log function, the exponential function base 10, to both sides :

$10^{2.4} = 10^{[\log 10(SNR)]}$

$10^{2.4} = SNR$

$SNR = 251$

Now we use the Shannon-Hartley law :

$C = B.\log_2(1+SNR)$

We use the change of base formula to find log base 2:

$C = 10^6 \log_2(252)$

$$\log_2(x) = \frac{\log_{10}(x)}{\log_{10}(2)}$$

$$C = 10^6 \frac{\log_{10}(252)}{\log_{10}(2)}$$

$C = 8(10^6)$bps

$C = 8$Mbps

(M stands for mega, million which is 10 to the sixth power)

(ii) The signal levels required to achieve the reachable data rate

Use the Nyquist formula :

$C = 2B.\log_2(M)$

$8(10^6) = 2(10^6).\log_2(M)$

$8 = 2.\log_2(M)$

$4 = \log_2(M)$

$M = 2^4$

= 16 signal levels are needed

Ans. (b) Transmission Impairment : With any communication system it must be recognized that the received signal will differ from the transmitted signal due to various transmission impairments. On any telecommunication link, there will be noise and distortion of some degrees as they will introduce random modifications in analog signals and bit errors in digital signal. Transmission impairments can be broadly classified in two categories as :

(i) Systematic Distortion : It is the distortion which occurs every time when we transmit a given signal over a given channel. Knowing the channel, we can predict what is going to occur. The pulses may always be distorted in a certain way. This type of distortion is something which might be possibly compensated electronically so that its effects are eliminated.

Table : Types of impairments which affect communication circuits

	Systematic Distortion (Static Impairments)	Fortuitous Distortion (Transient Impairments)
1.	Loss	White noise
2.	Attenuation distortion	Impulse noise
3.	Delay distortion	Cross talk
4.	Harmonic distortion	Intermediation noise
5.	Frequency offset	Echoes
6.	Bias distortion	Changes in amplitude
7.	Characteristic distortion	Line outages Radio fading

(ii) Fortuitous Distortion : It is something which occurs at random, so it is not predictable except in terms of probability. Fortuitous distortion refers to transient impairments rather than continuing conditions on the line. Fortuitous distortion is more difficult to compensate for, though steps can be taken to minimize its effects and repair the damage it does. This type of distortion occasionally produce an extra large noise, burst or impulse which destroys or creates one more bits at random.

Different fortuitous and systematic distortion are given below :

(a) White Noise : White noise is a mathematical concept. It is called 'White' because it contains all spectral frequencies equally on average, just as white light contains all the colours of the rainbow equally. This is the random hiss that forms a background to all electric signalling. It cannot be removed so it sets a theoretical maximum on the performance of any communication link and on the various modulation methods. The amplitude of the signal after attenuation must be kept sufficiently for above the white noise background to prevent an excess of hiss on radio or telephone circuits or an excess of errors in data transmission.

(b) Impulse Noise : Impulse noise is generally only a minor annoyance for analog data. For example, voice transmission may be corrupted by short clicks and crackels with no loss of intelligibility. Impulse noise is the primary source of error in digital data communication.

(c) Crosstalk : Crosstalk has been experienced by anyone who, while using the telephone, has been able to hear another conversation. It is an unwanted coupling between signal paths. It can occur by electrical coupling between nearby twisted pair or rarely, coaxial cable lines carrying multiple signals. Crosstalk is of the same order of magnitude (or less) as thermal noise.

(d) Intermodulation Noise : Intermodulation noise is produced when there is some non-linearity in the transmitter, receiver or intervening transmission system. In a non-linear system output is complex function of the input. Such non-linearity can be caused by component malfunction or the use of excessive signal strength. It is under these circumstances that the sum and difference terms occur.

(e) Echoes : Echoes on transmission lines are similar to crosstalk in their effects on data transmission. Where there is a change in impedance on the transmission line, a signal will be reflected so that it travels back down the line at reduced amplitude, thus forming an echo. This signal to echo power ratio can occassionally becomes less than 15 dB though it rarely falls below 10dB. It can be greater than white noise or crosstalk.

(f) Sudden Changes in Amplitude : Sometimes the amplitude of the signal changes suddenly. This may

be due to faults in amplifiers, unclean contacts with variable resistance, added load, new circuits switched in, maintenance work in progress or the switching to a different transmission path. These sudden changes can have an effect on certain data transmission systems, which could result in the loss or addition of a bit. The effect they have depend on the type of modem in use.

(g) Radio Fading : Radio links are subject to fading. Many long distance telephone circuits travel over microwave paths and fading sometimes occurs. Small fades are compensated by the ratio automatic gain control. Large fades may cause a serious degradation of the signal to noise ratio. Heavy rain and snow may cause fading.

(h) Changes in Phase : The phase of the transmitted signal sometimes changes. Some impulse noise cause both attenuations and phase transients. Brief phase changes are imperceptible in speech but sometimes cause data errors, especially when phase modulation is employed.

(i) Loss : The loss of signal strength on circuit is typically about 16dB. It may vary because of aging equipment, amplifier drift, temperature changes and other causes. Changes are adjusted during routine maintenance.

(j) Attenuation Distortion : The strength of a signal falls off with distance over any transmission medium. For guided media, this reduction in strength or attenuation is logarithmic and thus typically expressed as a constant number of decibels per unit distance. For unguided media, attenuation is a more complex function of distance and of the make-up of the atmosphere.

Attenuation is an increasing function of frequency. Fig.1 shows attenuation is a function of frequency for a typical leased line. In this fig.1 attenuation is measured relative to the attenuation at 1000 Hz.

The solid line shows equalization as can be seen, frequency components at the upper end of the voice band are attenuated much more than those at lower frequencies.The dashed line shows the effect of equalization. The flattened response curve improves the quality of voice signals. It also allows higher data rates to be used for digital data that are passed through a modem.

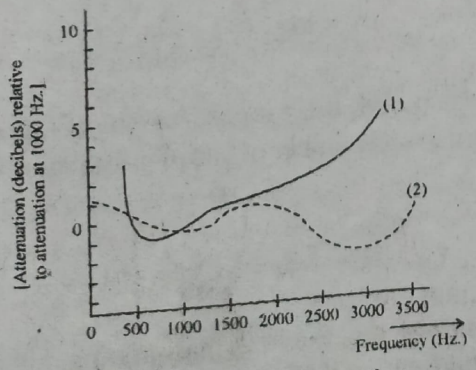


Fig. 1 : Attenuation distortion

(k) Delay distortion : The phase of the signal also is not transmitted linearly. The signal is delayed more at some frequencies than at other. This is referred to as **phase-frequency distortion** or **delay distortion.**

This phenomenon is peculiar to guided transmission media. Signals on wire pairs are propagated with some different speeds at different frequencies. Delay distortion is particularly critical for digital data. Consider that sequence of bits is being transmitted, using either analog or digital signals. Because of delay distortion, some of the signal components of one bit position will spill over into other bit position, causing inter-symbol interference, which is a major limitation to maximum bit rate over a transmission control.

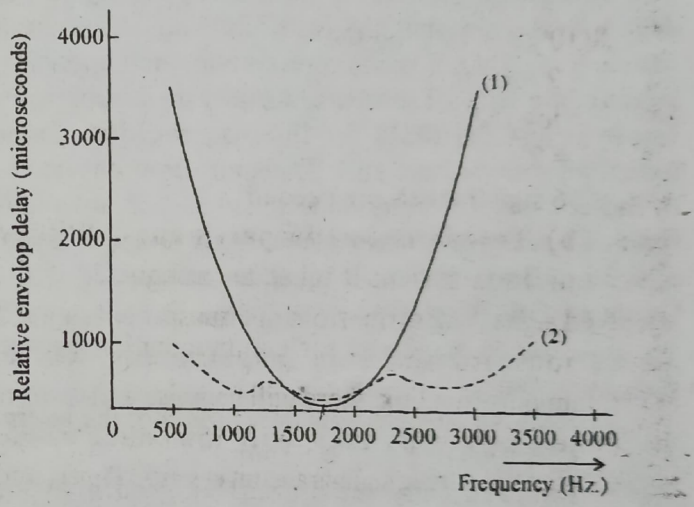


Fig. 2 : Delay distortion

Equalization technique can also be used for delay distortion. In fig.2 shown, dashed line shows effect of equalization. Envelop delay is defined as rate of change of phase with respect to frequency of received signal.

Q.22 ***(a) Give a detailed description of OSI model ?***
(b) What is the difference between OSI model & TCP/IP model?

[R.T.U. 2013, Raj. Univ. 2007]

Ans.(a) OSI Model : An ISO standard that covers all aspects of network communication is Open System Interconnection (OSI) model. An Open system is a model that allows any two different systems to communicate regardless of their underlying architecture. The OSI model is not a protocol, it is a model for understanding and designing a network architecture that is flexible, robust and interoperable. OSI reference model serves as frame work for standards which can be implemented at each layer by a variety of protocols. The OSI model consists of seven separate but related layers each of which defines a segment of the process of moving information across a network.

Characteristics of the OSI Layers

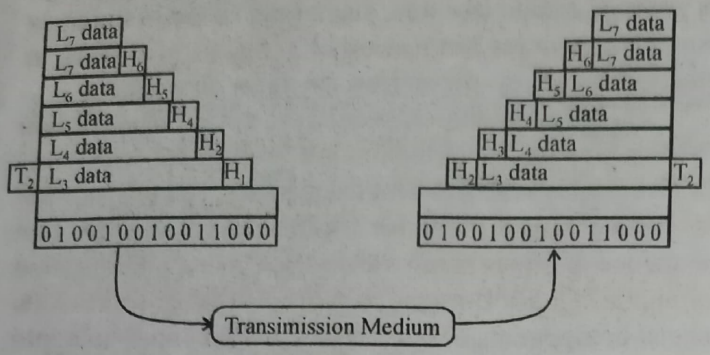


Fig. : Data movement using OSI model

The seven layers of the OSI reference model can be divided into three categories. Layers 1, 2 and 3 physical, data link and network are the network support layers. They deal with the physical aspects of moving data from one device to another. Session, presentation and application layers comprising of software, are the user support layers. These layers facilitate for inter-operatibility among unrelated software systems. Transport layer ensures end to end delivery. Information is added at each layer in the form of headers or trailers. Headers are added at layer 6, 5, 4, 3 and 2 whereas trailer is added at layer 2.

Goals of OSI Model

1. Provide standards for communication between systems.
2. Remove concern with description of the internal operation of a single system.
3. Remove any technical impediment to communication between systems.
4. Define the points of inter-connection for the exchange of information between systems.
5. Reduce the alternate in order to increase the ability to communicate without expensive conversions and translation between products.

Functions of the Layers : *Refer to Q.16.*

Ans.(b) Difference between OSI Model and TCP/IP Model : *Refer to Q.20.*

Q.23 ***What is channel capacity? Discuss Shannon capacity formula. Suppose there is spectrum of a channel 3 MHz & SNR_{dB} = 24dB then find out its capacity & How many signaling levels are required.*** *[R.T.U. 2013]*

Ans. Channel Capacity : There are a variety of impairments that distort or corrupt a signal. For digital data, the question that then arises is to what extent these impairments limit the data rate that can be achieved. The rate at which data can be transmitted over a given communication path, or channel, under given conditions, is referred to as channel capacity.

Shannon's Equation : The signal and noise are uncorrelated that is, they are not related in any way which would let us predict one of them from the other. The total power obtained, P_T, when combining these uncorrelated, apparently randomly varying quantities is given by

$$P_T = S + N$$

i.e. the typical combined rms voltage, V_T, will be such that

$$V_T^2 = V_S^2 + V_N^2$$

Since the signal and noise are statically similar their combination will have the same form factor value as the signal or noise taken by itself. We can therefore expect that the combined signal and noise will generally be confined to a voltage range $\pm\eta\ V_T$.

Consider now dividing this range into 2^b bands of equal size. (*i.e.* each of these bands will cover $\Delta V = 2\eta\ V_T/2^b$) To provide a different label for each band we require 2^b symbols or numbers. We can therefore always indicate which band the voltage level occupies at any moment in terms of a b-bit binary number. In effect, this process is another way of describing what happens when we take digital samples with a b-bit analog to digital converter working over a total range $2V_T$.

There is no real point in choosing a value for b which is so large that ΔV is smaller than $2\eta V_N$. This is because the noise will simply tend to randomize the actual voltage by this amount, making any extra bits meaningless. As a result the maximum number of bits of information we can obtain regarding the level at any moment will given by

$$2^b = \frac{V_T}{V_N}$$

i.e.

$$2^b = \sqrt{\frac{V_T^2}{V_N^2}} = \sqrt{\left(\frac{V_N^2}{V_N^2}\right) + \left(\frac{V_S^2}{V_N^2}\right)}$$

$$= \sqrt{1 + (S/N)}$$

which can be rearranged to produce

$$b = \log_2 \left\{ \left(1 + \frac{S}{N}\right)^{1/2} \right\}$$

If we make M, b-bit measurement of the level in a time, T, then the total number of bits of information collected will be

$$H = Mb = M.\log_2 \left\{ \left(1 + \frac{S}{N}\right)^{1/2} \right\}$$

This means that the information transmission rate, I, bits per unit time, will be

$$I = \left(\frac{M}{T}\right)\log_2\left\{\left(1+\frac{S}{N}\right)^{1/2}\right\}$$

From the Sampling Theorem we can say that, for a channel of bandwidth, B, the highest practical sampling rate, M/T, at which we can make independent measurement or samples of a signal will be

$$\frac{M}{T} = 2B$$

Combining expression we can therefore conclude that the maximum information transmission rate, C, will be

$$C = 2B\log_2\left\{\left(1+\frac{S}{N}\right)^{1/2}\right\}$$

$$= B\log_2\left\{1+\frac{S}{N}\right\}$$

This expression represents the maximum possible rate of information transmission through a given channel or system. The maximum rate we can transmit information is set by the bandwidth, the signal level, and the noise level. C is therefore called the channel's information carrying Capacity.

Spectrum of a Channel

We know that capacity is

$$C = B\log_2(1+SNR)$$

$$B = 3\ MHz$$

$$SNR_{dB} = 24dB$$

$$SNR = 10^{2.4}$$

$$C = 3\times10^6\ Log_2(1+10^{2.4})\ bps$$

$$= 3\times10^6\times8\ bps$$

$$= 24\ Mbps$$

Maximum number of bit or signaling level

$$2^b = \frac{V_T}{V_N} \text{ or } 2^b = \sqrt{(1+SNR)}$$

$$(SNR)_{dB} = 24dB$$

$$(SNR)_{dB} = 24dB$$

$$Log_{10}SNR = 1.2$$

$$SNR = 15.85$$

$$2^b = \sqrt{1+15.852^b} = 4.104$$

$$b\log_{10}^2 = \log_{10}^{4.104}$$

$$b = 2$$

Q.24 ***What are the various transmission media? Explain guided media in detail.*** *[R.T.U. 2013]*

Ans. Transmission Media: Transmission media is the physical path between transmitter and receiver in a data transmission system. Transmission media can be classified as guided or unguided. In both cases communication is in the form of electromagnetic waves. With guided media, the waves are guided along a solid medium, such as copper twisted pair, copper coaxial cable and optical fiber. The atmosphere and other space are examples of unguided media that provide a means of transmitting electromagnetic signals but do not guide them this form of transmission is usually referred to as wireless transmission.

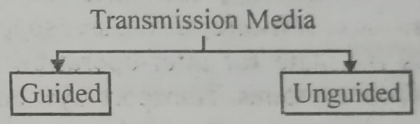


Fig.

Guided Media: A signal travelling along guided media is directed and contained by the physical limits of the medium. The transmission capacity in terms of bandwidth on data rate depends on material of media used and whether medium is point to point or multipoint.

Different categories of guided media are

- Twisted part cable
- Coaxial cable
- Fiber optic cable

Table : Point to point transmission characteristics of guided media

Transmission medium rate	Total data Spacing	Bandwidth	Repeater
Twisted Pair	4 Mbps	3 MHz	2 to 10 Km
Coaxial Cable	500 Mbps	350 MHz	1 to 10 Km
Optical Fiber	2 Gbps	2 GHz	10 to 100 Km

Twisted Pair Cable: Twisted pair cable comes in two forms: unshielded and shielded

1. Unshielded Twisted Pair (UTP) Cables: UTP cable is the most common type of telecommunication medium used today. Its frequency range is suitable for transmitting both data and voice. A twisted pairs consists of two conductors (usually copper), each with its own coloured plastic insulation. The plastic insulation is colour banded for identification as shown in fig. Colours are used both to identify the specific conductors in a cable and to indicate which wires belong in pairs and how they relate to other pairs in a larger bundle.

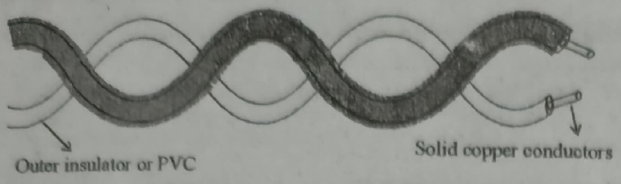


Fig. : Twisted pair cable

Two parallel flat wires were used for communication. However electromagnetic interference from devices such as a motor can create noise over those wires. If the two wires are parallel, the wire nearest to the source of the noise gets more interference and ends up with a higher voltage level than the wire further away, which results in an uneven load and a damaged signal as shown in fig.

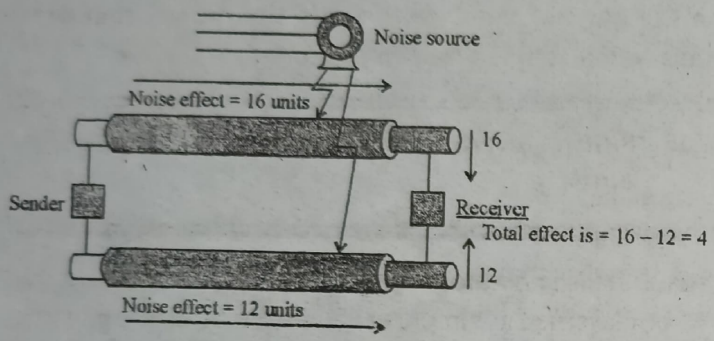


Fig. : Effect of noise on two parallel flat wires

2. Shielded Twisted Pair (STP) Cable: STP cable has a metal foil or braided mesh covering that encases each pair of insulated conductors as shown in fig. The metal casing prevents the penetration of electromagnetic noise. It also can eliminate crosstalk.

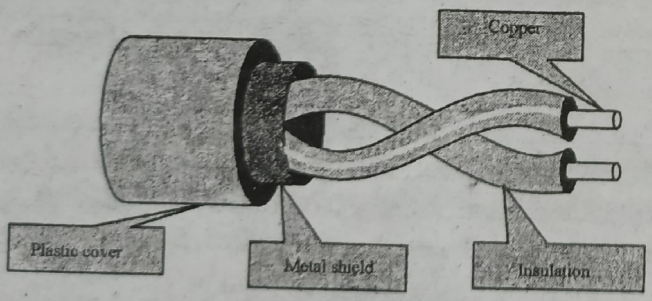


Fig. : Shielded twisted pair

STP has the same quality considerations and uses the same connections as UTP, but the shield must be connected in a ground. Materials and manufacturing requirements make STP more expensive than UTP but less susceptible to noise.

Coaxial Cable: High frequency electrical current flows in the outer skin of a conductor, making twisted pair and multicore cables inefficient. This skin effect in metal conductors increases attenuation with the square root of frequency. A coaxial cables surrounds the inner conductor with a dielectric such as poly ethylene and a coaxial tube of solid or braided metal surrounds the dielectric. Coaxial cable like twisted pair, consists of two conductors, but is constructed differently to permit it to operate over a wider range of frequencies. It consists of a hollow outer cylindrical conductor that surrounds a single inner wire conductor as shown in fig.

The inner conductor is held in place by either regularly spaced insulating rings or a solid dielectric material. The outer conductor is covered with a jacket or shield. A single coaxial cable has a diameter of from 0.4 to about 1 inch because of its shielded, concentric construction, coaxial cable is much less susceptible to interference and crosstalk than twisted pair. Coaxial cable be used over longer distances and supports more stations on a shared line than twisted pair.

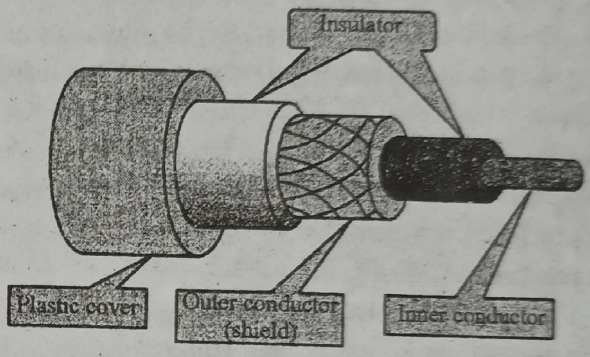


Fig. : Coaxial cable

Applications: Coaxial cable is the most versatile transmission medium and is enjoying widespread use in a wide variety of applications. The most important of these are :

- Television distribution.
- Long distance telephone transmission.
- Short run computer system links.
- Local area networks.

Coaxial cable is spreading rapidly as means of distributing TV signals to individual home cable TV. From its modest beginnings as Community Antenna Television (CATV), designed to provide service to remote area, cable TV will eventually reach almost as many homes and offices as the telephone. A cable TV system can carry dozens or even hundreds of TV channels at ranges up to a few tens of miles.

Coaxial cable is also used short range connections between devices. Using digital signaling, coaxial cable can be used to provide high speed I/O channels on computer systems. Another application area for coaxial cable is local area network. Coaxial cable can support a large number of devices with a variety of data and traffic types, over distances that include a single building or group of buildings.

Optical Fiber : It is made of glass or plastic and transmits signals in the form of light. Transmission of light signal in optical fiber is based on the phenomenon of total internal reflection. An optical fiber cable has a cylindrical shape and consists of three concentric sections : the core, the cladding and the jacket.

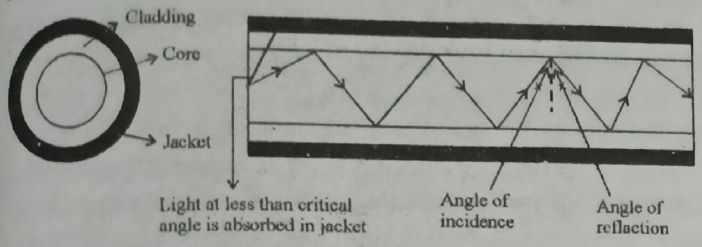


Fig. : Propagation of light in optical fiber

The core is the innermost section and consists of one or more very thin strands or fibers made of glass or plastic. Each fiber is surrounded by its own cladding, a glass or plastic coating that has optical properties different from those of the core. The outermost layer, surrounding one or a bundle of cladded fibers, is the jacket. The jacket is composed of plastic and other material layered to protect against moisture, abrasion, crushing, and other environmental dangers.

Propagation Modes: There are two mode of propagation of light in optical fibers: *multimode* and *single mode*. Each mode requires fiber with different characteristics. Multimode can be implemented in two forms : *step index or graded index.*

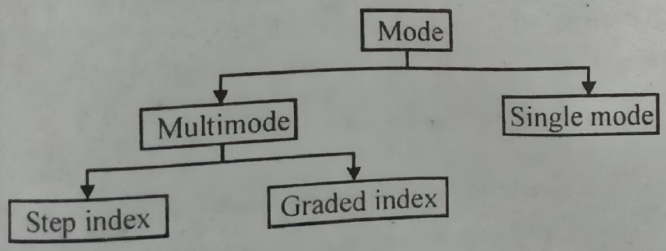


Fig.

Multimode: Multimode is so named because multiple beams from a light source move through the core in different paths. How these beams move within the cable depends on the structure of the core.

Multimode Step Index: In multimode step index fiber, the density of the core remains constant from the center to the edges. A beam of light moves through this constant density in a straight line until it reaches the interface of the core and the cladding. At the interface, there is an abrupt change to a lower density that alters the angle of the beam's motion. The term step index refers to the suddenness of this change.

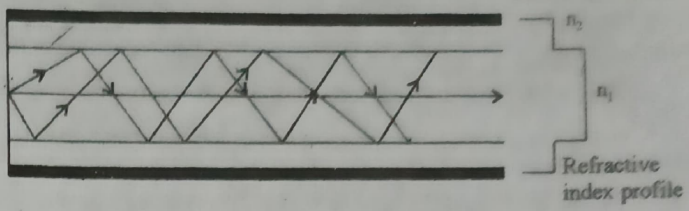


Fig. : Propagation in multimode step-index fiber

The light ray travelling the straight path through the centre and reaches the receiving end before the other rays, which follow a zigzag path. This difference in path length means that different beams arrive at the destination, at different times. As these different beams are recombined at the receiver, they result in a signal that is no longer an exact replica of the signal that was transmitted. This in known as *modal dispersion.*

Q.25 ***What is periodic analog signals? Explain in detail.***

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

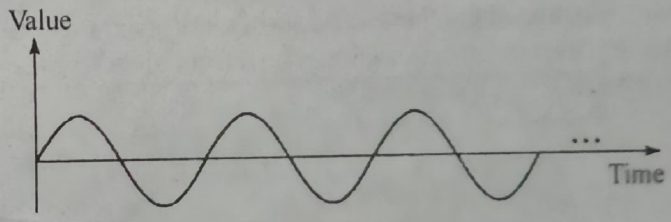


Fig. : A sine wave

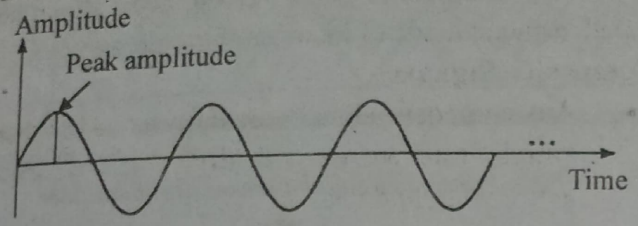
A sine wave can be represented by three parameter: the peak amplitude, the frequency, and the phase. These three parameters fully describe a sine wave.

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

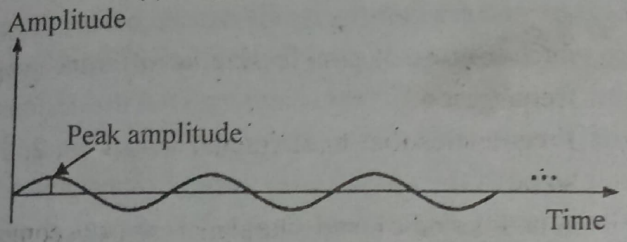
Period and Frequency : Period refers to the amount of time, in seconds, a signal needs to complete 1 cycle. Frequency refers to the number of periods in 1 s.

Phase : The term phase describes the position of the waveform relative to time 0. Phase is measured in degrees or radiant.

Two signals with the same phase and frequency, but different amplitudes



(a) A signal with peak amplitude



(b) A signal with low peak amplitude

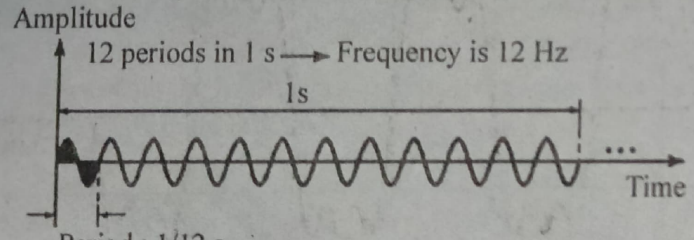
Fig.

Note : Frequency and period are the inverse of each other.

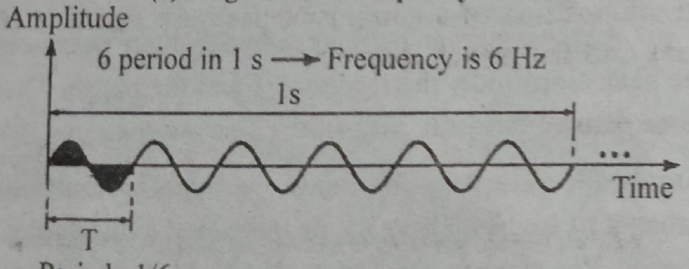
$$f = \frac{1}{T}$$

and $$T = \frac{1}{f}$$

Two signals with the same amplitude and phase, but different frequencies.



(a) A signal with a frequency of 12 Hz



(b) A signal with a frequency of 6 Hz

Fig.

Units of period and frequency.

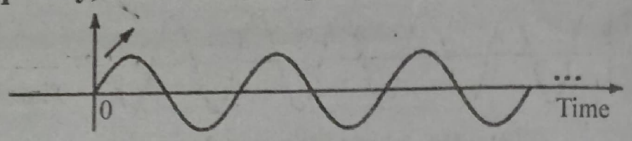
Unit	Equivalent	Unit	Equivalent
Seconds (s)	1 s	Hertz (Hz)	1 Hz
Milliseconds (ms)	10^{-3} s	Kilohertz (kHz)	10^{3} Hz
Microseconds (μs)	10^{-6} s	Megahertz (MHz)	10^{6} Hz
Nanoseconds (ns)	10^{-9} s	Gigahertz (GHz)	10^{9} Hz
Picoseconds (ps)	10^{-12} s	Terahertz (THz)	10^{12} Hz

Note: Frequency is the rate of change with respect to time. Change in a short span of time means high frequency. Change over a long span of time means low frequency.

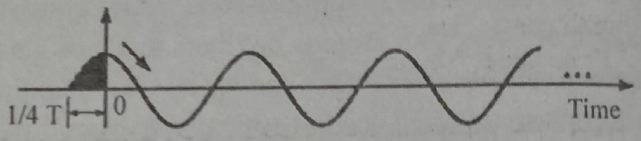
Note: If a signal does not change at all, its frequency is zero. If a signal changes instantaneously, its frequency is infinite.

Note: Phase describes the position of the waveform relative to time 0.

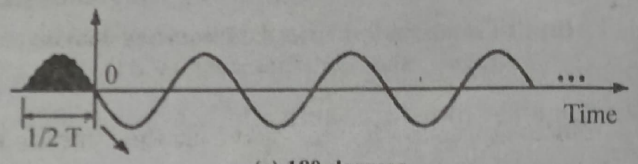
Three sine waves with the same amplitude and frequency, but different phases



(a) 0 degrees



(b) 90 degrees



(c) 180 degrees

Fig.

Wavelength : Wavelength is another characteristic of a signal traveling through a transmission medium. Wavelength binds the period or the frequency of a simple sine wave to the propagation speed of the medium. The wavelength is the distance a simple signal can travel in one period.

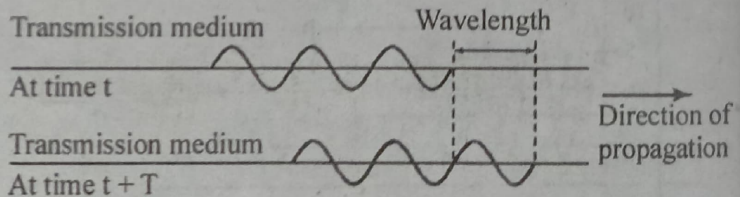


Fig.

Wavelength can be calculated if one is given the propagation speed (the speed of light) and the period/ frequency of the signal. We represent wavelength by A, propagation speed by c (speed of light), and frequency by 1, we get

$$\text{Wavelength} = \text{propagation speed} \times \text{period}$$

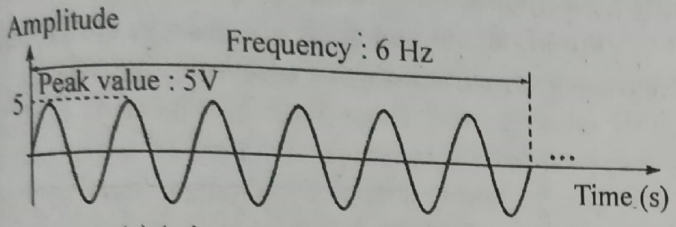
$$= \frac{\text{propagation speed}}{\text{frequency}}$$

Time and Frequency Domains

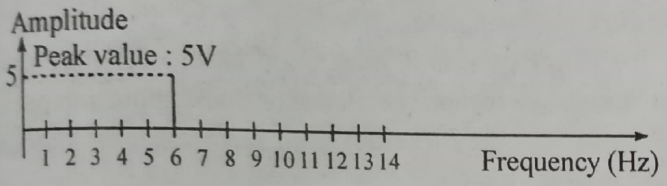
- To show the relationship between amplitude and frequency, we can use what is called a frequency domain plot.

- A frequency-domain plot is concerned with only the peak value and the frequency. Changes of amplitude during one period are not shown.
- Figure shows a signal in both the time and frequency domains.

The time-domain and frequency-domain plots of a sine wave.



(a) A sine wave in the time domain (peak value : 5V, frequency : 6 Hz)

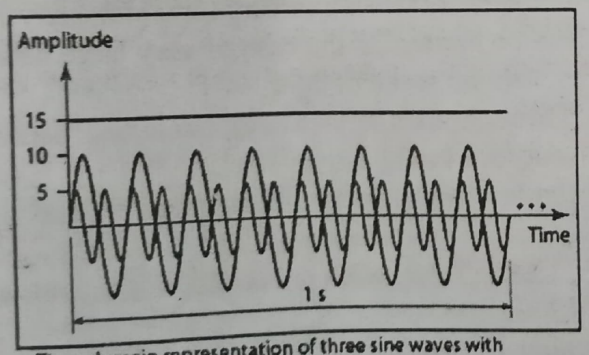


(b) The same sine wave in the frequency domain (peak value : 5V, frequency : 6 Hz)

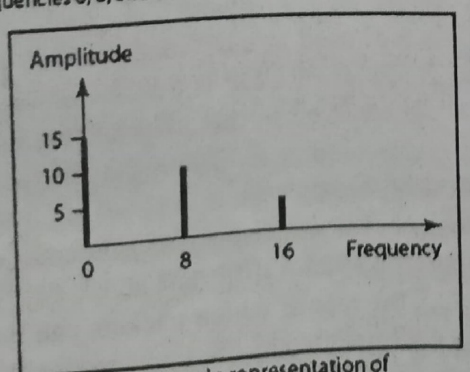
Fig.

Note: A complete sine wave in the time domain can be represented by one single spike in the frequency domain.

The time-domain and frequency-domain of three sine waves



a. Time-domain representation of three sine waves with frequencies 0, 8, and 16



b. Frequency-domain representation of the same three signals

Fig.

Note: A single-frequency sine wave is not useful in data communications; we need to send a composite signal, a signal made of many simple sine waves.

Composite Signals

- Any composite signal is actually a combination of simple sine waves with different frequencies, amplitudes, and phases.
- A composite signal can be periodic or non-periodic.
- A periodic composite signal can be decomposed into a series of simple sine waves with discrete frequencies.
- Frequencies that have integer values (1, 2, 3 and so on).
- A non-periodic composite signal can be decomposed into combination of an infinite number of simple sine waves with continuous frequencies, frequencies that have real values.

Note: If the composite signal is periodic, the decomposition gives a series of signals with discrete frequencies; if the composite signal is non-periodic, the decomposition gives a combination of sine waves with continuous frequencies.

A composite periodic signal

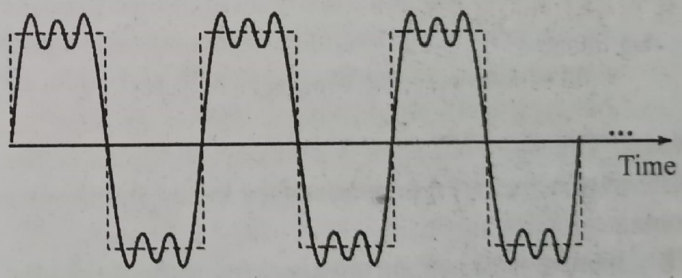
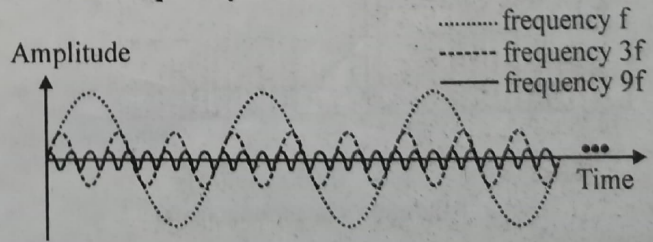
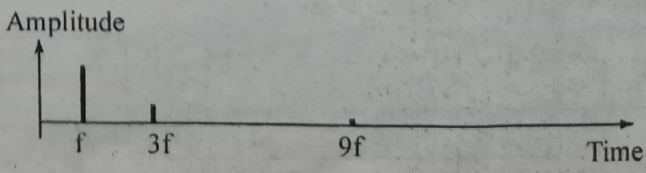


Fig.

Decomposition of a composite periodic signal in the time and frequency domains



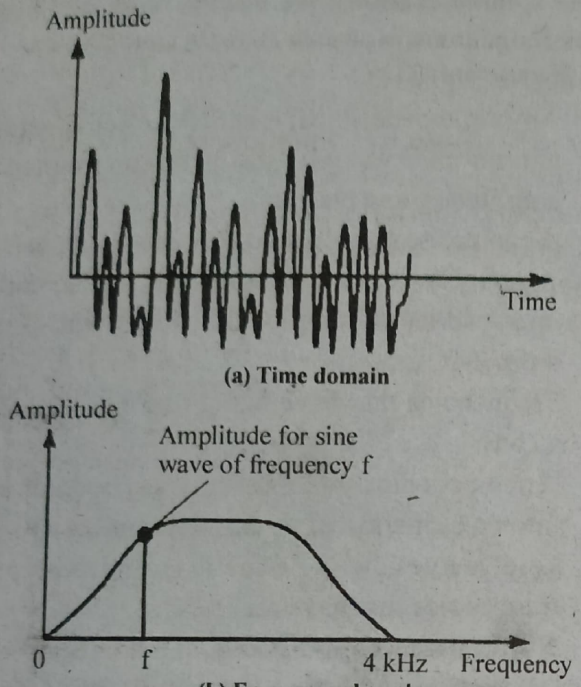
(a) Time-domain decomposition of a composite signal



(b) Frequency-domain decomposition of the composite signal

Fig.

The time and frequency domains of a nonperiodic signal



(a) Time domain

(b) Frequency domain

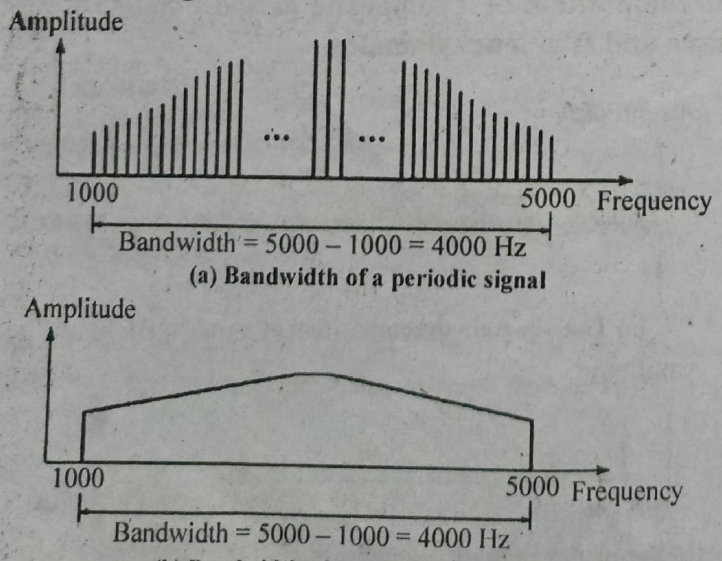
Fig.

Bandwidth

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

Note: The bandwidth of a composite signal is the difference between the highest and the lowest frequencies contained in that signal

The bandwidth of periodic and non-periodic composite signals.



(a) Bandwidth of a periodic signal

(b) Bandwidth of a nonperiodic signal

Fig.

Q.26 ***Write a detailed note on performance measurement.***

Ans. When troubleshooting network degradation or outage, measuring network performance is critical to determining when the network is slow and what is the root cause (e.g., saturation, bandwidth outage, misconfiguration, network device defect, etc).

Whatever the approach you take to the problem (traffic capture with network analyzers like Wireshark, SNMP polling with tools such as PRTG or cacti or generating traffic and active testing with tools such as smoke ping or simple ping or trace route to track network response times), indicators are usually called metrics and are aimed at producing tangible figures when measuring network performance.

This explains 3 major indicators for measuring network performance (i.e., latency, throughput and packet loss) and how they interact with each other in TCP and UDP traffic streams.

- **Latency** is the time required to transmit a packet across a network :
  - o Latency may be measured in many different ways : round trip, one way, etc.
  - o Latency may be impacted by any element in the chain which is used to transmit data: workstation, WAN link, routers, local area network (LAN), server,... and ultimately it may be limited, in the case of very large network, by the speed of light.
- **Throughput** is defined as the quantity of data being sent/received by unit of time.
- **Packet loss** reflects the number of packets lost per 100 packets sent by a host.

This can help you understand the mechanisms of network slowdowns.

Measuring Network Performance: UDP

UDP Throughput is not impacted by latency

UDP is a protocol used to carry data over IP networks. One of the principles of UDP is that we assume that all packets sent are received by the other party (or such kind of controls is executed at a different layer, for example by the application itself.)

In theory or for some specific protocols (where no control is undertaken at a different layer; e.g., one-way transmissions), the rate at which packets can be sent by the sender is not impacted by the time required to deliver the packets to the other party (= latency). Whatever that time is the sender will send a given number of packets

per second, which depends on other factors (application, operating system, resources, ...).

Measuring Network performance : TCP

TCP is directly impacted by latency

TCP is a more complex protocol as it integrates a mechanism which checks that all packets are correctly delivered. This mechanism is called acknowledgment: it consists of having the receiver transmit a specific packet or flag to the sender to confirm the proper reception of a packet.

TCP Congestion Window

For efficiency purposes, not all packets will be acknowledged one by one; the sender does not wait for each acknowledgment before sending new packets, indeed, the number of packets that may be sent before receiving the corresponding acknowledgement packet is managed by a value called TCP congestion window.

How the TCP congestion window impacts throughput

If we make the hypothesis that no packet gets loss the sender will send the first quota of packets (corresponding to the TCP congestion window) and when it will receive the acknowledgment packet, it will increase the TCP congestion window; progressively the number of packets that can be sent in a given period of time will increase (throughput). The delay before acknowledgment packets are received (= latency) will have an impact on how fast the TCP congestion window increases (hence the throughput).

When latency is high, it means that the sender spends more time idle (not sending any new packets), which reduces how fast throughput grows.

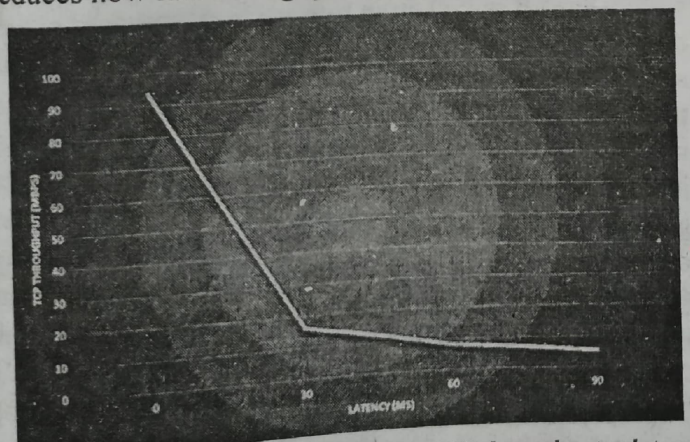


Fig. Measuring network performance - TCP throughput vs latency

Round trip late	TCP throughput
0ms	93.5 Mbps
30ms	16.2 Mbps
60ms	8.07 Mbps
90ms	5.32 Mbps

Q.27 ***Explain protocol and standards in detail.***

Ans. Protocols : In information technology, a protocol (from the Greek protocollon, which was a leaf of paper glued to a manuscript volume, describing its contents) is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols exist at several levels in a telecommunication connection. For example, there are protocols for the data interchange at the hardware device level and protocols for data interchange at the application program level. In the standard model known as Open Systems Interconnection (OSI), there are one or more protocols attach layer in the telecommunication exchange that both ends of the exchange must recognize and observe. Protocols are often described in an industry or international standard.

Standards : A common set of rules.

Standards Organization : Standards creation communities

IEEE (Institute of Electrical and Electronics Engineers)

IEEE's Constitution defines the purposes of the organization as "scientific and educational, directed toward the advancement of the theory and practice of Electrical, Electronics, Communications and Computer Engineering, as well as Computer Science, the allied branches of engineering and the related arts and sciences." The IEEE is incorporated under the Not-for-Profit Corporation Law of the state of New York, United States. It was formed in 1963 by the merger of the Institute of Radio Engineers (IRE, founded 1912) and the American Institute of Electrical Engineers (AIEE, founded 1884). It has more than 400,000 members in more than 160 countries, 45% outside the United States. In pursuing these goals, the IEEE serves as a major publisher of scientific Journals and a conference organizer. It is also a leading developer of industrial standards (having developed over 900 active industry standards) in a broad range of disciplines, including electric power and energy, biomedical technology and health care, information technology, information assurance. telecommunications, consumer electronics, transportation, aerospace, and nanotechnology. IEEE develops and participates in educational activities such as accreditation of electrical engineering programs in institutes of higher learning.

IEEE is one of the leading standards-making organizations in the world. IEEE performs its standards making and maintaining functions through the IEEE Standards Association (IEEE-SA). IEEE standards affect a wide range of industries including: power and energy,

biomedical and health care, Information Technology (IT), telecommunications, transportation, nanotechnology, information assurance, and many more. In 2005, IEEE had close to 900 active standards, with 500 standards under development. One of the more notable IEEE standards is the IEEE 802 LAN/MAN group of standards which includes the IEEE 802.3 Ethernet standard and the IEEE 802.11 Wireless Networking standard.

ANSI (American National Standards Institute) : Though ANSI itself does not develop standards, the Institute oversees the development and use of standards by accrediting the procedures of standards developing organizations. ANSI accreditation signifies that the procedures used by standards developing organizations meet the Institute's requirements for openness, balance, consensus, and due process.

ANSI was originally formed in 1918, when five engineering societies and three government agencies founded the American Engineering Standards Committee (AESC). In 1928, the AESC became the American Standards Association (ASA). In 1966, the ASA was reorganized and became the United States of America Standards Institute (USASI). The present name was adopted in 1969.Prior to 1918, these five engineering societies:

- American Institute of Electrical Engineers (AIEE, now IEEE)
- American Society of Mechanical Engineers (ASME)
- American Society of Civil Engineers (ASCE)
- American Institute of Mining Engineers (AIME, now American Institute of Mining, Metallurgical, and Petroleum Engineers)
- American Society for Testing and Materials (now ASTM International)

ANSI also designates specific standards as American National Standards, or ANS, when the Institute determines that the standards were developed in an environment that is equitable, accessible and responsive to the requirements of various stakeholders.

The American National Standards process involves:

- Consensus by a group that is open to representatives from all interested parties
- Broad-based public review and comment on draft standards
- Consideration of and response to comments
- Incorporation of submitted changes that meet the same consensus requirements into a draft standard
- Availability of an appeal by any participant alleging that these principles were not respected during the standards-development process.

ITU (International Telecommunications Union - formerly CCITT)

The International Telecommunication Union is the specialized agency of the United Nations which is responsible for information and communication technologies. ITU coordinates the shared global use of the radio spectrum, promotes international cooperation in assigning satellite orbits, works to improve telecommunication infrastructure in the developing world and establishes worldwide standards.

ITU coordinates the shared global use of the radio spectrum, promotes international cooperation in assigning satellite orbits, works to improve telecommunication infrastructure in the developing world and establishes worldwide standards. ITU also organizes worldwide and regional exhibitions and forums, such as ITU TELECOM WORLD, bringing together representatives of government and the telecommunications and ICT industry to exchange ideas, knowledge and technology. The ITU is active in areas including broadband Internet, latest-generation wireless technologies, aeronautical and maritime navigation, radio astronomy, satellite-based meteorology, convergence in fixed-mobile phone, Internet access, data, voice, TV broadcasting, and next-generation networks.

ISO (International Organization for Standards)

The International Organization for Standardization widely known as ISO, is an international standard-setting body composed of representatives from various national standards organizations. Founded on February 23, 1947, the organization promulgates worldwide proprietary industrial and commercial standards. It has its headquarters in Geneva, Switzerland. While ISO defines itself as a non-governmental organization, its ability to set standards that often become law, either through treaties or national standards, makes it more powerful than most non-governmental organizations. In practice, ISO acts as a consortium with strong links to governments ISO, is an international standard-setting body composed of representatives from various national standards organizations the organization promulgates worldwide proprietary industrial and commercial standards. ISO's main products are the International Standards. ISO also publishes Technical Reports, Technical Specifications. Publicly Available Specifications, Technical Corrigenda, and Guides .

EIA (Electronic Industries Association)

The Electronic Industries Alliance (EIA, until 1997 Electronic Industries Association) was a standards and trade organization composed as an alliance of trade associations for electronics manufacturers in the United States. They developed standards to ensure the equipment of different manufacturers was compatible and interchangeable. In 1924 the Associated Radio Manufacturers alliance was formed, which was renamed to Radio Manufacturers Association (RMA) the same year. Upcoming new electronic technologies brought new members and further name changes: Radio Television Manufacturers Association (RTMA) (1950), Radio Electronic Television Manufacturers (RETMA) (19 53) and Electronics Industries Association (EIA) (1957). The last renaming took place in 1997, when EIA became Electronics Industries Alliance (EIA), reflecting the change away from a pure manufacturers association A standard defining serial communication between computers and modems e. g. was originally drafted by the radio sector as RS-232. Later it was taken over by the EIA as EIA-232. Later this standard was managed by the TIA and the name was changed to the current TIA-232. Because the EIA was accredited by ANSI to help develop standards in its areas, the standards are often described as e. g. ANSI TIA-232 (or formerly as ANSI EIATIA-232').

ETSI (European Telecommunications Standards Institute)

The European Telecommunications Standards Institute (ETSI) is an independent, non-profit, standardization organization in the telecommunications industry (equipment makers and network operators) in Europe, with worldwide projection. ETSI has been successful in standardizing the Low Power Radio, Short Range Device, GSM cell phone system and the TETRA professional mobile radio system.

Significant ETSI standardisation bodies include TISPAN (for fixed networks and Internet machine-to-machine communications). ETSI inspired the creation of, and is a partner in 3GPP.

ETSI was created by CEPT in 1988 and is officially recognized by the European Commission and the EFTA secretariat. Based in Sophia Antipolis (France), ETSI is officially responsible for standardization of Information and Communication Technologies (ICT) within Europe. These technologies include telecommunications, broadcasting and related areas such as intelligent transportation and medical electronics. ETSI has 740 members from 62 countries/provinces inside and outside Europe, including manufacturers, network operators, administrations, service providers, research bodies and users - in fact, all the key players in the ICT and M2M (for ETSI has been successful in standardizing the Low Power Radio, Short Range Device, GSMTETRA professional mobile radio system.

W3C - World Wide Web Consortium

The World Wide Web Consortium (W3C) is the main international standards organization World Wide Web (abbreviated WWW or W3).

Founded and headed by Tim Berners-Lee, the consortium is made up of member organizations which maintain full-time staff for the purpose of working together in the development of standards for the World Wide Web. As of 18 February 2011, the World Wide Web Consortium (W3C) has 322 members.

W3C also engages in education and outreach, develops software and serves as an open forum for discussion about the Web.

W3C was created to ensure compatibility and agreement among industry members in the adoption of new standards. Prior to its creation, incompatible versions of HTML were offered by different vendors, increasing the potential for inconsistency between web pages. The consortium was created to get all those vendors to agree on a set of core principles and components which would be supported by everyone.

❑❑❑

2 DATA LINK LAYER

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 ***Differentiate between single-bit error and burst error.*** *[R.T.U. 2019]*

Ans. Single-Bit error : As name suggest single-bit errors occur when a single bit gets changed during transmission of data due to interference in network communication.

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

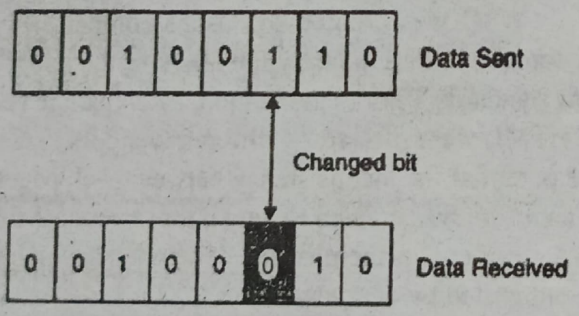


Fig. : Single Bit Error

Single-bit errors are least likely type of error because their duration or noise is normally longer than duration of 1 bit.

Burst Errors : When more than a single bit of data unit gets corrupted it is known as Burst error.

In comparison of single-bit errors, burst errors are more likely to occur. Because as we know that the duration of noise is generally longer than the duration of transferring 1bit, that means with longer duration noise can corrupt more than 1 bit easily. Number of bit affected depends on the data rate and duration of noise.

Q.2 ***Assume that, in a stop and wait ARQ system, the bandwidth of the line is 1 Mbps, and 1 bit takes 20ms to make a round trip. What is the bandwidth delay product? If the system data frames are 1000 bits in length, what is the utilization percentage of the link?*** *[R.T.U. 2015]*

Ans. The bandwidth-delay product is

$$1 \times 10^6 \times 20 \times 10^{-3} = 20{,}000 \text{ bits}$$

The system can send 20,000 bits during the time it takes for the data to go from the sender to the receiver and then back again. However, the system sends only 1000 bits. We can say that the link utilization is only 1000/20,000, or 5%. For this reason, for a link with high bandwidth or long delay, use of Stop-and-Wait ARQ wastes the capacity of the link.

Q.3 ***Can the value of a checksum be all 0's (in binary)? Defend your answer. Can the value be all 1's (in binary)? Defend your answer.*** *[R.T.U. 2015]*

Ans. The value of a checksum can be all 0's (in binary). This happens when the value of the sum (after wrapping) becomes all 1's (in binary).

It is almost impossible for the value of a checksum to be all 1's. For this to happen, the value of the sum (after wrapping) must be all 0's which means all data units must be 0's.

Q.4 ***Calculate the throughput for stop and wait flow control mechanism if the frame size is 4800 bits, bit rate is 9600 bps and distance between device is 2000 km. Speed of propagation over the transmission media is 2,00,000 km/s.***

[R.T.U. Dec. 2013]

Ans. The frame transmission time $= \frac{\text{Frame size}}{\text{Bit rate}}$

$$t_f = \frac{4800}{9600}$$

$$t_f = 0.5$$

$$\text{Propagation time} = \frac{2000}{200000}$$

$$t_p = 0.01$$

$$A = \frac{0.01}{0.5} = 0.02 \quad \left\{A = \frac{t_p}{t_f}\right\}$$

$$\eta = \frac{1}{1+2A}$$

$$= \frac{1}{1+2\times 0.02} = \frac{1}{1+0.04}$$

$$= 0.96$$

$$= 96\%$$

Q.5 ***A channel is operating at 4800 bps and propagation delay is 20 ms. What would be the minimum frame size for stop-and- wait flow control to get 50% link utilization efficiency?***

[R.T.U. Dec. 2013]

Ans. Bit rate R = 4800 bps

Propagation time $t_p = 20\,ms$

$$\eta = 50\%$$

Frame size N = ?

$$U_{max} = \frac{1}{1+2A}$$

$$0.5 = \frac{1}{1+2A}$$

$$1+2A = \frac{1}{0.5} = 2 \quad \left\{\because \quad A = \frac{t_p}{t_f}\right.$$

$$2A = 1$$

$$\frac{1}{2} = \frac{t_p}{t_f}$$

$$0.5t_f = t_p$$

$$t_f = \frac{L}{R}$$

$$t_f = \frac{20\times 10^{-3}}{0.5} = 0.04$$

$$0.04 = \frac{L}{4800}$$

$$L = 192\,bits$$

Q.6 ***Draw the sender and receiver window for a system using GO-Back-N-AKQ given the following:***

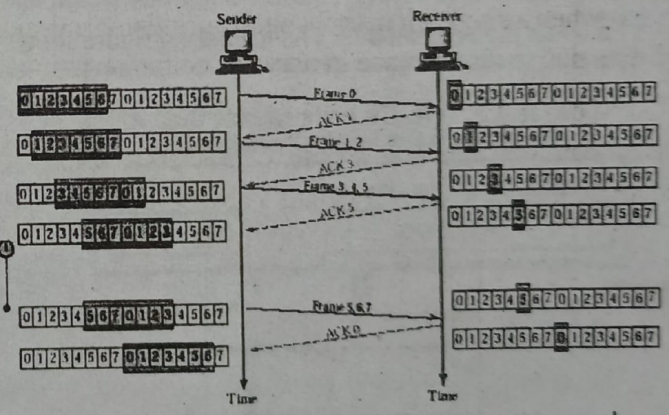
(i) Frame 0 is sent; frame 0 is acknowledged

(ii) Frame 1 and 2 are sent; frames 1 and 2 are acknowledged

(iii) Frames 3, 4 and 5 are sent; frame 4 is acknowledged; timer for frame 5 expires.

[R.T.U. Dec. 2013]

Ans.



PART-B

Q.7 ***Explain block coding with suitable diagrams.***

[R.T.U. 2019]

Ans. Block Coding : To ensure accuracy of the received data frame, redundant bits are used. For example, in even-parity, one parity bit is added to make the count of 1s in the frame even. In this way the original number of bits is increased. It is called block coding.

Block coding is represented by slash notation, mB/nB means, m-bit block is substituted with n-bit block where n > m. Block coding involves three steps:

- Division
- Substitution
- Combination.

Block Coding Concept

Block coding changes a block of m bits into a block of n bits, where n is larger than m. Block coding is referred to as an mB/nB encoding technique.

As block coding normally involves three steps: division, substitution and combination. In the division step, a sequence of bits is divided into groups of m bits.

For example, in 4B/5B encoding, the original bit sequence is divided into 4-bit groups. The heart of block coding is the substitution step. In this step, we substitute an m-bit group for an n-bit group.

For example, in 4B/5B encoding we substitute a 4-bit code for a 5-bit group. Finally, the n-bit groups are combined together to form a stream. The new stream has more bits than the original bits. The following figure shows the procedure.

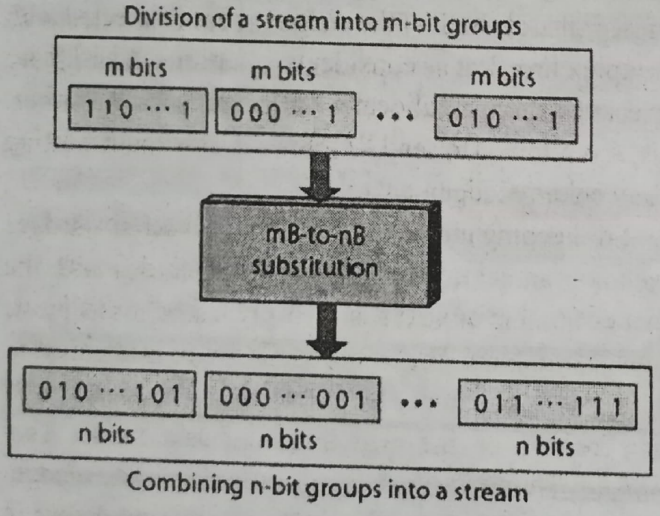


Fig.

Q.8 ***Explain Go-back-N ARQ protocol.*** *[R.T.U. 2019]*

Ans. Go-Back-N ARQ : Go-Back-N Automatic Repeat reQuest (Go-Back-N ARQ), is a data link layer protocol that uses a sliding window method for reliable and sequential delivery of data frames. It is a case of sliding window protocol having sending window size of N and receiving window size of 1.

Working Principle

Go – Back – N ARQ uses the concept of protocol pipelining, i.e. sending multiple frames before receiving the acknowledgment for the first frame. The frames are sequentially numbered upto a finite number of frames. The maximum number of frames that can be sent depends upon the size of the sending window. If the acknowledgment of a frame is not received within an agreed amount of time period, all frames starting from that frame are retransmitted.

The size of the sending window determines the sequence number of the outbound frames. If the sequence number of the frames is an n-bit field, then the range of sequence numbers that can be assigned is 0 to 2^{n-1}. Consequently, the size of the sending window is 2^{n-1}. Thus, in order to accommodate a sending window size of 2^{n-1}, an n-bit sequence number is chosen.

The sequence numbers are numbered as modulo-n. For example, if the size of sending window is 4, then the sequence numbers will be 0, 1, 2, 3, 0, 1, 2, 3, 0, 1, and so on. The number of bits in the sequence number is 2, to generate the binary sequence 00, 01, 10, 11.

The size of the receiving window is 1.

Sender Site Algorithm of Go-Back-N Protocol

```
begin
  frame s; //s denotes frame to be sent
  frame t; //t is temporary frame
  S_window = power(2,m) – 1; //Assign maximum window size
  SeqFirst = 0; // Sequence number of first frame in window
  SeqN = 0; // Sequence number of Nth frame window
  while (true) //check repeatedly
  do
   Wait_For_Event(); //wait for availability of packet
    if ( Event(Request_For_Transfer)) then
     //check if window is full
     if (SeqN–SeqFirst>= S_window) then
```

```
    doNothing();
  end if;
  Get_Data_From_Network_Layer();
  s = Make_Frame();
  s.seq = SeqN;
  Store_Copy_Frame(s);
  Send_Frame(s);
  Start_Timer(s);
  SeqN = SeqN + 1;
 end if;
 if ( Event(Frame_Arrival)) then
   r = Receive_Acknowledgement();
   if ( AckNo>SeqFirst&&AckNo<SeqN ) then
     while ( SeqFirst<= AckNo )
       Remove_copy_frame(s.seq(SeqFirst));
       SeqFirst = SeqFirst + 1;
     end while
     Stop_Timer(s);
   end if
 end if
 // Resend all frames if acknowledgement havn't
been received
 if ( Event(Time_Out)) then
   TempSeq = SeqFirst;
   while ( TempSeq<SeqN )
     t = Retrieve_Copy_Frame(s.seq(SeqFirst));
     Send_Frame(t);
     Start_Timer(t);
     TempSeq = TempSeq + 1;
   end while
  end if
 end
```

Receiver Site Algorithm of Go-Back-N Protocol

```
 Begin
   frame f;
   RSeqNo = 0; // Initialise sequence number of
expected frame
 while (true) //check repeatedly
 do
   Wait_For_Event(); //wait for arrival of frame
   if ( Event(Frame_Arrival) then
     Receive_Frame_From_Physical_Layer();
     if ( Corrupted ( f.SeqNo )
       doNothing();
     else if ( f.SeqNo = RSeqNo ) then
       Extract_Data();
       Deliver_Data_To_Network_Layer();
       RSeqNo = RSeqNo + 1;
       Send_ACK(RSeqNo);
     end if
   end if
 end while
 end
```

Q.9 ***Explain sliding window protocols.*** *[R.T.U. 2017]*

Ans. Sliding Window Protocol : In sliding window flow control protocol, both of the endpoints are connected with full duplex link. Let us consider two stations, A and B are connected stations. B allocates buffer space for W frames. Thus, A is allowed to send W frames to B without waiting for any acknowledgements.

For keeping into a track of frames, B acknowledges by sending an acknowledgement that includes with the sequence number of next frame (every frame has its frame sequence number). The acknowledgement is attached to the outgoing data frame. In effect, the acknowledgement gets a free rid on the next outgoing data frame. This technique of temporarily delaying outgoing acknowledgements, so that they can be hooked onto the next outgoing data frame is known as **piggy backing.**

The acknowledgement also implicitly announces that B is prepared to receive the next W frames, beginning with the number specified. For example B could receive frames 2, 3, 4 but with hold acknowledgement until frame 4 has arrived. By then returning an acknowledgement with sequence number 5, B acknowledges frames 2, 3 and 4 at one time.

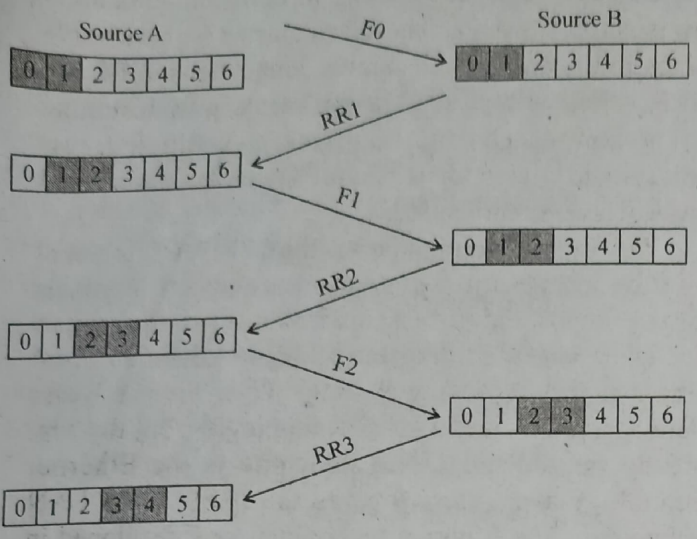


Fig. : One bit sliding window protocol by the time the RR reaches A, it already send the next frame

A records the list of sequence numbers that is allowed to transmit and B records a list of sequence number that is prepared to receive. Each of these lists can be thought as window of frames. Generally, if the sequence field is of K–bit then range of sequence numbers in 0 to $2^K - 1$ and frame are numbered modulo 2^K.

Example : Let us take an example of sliding window of 3 size 1, with a 2 bit sequence number. Initially A and B have windows indicating A, may transmit the frame with frame 0(F0). A maintains and copy of the transmitted frame. The window also indicate that A may transmit the next frame beginning with frame number 1. B then transmit an RR1 (Receive Ready) which means it (B) is ready to receive the frame number 1. With this, 4 is backup to permission to send the frame (F1); also 4 may discard the buffered frame that is now acknowledged. A proceeds to send frame 2.B returns RR2 which acknowledges F1 and allows the transmission of F2.

Q.10 ***A 1 km, 10 Mbps CSMA/CD LAN has a propagation speed of 200 m/μ sec. Data frames are 256 bits long, including 32 bits of header, check sum and other overhead for the receiver to capture the channel to send a 32-bit acknowledgement frame. What is the effective data rate, excluding overhead, assuming there are no collisions ?*** *[R.T.U. 2016, R.T.U. Dec. 2013]*

Ans. Transfer consists of a sequence of cycle each cycle consists of :

(i) Data packet = Data packet transmission time (T_d) + propagation time (T_P)

(ii) ACK Packet = ACK packet transmission time (T_a) + propagation time (T_p)

$\therefore$ Total time for each cycle (C)

$$C = T_d + T_a + 2T_P$$

$$T_d = \frac{\text{Number of bits data frame}}{\text{Data rate}}$$

$$= \frac{256}{10\times10^6} bps$$

$$T_a = \frac{\text{Number of bits per acknowledgement frame}}{\text{Data rate}}$$

$$= \frac{32}{10\times10^6}$$

$$T_P = \frac{\text{Distance between two terminals}}{\text{Propagation speed}}$$

$$= \frac{1\times10^3}{200\times10^6} \text{sec}$$

$$C = \frac{256}{10\times10^6} + \frac{32}{10\times10^6} + \frac{1\times10^3}{200\times10^6}$$

$$= \frac{1}{10^6}[25.6 + 3.2 + 5] = \frac{33.8}{10^6} \text{sec}$$

$$= 33.8\ \mu \text{ sec}$$

Ans.

Q.11 ***What is vulnerable time in case of pure and slotted ALOHA? How we can determine the underload and overload situation for channel in pure and slotted ALOHA.*** *[R.T.U. 2015]*

Ans. Vulnerable time in case of pure and slotted ALOHA is equal to 1 T_{fr} (Frame transition time).

Explanation: Pure ALOHA has a vulnerable time of 2 x T_{fr} . This is so because there is no rule that defines when the station can send. A station may send soon after another station has started or soon before another station has finished. Slotted ALOHA was invented to improve the efficiency of pure ALOHA. In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot. Because a station is allowed to send only at the beginning of the synchronized time slot, if a station misses this moment, it must wait until the beginning of the next time slot. This means that the station which started at the beginning of this slot has already finished sending its frame. Of course, there is still the possibility of collision if two stations try to send at the beginning of the same time slot. However, the vulnerable time is now reduced to one-half, equal to T_{fr}.

Conditions for overloading and underloading:

To express the throughput of the ALOHA random access scheme, it is often assumed that message transmission attempts occur according to a Poisson process with rate G attempts per slot. When G > 1, the channel overloads. When G < 0.5, the channel underloads.

Q.12 ***A pure ALOHA network transmit 200 bit frames on a shared channel of 200 kbps. What is the throughput if the system (all station together) produces 1000 frame per second?*** *[R.T.U. 2015]*

OR

A pure ALOHA network transmits 200 bits frames on a shared channel of 200 kbps. What is throughput if the system (all stations together) produce

(i) 1000 frames/sec.

(ii) 500 frames/sec. *[R.T.U. 2011]*

Ans. Average frame transmission time T_{tr} is 200 bits/ 200 kbps or 1 ms. The vulnerable time is 2×1 ms = 2ms.

Throughput for pure ALOHA is

$$S = G \times e^{-2G}$$

maximum throughput $S_{max} = 0.184$ when $G = \frac{1}{2}$

(i) If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case

$$S = G \times e^{-2G} \text{ or } S = 0.135$$
$$= 13.5\%$$

This means that the throughput is 1000 × 0.135 = 135 frames. Only 135 out of 1000 will probably survive.

(ii) If the system creates 500 frames per second *i.e.* (1/2) frame per millisecond. The load is (1/2). In this case

$$S = G \times e^{-2G}$$
$$S = 0.184 = 18.4\%$$

This means that the throughput is 500 × 0.184 = 92 and that only 92 frames out of 500 will probably survive. Note that this is the maximum throughput case, percentage wise.

Q.13 ***Give the applications of CSMA/CD.*** *[R.T.U. 2014]*

Ans. Applications of CSMA/CD Protocol : CSMA/CD was used in now obsolete shared media Ethernet variants (10BASE5, 10BASE2) and in the early versions of twisted-pair Ethernet which used repeater hubs. Modern Ethernet networks, built with switches and full-duplex connections, no longer need to utilize CSMA/CD because each Ethernet segment, or collision domain, is now isolated. CSMA/CD is still supported for backwards compatibility and for half-duplex connections. IEEE Std 802.3, which defines all Ethernet variants, for historical reasons still bears the title "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications".

The foremost application of the CSMA/CD protocol is in the technology that connects computer terminals located within a company, an institution, university campus etc., using wires. Such a technology is known as Local Area Network (LAN) technology. Over the past years there appeared several LAN technologies, but the first and by far the most prevalent one is the Ethernet technology also referred to as the IEEE 802.3 LAN technology. The Ethernet technology was developed in the mid seventies by Bob Metcalfe and David Boggs. Since then, although it faced challenges by several alternative LAN technologies (token ring, FDDI, ATM), it still dominates the marketplace. One of the reasons for this success is that the hardware required for its deployment became very cheap, which in turn is due to the large production volume and to the simplicity of the multiple access protocol used for communication, which is the CSMA/CD protocol with exponential backoff. Moreover, the Ethernet technology proved capable of adapting itself to user demands for increased transmission rates. Currently, Ethernet LANs run at speeds of 10Mbps, 100Mbps, and even 1Gbps.

Q.14 ***Consider two stations on a baseband bus at a distance of 1 km from each other. Let the data rate be 1 Mbps, the frame length be 100 bits and the propagation velocity be 2×10^8 m/sec. Assume that each station generates frames at and average rate of 1000 frames/sec.***

Find the probability of collision for a station which beings to transmit at time t for

(i) Pure ALOHA and

(ii) Slotted ALOHA *[R.T.U. Dec. 2013]*

Ans. For pure ALOHA, a packet transmitted by one station at time t will suffer a collision if the other station transmits in the interval.

$$(t - T_t - T_p, t + T_t + T_p)$$

where T_t = Packet transmission time

T_p = Propagation time

The probability that a station will transmit in a time interval T is :

$$P_r(Collision) = 1 - e^{-\lambda T}$$

where λ is the transmission rate.

We have

$$T_t = 100 \text{ bits} / 10^6 \text{ bps} = 10^{-4} \text{ sec}$$

$$T_p = \frac{10^3 m}{2\times10^8 m/\text{sec}} = 5\times10^{-6} \text{ sec}$$

$$T = 2(T_t + T_p) = 2.1\times10^{-4} \text{ sec}$$

$\therefore \lambda = 10^4$ packets/sec

$$\therefore \quad P_r(collision) = 1 - e^{-2.1} = 0.88$$ **Ans.**

For slotted ALOHA, the period of vulnerability is only $(t, t + T_t + T_p)$.

Thus,

$$T = T_t + T_p = 1.05\times10^{-4}$$

$$P_r(collision) = 1 - e^{-1.05}$$

$$= 0.65$$ **Ans.**

Q.15 ***What is sliding window protocol? What should be the size of window? Explain.*** *[R.T.U. 2013]*

Ans. Sliding window protocol : *Refer to Q.9.*

The size of window: Unlike the stop-and-wait mechanism, in the sliding window flow control, each data frame is not individually acknowledge, and therefore the sending end can send a number of frames one after the other without waiting for acknowledgement which results in better throughput or link utilization.

To calculate link utilization, let us consider the following two possible situation :

1. Station A receives an acknowledgement before it exhausts the window.
2. Station A exhausts the window before it receives an acknowledgement.

In the first situation, station A can keep on sending data frames without interruption and the link is never idle. The throughput or link utilization is unity. If the window size is W, this situation will occur when the time required to transmit W frames is more than the earliest possible arrival of an acknowledgement, *i.e.* $t_f + 2t_p$. It is assumed that the size of acknowledgement is very small.

$$Wt_f \geq t_f + 2t_p$$

$$W \geq 1 + 2A, \qquad \text{where } A = t_p/t_f$$

In the second situation, A sends W frames in time $W t_f$ and then suspends further transmission of the frames until an acknowledgement is received. If we assume that B receipt at the first frame, A shall receive the acknowledgement after time $t_f + 2t_p$. Therefore this situation will occur when,

$$Wt_f < 1 + 2t_p$$

$$W \leq 1 + 2A$$

the channel or link is engaged for time $t_f + 2t_p$. While station A has utilized it for time Wt_f. Therefore η is given by:

$$\eta = \frac{W.t_f}{t_f + 2t_p} = \frac{W}{1+2A}$$

Q.16 ***Explain the concept of checksum with suitable example. Also state that what kind of arithmetic is used to add data items in checksum calculation.***

OR

Define Checksum. *[R.T.U. 2013]*

Ans. Checksum : Checksum is the error detection method. The checksum is used in the internet by several protocols although not at the data link layer.

Like linear and cyclic codes, the checksum is based on the concept of redundancy. Several protocols still use the checksum for error detection, although the tendency is to replace it with a CRC.

Concept : The concept of the checksum is illustrated using example given below :

Example : Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers. For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12,0,6,36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum. If both are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.

One's Complement Arithmetic

We can make the job of the receiver easier if we send the negative (complement) of the sum, called the *checksum.* In this example, we send (7, 11, 12,0,6, -36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.

The previous example has one major drawback. All of our data can be written as a 4-bit word (they are less than 15) except for the checksum. One solution is to use **one's complement** arithmetic. In this arithmetic, we can represent unsigned numbers between 0 and $2n - 1$ using only n bits. If the number has more than n bits, the extra leftmost bits need to be added to the n rightmost bits (wrapping). In one's complement arithmetic, a negative

number can be represented by inverting all bits (changing 0 to 1 and 1 to 0). This is the same as subtracting the number from $2n - 1$.

Internet Checksum : Traditionally, the Internet has been using a 16-bit checksum.

Sender site: The sender calculates the checksum by following these steps :

1. The message is divided into 16-bit words.
2. The value of the checksum word is set to 0.
3. All words including the checksum are added using one's complement addition.
4. The sum is complemented and becomes the checksum.
5. The checksum is sent with the data.

Receiver site: The receiver uses the following steps for error detection :

1. The message (including checksum) is divided into 16-bit words.
2. All words are added using one's complement addition.
3. The sum is complemented and becomes the new checksum.
4. If the value of checksum is 0, the message is accepted; otherwise, it is rejected.

The nature of the checksum (treating words as numbers and adding and complementing them) is well-suited for software implementation. Short programs can be written to calculate the checksum at the receiver site or to check the validity of the message at the receiver site.

Q.17 ***Generate the CRC code for message 1101010101. Given generator polynomial.***

$g(x) = x^4 + x^2 + 1$ *[R.T.U. Dec. 2013]*

Ans. For polynomial division T(x) / G(x)

T(x) = 1101010101

$(x^9 + x^8 + x^6 + x^4 + x^2 + 1)$

$G(x) = x^4 + x^2 + 1$

= 10101

(1) The degree of polynomial $G(x) = x^4 + x^2 + 1 = 4$

So we append & zero to string $T(x)$.

Now the string becomes

11010101010000

(2) Divide B(x) by G(x) {After appending O, to $T(x)$ it becomes $B(x)$}

```
          1110001110
       ┌──────────────
10101 | 11010101010000
        11010
        10101
        -----
        011111
         10101
         -----
         010100
          10101
          -----
          000011010
              10101
              -----
              011110
               10101
               -----
               010110
                10101
                -----
                000110  ← Remainder
```

11010101010000

0110

11010101010110 ⇐ Code Word

PART-C

Q.18 ***Explain pure ALOHA protocol with suitable diagrams.*** *[R.T.U. 2019]*

Ans. ALOHA : It is the simplest possible broadcast protocol and it sets a basis with which other broadcast protocols can be compared. It is also known as **Pure ALOHA**. The basic idea of ALOHA is simple since users transmit immediately whenever they have data to send. To determine whether a transmission was successful, a sender waits for an acknowledgement from the receiver for a time period equal to one propagation time (the time it takes to travel a packet from the sender to the receiver and back again). If no acknowledgement is received, the packet will be sent again.

There will obviously be collisions between packets sent within a packet transmission time t_p from different users as indicated in fig. 1. We first assume that all packets have the same length and each requires one time unit t_p (called a slot) for transmission. Consider an attempt by a user to send packet A starting at time t_o. If another user generates packet B between t_o and $t_o + t_p$ the end of packet B will collide with the beginning of packet A. This can

occur because, owing to long propagation delays, the sender of packet A did not know that packet B was already under way when the transmission of packet A was started. Similarly, if another user attempts to transmit packet C between $t_0 + t_p$ and $t_0 + 2t_p$ the beginning of packet C will collide with the end of packet A. Thus if two packets overlap by even the slightest amount in the vulnerable period, collision will occur and both packets will be corrupted.

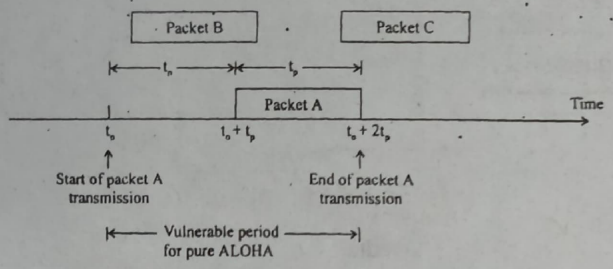


Fig. 1 : Packet transmission in ALOHA

Now let S be the channel throughput (the average number of successful transmission per time period t_p) and let G be the total traffic entering the channel from an infinite population of users (that is, G denotes the number of packet transmission that are attempted in time period t_p). To find the throughput, we first assume that the probability P_k of K transmission attempts per packet time follows a Poisson distribution with a mean G per packet time. This is given by

$$P_K = \frac{G^K e^{-G}}{\lfloor K} \quad \ldots(1)$$

The throughput S is then just the offered load G times the probability of a transmission being successful. Thus

$$S = GP_0 \quad \ldots(2)$$

Where P_0 is the probability that a packet does not suffer a collision. (P_0 is the probability of no other traffic being generated 'during a vulnerable period, which is, two packet, time long). From equation (1) the probability of zero packets being generated is equal to

$$P_0 = e^{-G} \quad \ldots(3)$$

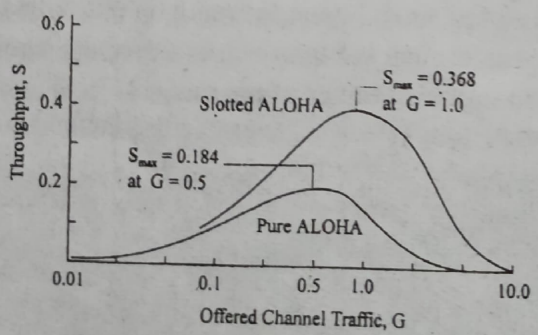


Fig. 2 : Relation between throughput and offered channel traffic

In an interval two frame times long, the mean number of frames generated is 2G. The probability of no other traffic being initiated during the entire vulnerable period is given by $P_0 = e^{-2G}$ by equation (2) we have

$$S = Ge^{-2G} \quad \ldots(4)$$

The throughput given by equation (4) is plotted in fig. (2). The maximum value of S occurs at G = 0.5 where S = 1/2e which is about 0.184. This means that the best channel utilization that can be achieved is around 18 percent for the pure ALOHA method.

Q.19 ***Compare and discuss the throughput of pure and slotted ALOHA.*** *[R.T.U. 2017]*

OR

Explain Pure ALOHA and Slotted ALOHA. Give relationship in terms of their throughput. *[R.T.U. 2014]*

OR

Show that slotted ALOHA has a maximum throughput of twice the pure ALOHA maximum throughput. *[R.T.U. 2016]*

OR

Show that the slotted ALOHA has a maximum throughput of twice the maximum throughput. *[R.T.U. Dec. 2013, 2013, 2012, 2007]*

Ans. Pure ALOHA : *Refer to Q.18.*

Slotted ALOHA

To increase the efficiency of the ALOHA method, the slotted ALOHA scheme was introduced. In this type of ALOHA the channel is divided into time slots which are exactly equal to a packet transmission time.

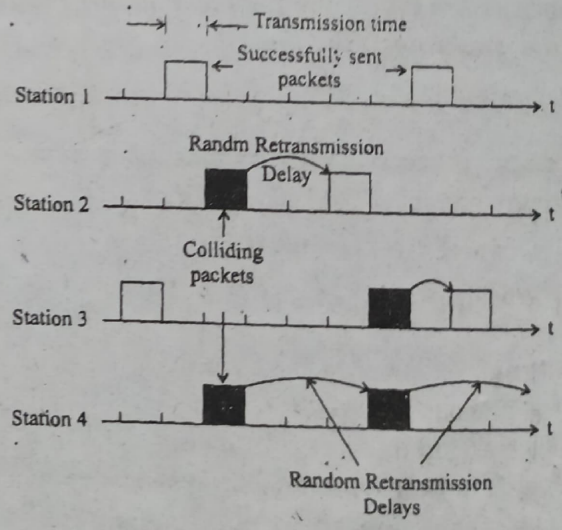


Fig. : Packet transmission in slotted ALOHA

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 vulnerable period in which this packet can collide with other data is reduced to one packet time period (PTP) versus two PTP for pure ALOHA. Examples of transmission attempts and random retransmission delays for colliding packets are shown in fig. for four network users.

Since the vulnerable period is now reduced by half, the probability of no other traffic occurring during the same time period as the packet, we wish to send is $P_0 = e^{-G}$. This in turn leads to a throughput

$S = G\, e^{-G}$

As is shown in fig. (see fig. 2 in Q.18) the maximum efficiency of the slotted ALOHA system occurs at G = 1 where S = 1/e or about 0.368, which is twice that of pure ALOHA.

Vulnerability period *i.e.* the period in which a transmitted frame will suffer collision $\propto \frac{1}{\text{efficiency}}$

If pure ALOHA is suffering 2T vulnerability period and slotted ALOHA is suffering T vulnerability period, it means efficiency of slotted ALOHA is twice that of pure ALOHA.

So, we proved efficiency of slotted ALOHA = 2 (Efficiency of pure ALOHA).

Q.20 ***What is two dimensional parity check?***

[R.T.U. 2017]

Ans. Parity Check: This error detecting technique is least expensive. Parity checking can be simple or two dimensional.

Simple Parity Check: In this technique a redudent bit called a parity bit is added to every data unit so that the total number of 1's in the unit (including parity bit) becomes even (or odd).

But in many cases even parity checking are applicable where the number of 1's should be even.

From figure suppose we want to transmit the binary data 1100001; gives us 3, an odd number. Before transmitting, we pass the data unit through a parity generator. The parity generator counts the 1's and append the parity bit (1 in this case) to the end. The total no. of 1's is now 4, an even number. The system now transmits entire expanded unit reaches its destination, the receiver parts all 8 bits through an even parity checking function. If the receiver sees 11000011 it counts four 1's, an even number and the data unit passes.

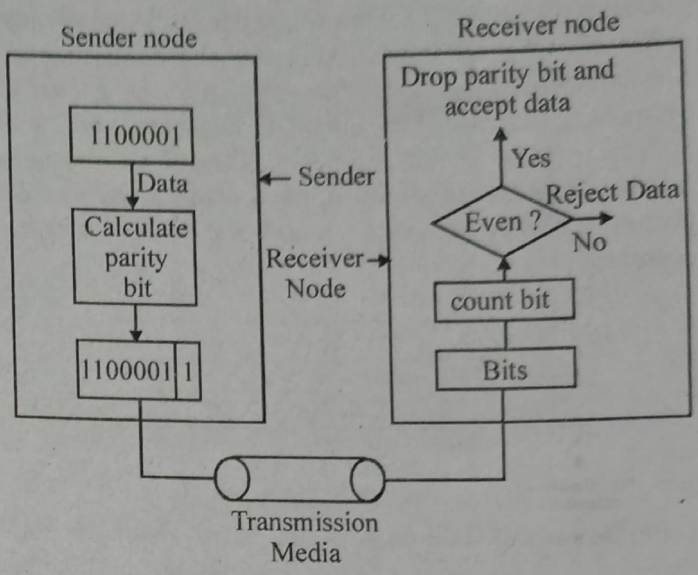


Fig. : Even parity concept

But what if the data unit has been damaged in transmit? What if, instead of 11000011, the receiver sees 11001011? Then when the parity checker counts the 1's, it gets 5, an odd number. The receiver knows that an error has been introduced into the data somewhere and therefore rejects the whole unit.

Performance: Simple parity check can detect all single bit errors. Let's say we have an even parity data unit where the total number of 1's, including parity bit is 4: 100010011. If any 3 bits change value, the resulting parity will be odd and the error will be detected. The checker would return a result of 1, and the data unit would be rejected. The same holds true for any odd number of errors. Suppose however that 2 bits of data unit are changed: 100010011: 4.

In this case the number of 1's in the data unit is still even. The parity checker will and return an even number although the data unit contains two errors. This method cannot detect errors where the total number of bits changed are even. If any two bits change in transmission, the changes cancel each other and the data unit will pass a parity check though the data unit is damaged. The same holds true for any number of errors.

Two Dimensional Parity Check : In this method, a block of bits is organized in a table (rows and columns). First we calculate the parity bit for each data unit, then we organize them into a table.

From figure, we have four data units shown in four rows and eight columns. We then calculate the parity bit for each column and create a new row of 8 bit; they are the parity bits for the whole block. Note that the first parity bit in the fifth row is calculated based on all the first bits; the second parity bit is calculated based on all second bits

and so on, we then attach the 8 bits to the original data and send them to the receiver.

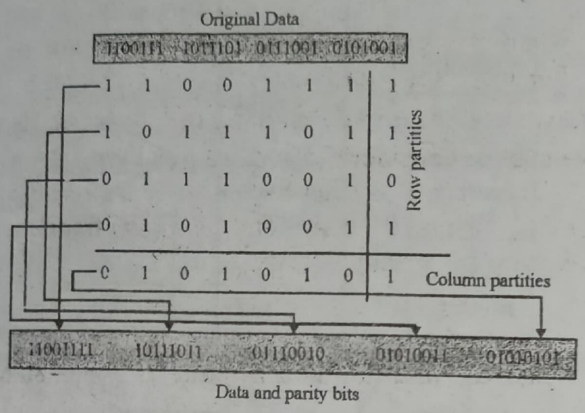


Fig. : Two dimensional parity

Performance: Two dimensional parity check method is very popular in today's life. In this method a redundancy of n bits can easily detect a burst error of n bits.

A burst error of more than n bits is also detected by this method with a very high probability. There is, however, one pattern of errors that remains elusive. If 2 bits in one data unit are damaged and 2 bits in exactly the same positions in another data unit are also damaged, the checker will not detect an error.

Consider, for example, two data units; 11110000 and 11000011. If the first and last bits in each of them are changed, making the data units read 0111001 and 01000010, the error be detected by this method.

Q.21 ***Explain the types of errors and classification of codes.*** *[R.T.U. 2017, 2012, 2010]*

Ans. Types of errors

The errors intorduced in the transmitted data during their transmission may be categorized as under

(i) Content errors

(ii) Flow integrity errors

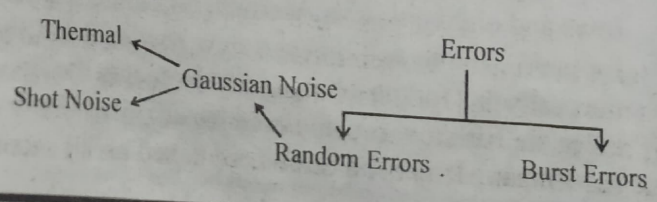
Content errors:

The content errors are nothing but errors in the contents of a message i.e., a 0 may be received as 1 or vice-versa.

Flow integrity errors:

Flow integrity errors meaning missing blocks of data. It is possible that a data block may be lost in the between as it has been delivered to a wrong destination.

Types of Errors



The errors in a digital communication system are caused by noise in the communication channel (Gaussian noise introduce in analog part of common channel).

Random errors due to white Gaussian noise are introduced. Gaussian noise had been our chief concern in designing and evaluating modulators and demodulators. Sources of Gaussian are :

(a) Thermal Noise : Due to vibration of individual molecules about their position of equilibrium in a crystal lattice, the conduction electron of metals wander randomly throughout the volume of metal, similarly molecule of an enclosed gas are in constant motion colliding with one another and colliding also with the walls of container. These agitations of molecules are called thermal agitations because they increase with temperature.

(b) Shot Noise : Result from a phenomenon associated with flow of current across semiconductor junctions. The charge carriers, electrons or holes enter the junction region from one side, drift or are accumulated at the junction and are collected on other side. The average junction current determines the average interval that elapses between time when two successive carriers enter the junction. The exact interval that elapses is subject to random fluctuations. This randomness give rise to shot noise. As we know that power spectral density of Gaussian noise at receiver input is white Gaussian noise. The transmission errors introduced during a particular interval by white Gaussian noise does not affect the performance of system during subsequent signalling interval.

(c) Burst Errors : Which is due to impulse noise by long quite intervals followed by high amplitude noise burst. This type of noise occurs from many natural and man-made causes such as lightning and switching transients. When such noise occurs, it affects more than one symbol or bit and there is usually a dependence of errors in successive transmitted symbols.

Error control schemes for dealing with random errors are random error correcting codes and coding scheme designed to correct burst errors are burst over correcting codes.

Shot Noise : Shot noise appears in active devices due to the random behaviour of charge carriers (electrons and holes). In electron tubes, shot noise is generated due to the random emission of electrons from cathodes; in semiconductor devices, it is caused due to the random diffusion of minority carriers or random generation and recombination of electron-hole pairs.

Current in electron devices (tubes or solid state) flows in the form of discrete pulses, every time a charge carrier moves from one point to the other (e.g., cathode to plate). Therefore, although current appears to be continuous, it is

still a discrete phenomena. The nature of current variation with time is shown in fig.

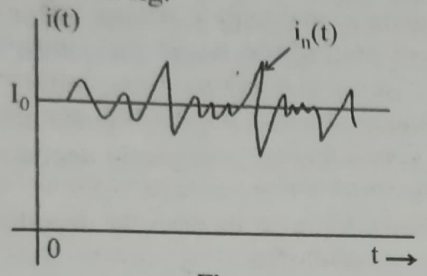


Fig.

The current fluctuates about a mean value I_0. This current $i_n(t)$ which wiggles around the mean value is known as shot noise. The wiggling nature of the current is not visualized by normal instruments and normally we think that the current is a constant equal to I_0. The wiggling nature of the current can be observed in a fast sweep oscilloscope.

The total current i(t) may be expressed as

$$i(t) = I_0 + i_n(t) \quad \ldots(1)$$

where I_0 is the constant (mean) and $i_n(t)$ is the fluctuating (noise) current.

Power Density Spectrum of Shot Noise in Diodes : The time varying component $i_n(t)$ of the current i(t) specified by eq.(1) is random in nature and it cannot be expressed as a function of time, i.e. it is an indeterministic function. However, this indeterministic stationary random $i_n(t)$ can be specified by its power density spectrum.

The number of electrons contributing to the random stationary current $i_n(t)$ are large. Assuming that the electrons do not interact with each other during their movement or emission (e.g., temperature limited diode current), the process may be considered statistically independent. According to central limit theorem, such a process has a Gaussian distribution. Hence, shot-noise is Gaussian-distribution with a zero mean.

The total diode current may be taken as the sum of the current pulses, each pulse being formed by the transit of an electron from cathode to anode. It can be seen that for all practical purposes the power density spectrum of the statistically independent non-interacting random noise current $i_n(t)$ is given by:

$$S_i(\omega) = qI_0 \quad \ldots(2)$$

where q is the electronic charge ($q = 1.59 \times 10^{-10}$) coulombs and I_0 is the mean value of the current in amperes. The power density spectrum in eq.(2) is frequency independent. This type of frequency independence is only up to a frequency range decided by the transit time of an electron to reach from anode to cathode. Beyond this frequency range, the power density varies with frequency as shown in fig.(a). The transit time of an electron, in a diode depends on anode voltage V and is given as

$$\tau = 3.36 \times \frac{d}{\sqrt{V}} \mu sec$$

where d is spacing between anode and cathode.

For instance, in a diode with d = 2mm and V = 40 volts, we have $\tau \cong 10^{-3}$ μ sec. In fig.(a), the power density curve may be considered flat close to the origin, i.e. $|\omega\tau| \leq 0.5$. Therefore $S_i(\omega)$ can be considered constant up to $|\omega\tau| = 0.5$. For $\tau = 10^{-3}$ μ sec. The maximum frequency up to which power density $S_i(\omega)$ remains constant is given by

$$\omega = 0.5 \times 10^9 = 5 \times 10^8 \text{ rad/s}$$

This is equivalent to a linear frequency $(f) = \frac{\omega}{2\pi} = 80$ MHz.

Therefore for all practical purposes, the $S_i(\omega)$ may be considered to be frequency-independent below 100 MHz.

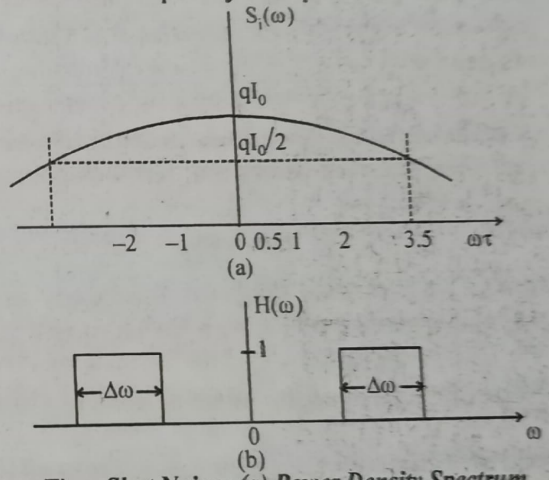


Fig. : Shot Noise : (a) Power Density Spectrum (b) Bandwidth of Measuring System

Resistor Noise : The noise arising due to random motion of free charged particles (usually electrons) in a conducting media, such as a resistor, is called resistor noise. This noise is also known as Jhonson noise, after J.B. Johnson who, investigated this type of noise in conductors. The random agitation is a universal phenomenon at atomic levels and is caused by the energy supplied through flow of heat. The intensity of random motion is proportional to thermal (heat) energy supplied (*i.e.* temperature) and is zero at a temperature of absolute zero. This noise is also known as thermal noise. The path of the electron motion is random because of their collisions with lattice structure. The net motion of all the electrons gives rise to an electric current to flow through the resistor, causing the noise.

Power Density Spectrum of Resistor Noise :

The free electrons contributing to resistor noise are large in number. If their random motion is assumed to be statistically independent, then the central limit theorem predicts the resistor noise to be, Gaussian, distributed with a zero mean. It can be shown that the power density

spectrum of the current contributing the thermal noise is given by :

$$S_i(\omega) = \frac{2kTG}{1+\left(\frac{\omega}{\alpha}\right)^2} \quad \ldots(1)$$

where T is ambient temperature in degree Kelvin, G is the conductance of the resistor in mhos, k is the Boltzmann constant and α is the average number of collisions per second per electron.

The variation of power density spectrum with frequency is shown in Fig.

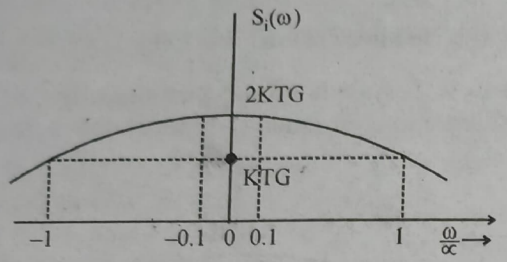


Fig. : Power Density Spectrum of the Resistor Noise Current

It is obvious from the figure that the spectrum may be considered to be flat for $\frac{\omega}{\alpha} \le 0.1$. The power density spectrum $S_i(\omega)$ for this range of frequency is nearly constant and is given by

$$S_i(\omega) = 2kTG \quad \ldots(2)$$

The value of α is of the order of 10^{14} and hence the frequency corresponding to $\frac{\omega}{\alpha} < 0.1$ is of order of 10^{13} Hz. Therefore, the frequency independent expression of $S_i(\omega)$ given by eq. (2) holds up to a frequency range of 10^{13}Hz. This range covers almost all the practical applications in communication systems. Hence, for all practical purposes, the power density spectrum $S_i(\omega)$ is considered to be independent of frequency.

Classification of codes :

The codes are basically classified as under:

(i) Errors detecting codes: The error detecting codes are capable of may detecting the errors. They cannot connect errors.

(ii) Error correcting codes

(1) Block codes (2) Convolution codes

The error correcting codes are capable of detecting as well as correcting the errors. These codes can be classified into block codes and convolution codes or linear and non-linear codes.

For error- free transmission, following codes are used :

(A) Block Codes

(B) Burst and Random Error Correcting Codes

(C) Interleaving

(A) Block Codes

(i) For Error Correction

1. We compare the performance of system using block codes for error correction with systems (n,k) using no error control coding.
2. Two measures of performance are :
   (a) Problem of incorrectly decoding a message bit.
   (b) Problem of incorrectly decoding a block of message digits.
3. We will do the comparison on the condition that rate of information transmission is same for coded and uncoded systems and both systems are operating with average signal power and noise power spectral density.
4. Coded or uncoded a block of say k message bits must be transmitted in duration of time.

   $$T_W = \frac{k}{r_b}$$

   where r_b = message bit rate
5. $\therefore \quad r_w = \frac{1}{T_W} = \frac{r_b}{k}$

   if system uses an (n, k) block code, then bit rate going into channel

   $$r_c = r_b\left(\frac{n}{k}\right) \text{ or } r_c > r_b$$
6. Now

   r_b = Message bit rate.

   r_c = Channel bit rate.

   q_c = Channel bit error probability for coded system.

   q_u = Channel bit error probability for uncoded system.

   P_{be}^u = Probability of incorrectly decoding a message bit in uncoded system.

   P_{be}^c = Probability of incorrectly decoding a message bit in coded system.

   P_{we}^u = Probability of incorrectly decoding a word of message bits in uncoded system.

   P_{we}^c = Probability of incorrrectly decoding a word if message bits in coded system.
7. Now in uncoded case

   $P_{be}^u = q_u$ and probability that word of k message bit incorrectly received.

   P_{we}^u = 1 – P(all k message bits are correctly received)

   $= 1 - (1 - q_u)\,k$ when $kq_u \le 1$

   $= p^4{}_{we} = kq_u$

since transmission errors are assumed to be independent.

8. In coded system, a word of k message digits will be incorrectly decoded when more than t errors occur in a n-bit codeword since block code is assumed to be able to correct upto t errors.

Thus

$$P_{we}^{c} = P(t+1 \text{ or more errors in a codeword})$$

$$P_{we}^{c} = \sum_{i=t+1}^{n} \binom{n}{i} q_c^i (1-q_c)^{n-i} = \sum_{i=t+1}^{n} P(n,i)$$

where $P(n,i) = \binom{n}{i} q_c (1-q_c)^{n-i}$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}; \quad r_{bw} = \frac{r_b}{k}$$

$$r_b P_{be}^{c} = r_w (t+1) \frac{k}{n} P_{we}^{c}$$

If $nq_c \le 1, P(n,i+1) \le p(n,i)$

$$P_{we}^{c} = \binom{n}{t+1} (q_c)^{t+1} (1-q_c)^{n-t-1}$$

P_{be}^{c} = message bit error probability.

P_{be}^{c} implies that majority of decoding errors are due to (t + 1) bit errors in an n-bit codeword.

Out of (t + 1) error, the fraction k/n represent the errorneous message bits. Hence average message bit error rate.

$$r_b P_{be}^{c} = r_w (t+1) \frac{k}{n} \cdot P_{we}^{c}$$

$$\therefore \quad P_{be}^{c} = \frac{(t+1)}{n} P_{we}^{c}$$

(ii) For Error Detection

1. We compare the performance of data transmission system using block codes for error detection with systems using direct transmission without error control coding.
2. We do the comparison under the assumed same assumption that S_{au}, η, r_b remain same for both coded and uncoded systems.
3. We use probability of incorrectly decoding a block of k message bits as our measure of comparative performance.
4. Here we assume that (n, k) block code is capable of detecting upto 2t errors per block.
5. Decoder checks the received codewords for errors and when error is deduced, the decoder may either discard or retransmit the message.
6. We know that data rate r_e over channel is $r_b\left(\frac{n}{k}\right)$ when an (n , k) error correcting block code is used. The data rate r_e will have to be higher than $r_b\left(\frac{n}{k}\right)$ for (n, k) error detecting block codes because of retransmission.

Stop and Wait Transmission Method :

1. The transmitter begins transmission at time say t_0 and completes the transmission of block of n bits at time $t_0 + t_n$.
2. Decoder starts receiving message block at time $+\Delta$ where Δ is propagation delay.
3. At time $t_0 + \Delta + t_n$ the decoder checks the n–bit block that was received and sends a positive acknowledgement (ACK) or a negative acknowledgement (NACK) to transmitter depending upon whether or not it detects an error in received block.
4. If ACK is positive, then next message block is transmitted if not previous is retransmitted. In either case transmitter is waiting from $t_0 + t_n$ to $t_0 + t_n + 2\Delta$ for acknowledgement from receiver.
5. To average channel all error rate blocks. Let us consider the transmission of N blocks of data over channel at rate r_c bits per second.
6. Total time needed to complete transmission we know r_c - bits per second rate.

$$T = \frac{1}{r_c}$$

Total n bits

$$\therefore \quad T = \frac{n}{r_c}$$

Delay = 2Δ

$$T = \left(\frac{n}{r_c} + 2\Delta\right)$$

$$\therefore \quad \text{for N block time} = N\left(\frac{n}{r_c} + 2\Delta\right)$$

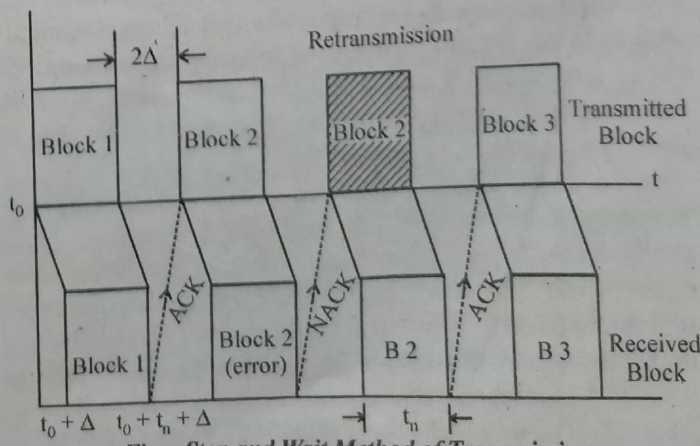


Fig. : Stop and Wait Method of Transmission

7. P_{we}^{c} = Probability of incorrectly decoding a message block in coded system for N blocks.

NP_{we}^{c} – Block in errors

$\therefore$ $N(1 - P_{we}^{c})$ – Blocks are correctly accepted by receiver block having k message bits

$\therefore$ Time taken

$$\frac{Nk\left(1-P_{we}^{c}\right)}{r_b} = \frac{N\left(n+2\Delta r_c\right)}{t_c}$$

(B) Burst and Random Error Correcting Codes

As we know that in some coding techniques error occurs either at random or in bursts for channels in which both random or burst errors occur. It is better to design codes capable of correcting random errors and/ or single or multiple errors.

Burst of length q is defined as a vector whose non-zero components are confined to q consecutive digits position, the first and last of which is non-zero.

for example

$v = (001\underline{010001}000)$

Burst of length = 7

A code that is capable of correcting all burst errors of length q or less is called as q–burst error correcting code or code is said to have burst correcting ability.

Theorem : The number of parity check bits of a q burst error correcting code must have at least 2q is

$$n - k \geq 2q$$

Proof : We can proof this theorem by showing that following two statements are true.

(i) A necessary condition for a (n, k) linear code to be able to correct all burst of length q or less can be a code vector.

(ii) The number of parity check digits of a (n, k) linear code has no burst of length b or less as a code vector is at least b is $n - k \geq b$.

To prove part (i) consider a code vector v with burst of length 2q or less this code vector can be expressed as a vector sum of v_1 and v_2 of lengths q or less. Thus in standard array of code v_1 and v_2 must be in same coset. If one of these vectors is a coset leader, then other will be uncorrectable error pattern. This code will not be able to correct all burst of length q or less.

To prove part (ii) consider vector whose non-zero components are confined to first b bits, therefore, there are 2b such vectors no two such vectors can be in same coset of standard array for this code; otherwise their sum which is a burst of b or less would be a code vector. Hence these 2b vectors must be in 2b distinct cosets. There are total 2^{n-k} cosets for an (n, k) code. Thus (n – k) must be at least equal to by combining (i) and (ii). Hence proved.

Burst error correcting capacity of (n, k) code is at most $\left(\frac{n-k}{2}\right)$

$\therefore$ upper bound of burst error correcting capability of (n, k) code

$$q \leq \frac{n-k}{2}$$

Burst correcting efficiency $y = \frac{2q}{n-k}$

If a code is needed for detecting a burst of length $\leq d$ then, number of check bits needed must satisfy.

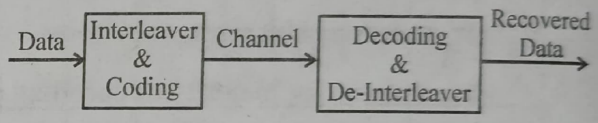
$$n - k \geq d$$

Decoding algorithm for correcting burst errors are similar to algorithm for cyclic codes designed to correct random errors.

(C) Interleaving

(i) Block Interleaving

1. A primary technique which is effective in overcoming burst errors is interleaving.

2. 

3. Before the data stream is applied to channel the data goes through a process of interleaving and error correction coding.
4. At the receiving side, data is decoded as data bits are evaluated in a manner to take advantage of error correcting and detecting features which result from coding and process of interleaving is undone.
5. Group of k*l* bits is loaded into shift registers which is organized into k rows with *l* bits/ row.
6. The data stream is entered into storage element at a_{11}. At each shift each bit moves one position to right while the bits in rightmost storage element moves to leftmost stage of next row.
7. When k*l* data bits have been entered, register is full, the first bit being in a_{kl} and last bit = a_{11}.
8. Now this data stream is diverted to the second shift register and process of coding is applied to data held stored in first register.
9. In this coding process, the information bits in column (eg., $a_{11}, a_{21} \ldots\ldots a_{k1}$) are viewed as the bits of uncoded word to which parity bits are added : code word.

$a_{11}, a_{22} \ldots\ldots\ldots a_{k1}, C_{11}, C_{21} \ldots\ldots C_{r1}$ is formed k

information bits and r check bits.
b bits/row

$$\left.\begin{matrix} a_{11} & a_{12} & \cdots & a_{1\ell} \\ a_{21} & a_{22} & \cdots & a_{2\ell} \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{k\ell} \end{matrix}\right\} \text{k bits/ column}$$

$$\left.\begin{matrix} c_{11} & c_{12} & \cdots & c_{1\ell} \\ c_{21} & c_{22} & \cdots & c_{2\ell} \\ \vdots & \vdots & \vdots & \vdots \\ c_{r1} & c_{r2} & \cdots & c_{r\ell} \end{matrix}\right\} \text{r parity bits/ column}$$

Interleaved code = $(\lambda n, \lambda k)$
$= (5 \times 15, 5 \times 7) = (75, 35)$

Each row is a 15 bit code word →

Five code word ↓

1	6		36	41			66	71
2	7		37	42			67	72
3	8	...	38	43	...	...	68	73
4	9		39	44			69	74
5	10		40	45			70	75

with $\lambda = 5$ with a burst error correcting ability = 10. The arrangement of codewords by interleaving as shown in 4 rows and 5 columns.

Let us assume that errors occur at position 5, 37, 43 and 69. The decoder operates on column and in each row max 2 errors are possible.

∴ By this interleaving, these errors are corrected. Here 5, 69 are random errors and 37 and 43 are burst errors.

10. As we see that information bits in of particular codeword is *l* bits apart that original bits stream or we can say two adjacent bits are *l*-bits distance apart like.

    $a_{11}, a_{21} \ldots\ldots a_{k1}$

11. When coding is complete, the entire content of $k \times l$ information bits as well as $r \times l$ parity bits are transmitted over channel.
    Generally transmission carried out row by row in order.

$C_{r\ell} - C_{r1} \ldots\ldots C_{1\ell} \ldots\ldots C_{11} a_{k\ell} \ldots\ldots a_{k\ell} \ldots\ldots$
$a_{2\ell} \ldots\ldots a_{21} a_{1\ell} \ldots\ldots a_{12} a_{11}$

12. Here we see that parity bits are transmitted. The received data is again received in same order as in transmission and error correction is performed. The parity bits are discarded.
    For example, we have (15, 7) BCH code generated by $g(x) = x^8 + x^4 + x^2 + x + 1$
    Given $d_{min} = 5$

∴ q = error correcting ability of BCH

$$q \le \frac{d_{min} - 1}{2}$$

$$\le \frac{5-1}{2} = \le 2$$

∴ double error correcting ability
Here we have $\lambda = 5$.

(ii) Convolutional Interleaving

1. This is an another interleaving scheme.
2. In fig. we have transmitter and the receiver and four switches. There are total *l* lines on the both transmitter and receiver side.
3. On the transmitter side, we have 0 storage elements on line 1 and storage element increases by 5.
   As we move on transmitter side line to line or receiver side, first line consists of $(l-1)$S storage element, second line $(l-2)$S and so on. The sum of storage elements on a specific line of transmitter and receiver is constant and equal to $(l-1)$ S.
4. Here in fig. we have four switches that operates in step and more from line to line at rate of d(k). Thus each switch move from line1, then to line 2 and so on.
5. Let us consider a single line l_i on transmitter side. Suppose that during a particular bit interval of d(k) there is a switch contact at both input and output sides of line l_i.

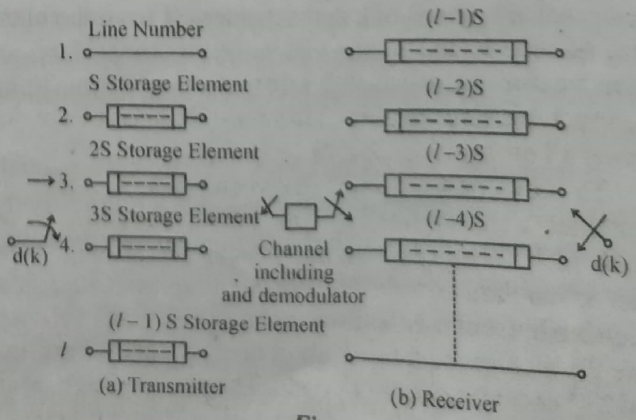


Fig.

6. At the end of bit interval, a clock signal causes the shift register of line l_i to enter the bit on input side at leftmost place and start moving content of each storage element one bit right.
7. This process has started and synchronous clock advances the switch to next line $l_i + 1$.
8. When shift register response is completed there will be a new bit at output end of line l_i. But because of propagation delay through storage elements, switch has already lost contact with line l_i before new bit appeared at output.
9. In summary during interval of input d(k) during which switches were connected to line l_i, there is one bit shift of shift register on line l_i which accepts d(k) into register.
10. Such a shift is not noticed until the next time switch make contact with line l_i.
11. Now, let us consider that initially all shift registers are short circuit. (at both transmitter and receiver side). In this case received sequence is same as transmitted.
12. With shift register in transmitter and receiver each line has $(l-1)$ delay and therefore output segment will be interleaved.

    Suppose that two successive input bit stream are transmitted d(k) and d (k + 1) then d(k + 1) will be $(d + k + l_s)$.

    $l = 5$, $S = 3$.

    $lS = 15$ meaning 15 bits are interleaved between two bits.

Source Code : There are two types of source code. Source codes are often divided into two broad categories:

1 Block codes

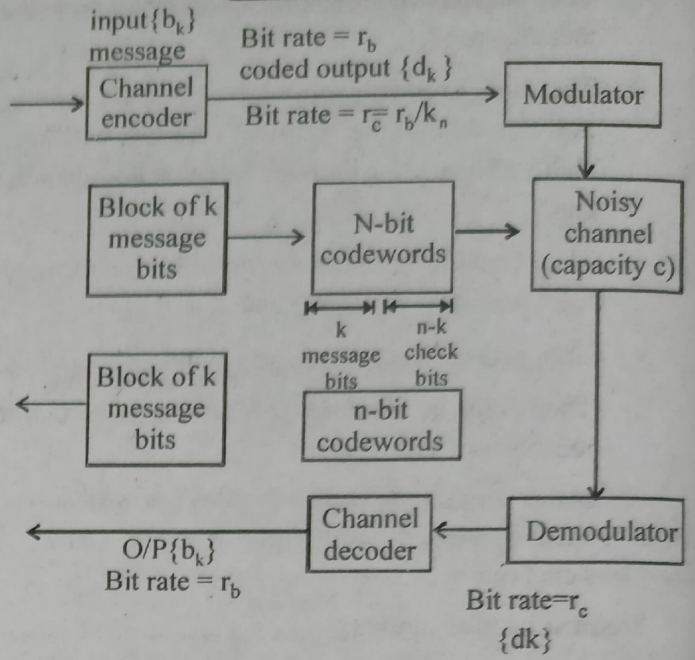
2 Convolutional codes

In block codes, a block of k information bits is followed by a group of r check bits that are derived from the block of information bits. At the receiver, the check bits, are used to verify the information bits in the information block preceding the bits.

In convolutional codes check bits are continuously interleaved with information bits; the check bits verify the information bits not only in the block immediately preceding them but in other blocks as well.

Error Control Coding

The practical alternative for reducing the probability of error is the use of error control coding, also known as channel coding.



Error control coding is the calculated use of redundancy.

The functional blocks that accomplish error control coding are the channel encoder and the channel decoder.

[Error control coding. Channel bit error probability is a/c = P {d ≠ k d_k} and message bit error probability is P_E = P {b ≠ k, b_k}

b_k - binary output r_b bits/sec.

The channel encoder and decoder are functional blocks in system that by acting together, reduce the overall probability of error. The encoder divides the input message bits into block of k message bits and replace each k bit message block D, with an n bit codeword Cw by acting n – k check bits to each message block. The mapping rules for coding and decoding are to be chosen such that error control coding lowers the overall probability of error. Important aspects of error controlling code:

1 It is possible to detect and correct errors by adding extra bits called check bits to the message stream.

2 It is not possible to detect and correct all errors.

Rate efficiency of a coding scheme is defined as r_b/r_c. There are various types of error correcting codes, some of them are given below-

1. Parity check bit coding for error detection
2. Block codes
3. BCH codes

Various applications of the error correcting codes-

1. They help in removing any type of errors from the message send.
2. They help in adding some extra bit to the sending message so that the received message will have ability of correcting the code comes at the receiving end.
3. They provide accurate transmission of message from one place to other place.
4. They provide good efficiency of message sending.
5. They help in sending correct message to the receiver.

An another classification of codes are as follows: Let us consider the following fuble where a source of size 4 has enceted in binary codes symbol 0 and 1.

Instantaneous codes:

A uniquelly decodable code is called an instantaneous code if the end of any codeward is recognizable without examining. Subsequent code symbols. Prefix free codes are sometimes known as instantaneous codes.

Optimal codes :

A code is said to be optinal if it is instantaneous and has minimum average L for a given source with a given probability assignment for the source symbol.

Q.22 *(a) Write short notes on Cyclic codes.[R.T.U. 2017]*

OR

What are the cyclic codes? Write the advantages and disadvantages of cyclic codes. *[R.T.U. 2016]*

(b) Write short notes on encoder and decoder for cyclic codes. *[R.T.U. 2017]*

Ans.(a) Cyclic Codes : Cyclic code has the property that a cyclic shift of one codeword of the code forms another codeword.

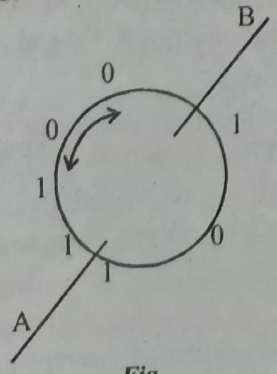


Fig.

Meaning of cyclic shift is explained from figure i.e. n bit word instead of being written out horizontally is written around a circle. Starting at any point A the 7-bit word encountered by a clockwise rotation is 1101001; starting at some other arbitrary point say B we would read 0111010. The two words are related such that one is derived from other by cyclic shift. There are seven possible starting plans as shown in fig. Order in which the words are generated depends on the direction, clockwise or counterclockwise of the shift, but the end result of the resultant collection of words is not affected by the shift direction.

A procedure for generating an (n, k) cyclic code is the following :

The bits of the uncoded word $\overline{A} = (A_0 A_1 .. A_{k-1})$ are written as the polynomial.

$$A(x) = A_0 \oplus A_1 x \oplus A_2 x^2 \oplus ... \oplus A_{k-1} X^{k-1} \quad ...(1)$$

The bits of the coded word $\overline{T} = [T_0 T_1 ... T_{n-1}]$ are written as the coefficients of the polynomial.

$$T(x) = T_0 \oplus T_1 x \oplus T_2 x^2 \oplus ... \oplus T_{n-1} X^{n-1} \quad ...(2)$$

We next form the "generating" polynomial g(x) of degree r = n – k.

$$g(x) = 1 \oplus g_1 x \oplus g_2 x^2 \oplus ... \oplus g_{r-1} x^{r-1} \oplus x^r \quad ...(3)$$

and we determine the values of the coefficient $g_1, g_2, ... g_{r-1}$ from the condition that g(x) be a factor of the polynomial

$$f(x) = x^n \oplus 1 \quad ...(4)$$

where n is the number of bits in the codeword. Finally, when g(x) is determined, T(x) is found from the equation-

$$T(x) = g(x) A(x) \quad ...(5)$$

As an example of the application of this procedure, let us generate a (7, 4) code since n = 7

$$f(x) = x^7 \oplus 1 \quad ... (6)$$

It can be verified that factors of f(x) are

$$f_1(x) = \lambda_1(x) . \lambda_2(x) . \lambda_3(x)$$

$$= (1 \oplus x)(1 \oplus x \oplus x^3)(1 \oplus x^2 \oplus x^3) \quad ...(7)$$

To generate a code with n = 7 bits, T(x) in equation (2) must be a polynomial of degree n – 1 = 6.

Advantages and Disadvantages of Cyclic Codes

As we have seen that cyclic codes are the subclass of linear block codes, they have some advantages over noncyclic block codes as given below-

Advantages:

(1) The error correcting and decoding methods of cyclic codes are simpler and easy to implement. These methods eliminate the storage needed for lookup table decoding. Therefore the codes becomes powerful and efficient.

(2) The encoders and decoders for cyclic codes are simpler compared to noncyclic codes.

(3) Cyclic codes also detect error burst that span many successive bits.

(4) Cyclic codes have well defined mathematical structure. Hence very efficient decoding schemes are possible.

Disadvantages:

(1) The error detection in cyclic codes is simpler but error correction is little complicated since the combinational logic circuits in error detector are complex.

Ans.(b) Cyclic codes are an important class of linear block codes in which the cyclic shifting of the message bits results in another code vector, hence the name cyclic code. In other words a cyclic shift in a code word in C results in another code word in C. For Example C = {(110), (101), (011)} is cyclic, while C = {(000), (100), (111)} is not cyclic. A cyclic code is defined in terms of a generator polynomial g(X) of degree less than n-k , where k is the length of input message and n is the length of the encoded message. Cyclic codes are studied usually in polynomial form since it is easy to represent the code vectors as the coefficients of the polynomial. For example the message 110001 is represented as 1+ X + X?. Cyclic Codes are used both for encoding and decoding of the message bits. In an encoding process the message signal is divided by a certain sequence called generator number and corresponding to that the remainder bits form the parity check sequence which are concatenated to either front or back of the message signal to encode it. The process of encoding can be implemented using shift registers as shown in the Fig. 1, where g0,g1,..gn-k-1 is for generator bits, b0,b1,..bn-k-1 is for remainder bits and u(X) is for message bits. The decoding process is a more complex process and is followed by error detection and correction. In error detection the received vector is divided by the generator sequence and the computed remainder forms the syndrome. If the computed syndrome is identical to zero, then the received vector is error free, else the process advances to error correction. In error correction the computed syndrome is compared with the error pattern and accordingly the erroneous bit is XOR-ed with the error bit and the corrected received vector is obtained.

Q.23 ***Explain the stop and wait ARQ Protocol and also discuss the Piggy backing method.*** *[R.T.U. 2016]*

Ans. Stop and Wait ARQ Protocol : SW ARQ protocol is a simple protocol for handling frame transmission errors when the round-trip time ($2\tau_p$) for frame propagation and reception of acknowledgement is smaller than frame transmission time (τ_t). The propagation delay τ_p is given by

$$\tau_p = \frac{d}{c}$$

where c is speed of light and d is the distance between transmitter and receiver. The transmission delay τ_t is given by

$$\tau_t = \frac{L}{\lambda}$$

Where L is the number of bits in a frame and λ is the transmission rate in bits per second.

Thus, ARQ protocols are efficient and useful when we have

$$2\tau_p << \tau_t \quad ...(1)$$

If the above inequality is not true, then forward error correction (FEC) techniques should be used.

When the sender transmits a frame on the forward channel, the receiver checks it for errors. If there are no errors, the receiver acknowledges the correct transmission by sending an acknowledge (ACK) signal through the feedback channel. In that case, the transmitter proceeds to send the next frame. If there were errors in the received frame, the receiver sends a negative acknowledgement signal (NAK) and the sender sends the same frame again. If the receiver does not receive ACK or NAK signals due to some problem in the feedback channel, the receiver waits for a certain timeout period and sends the frame again.

Based on the above discussion, we conclude that the time between transmitted frames is equal to $2\tau_p$, where τ_p is the one-way propagation delay.

Fig.1 shows an example of transmitting several frames using stop-and-wait ARQ. Frame 1 was correctly received as indicated by the ACK signal and the sender starts sending frame 2.

Frame 2 was received in error as indicated by NAK and the grey line. The transmitter sends frame 2 one more time. For some reason, no acknowledgement signals were received (indicated by short grey line) and the sender sends frame 2 for the third time after waiting for the proper timeout period.

Frame 2 was received correctly as indicated by the ACK signal and the sender starts sending frame 3.

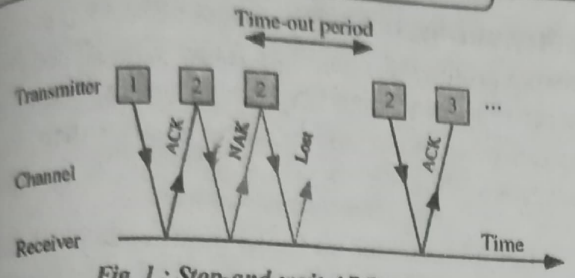


Fig. 1 : Stop-and-wait ARQ protocol

Modelling Stop and Wait ARQ: We perform Markov chain analysis of the stop-and-wait algorithm. We make the following assumptions for our analysis of the stop-and-wait ARQ (SW ARQ):

1. The average length of a frame is n bits.
2. The forward channel has random noise and the probability that a bit will be received in error is $\in$. Another name for $\in$ is bit error rate (BER).
3. The feedback channel is assumed noise-free so that acknowledgement signals from the receiving station will always be transmitted to the sending station.
4. The sender will keep sending a frame until it is correctly received.

The state of the sender while attempting to transmit a frame depends only on the outcome of the frame just sent. Hence we can represent the state of the sender as a Markov chain having the following properties :

1. State i of the Markov chain indicates that the sender is retransmitting the frame for the i^{th} time. State 0 indicates error-free transmission.
2. The number of states is infinite since no upper bound is placed on the number of retransmissions.
3. The time step is taken equal to the sum of transmission delay and round-trip delay $T = \tau_1 + 2\tau_p$.

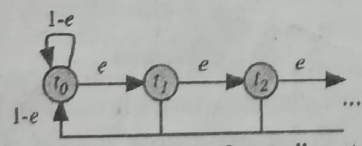


Fig. 2 : State transition diagram of a sending station using the SW ARQ error control protocol

The state transition diagram for the SW ARQ protocol is shown in fig.2. In the fig.2, e represents the probability that the transmitted frame contained an error, e is given by the expression

$$e = 1 - (1 - \in)^n \qquad ...(2)$$

For a noise-free channel $\in = 0$ and so $e = 0$. When the average number of errors in a frame is very small (i.e., $\in n << 1$), we can write

$$e \approx \in n \qquad ...(3)$$

The quantity $\in n$ is an approximation of the average number of bits in error in a frame. Naturally, we would like the number of error to be small so as not to waste the bandwidth in retransmissions. Thus, we must have

$$e = \in n << 1 \qquad ...(4)$$

Equation (2) assumed no forward error correction (FEC) coding is implemented.

We organize the state distribution vector as follows :

$$s_i = [s_0\ s_1\ s_2\ \ldots]^t \qquad ...(5)$$

Where s_i corresponds to retransmitting the frame for the i^{th} time, s_0 corresponds to transmitting the frame once with zero retransmissions. This is the case when the frame was correctly received without having to retransmit it.

The corresponding transition matrix of the channel is given by

$$P = \begin{bmatrix} 1-e & 1-e & 1-e & 1-e & \cdots \\ e & 0 & 0 & 0 & \cdots \\ 0 & e & 0 & 0 & \cdots \\ 0 & 0 & e & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad ...(6)$$

At equilibrium, the distribution vector is obtained by solving the two equations

$$Ps = s \qquad ...(7)$$

$$\sum s = 1 \qquad ...(8)$$

The solution to the above two equations is simple :

$$s = (1 - e) \times [1\ e\ e^2\ \ldots]^t \qquad ...(9)$$

SW ARQ Performance

The average number of retransmissions for a frame is given by

$$N_t = (1 - e) \times (s_1 + 2s_2 + 3s_3 + \ldots)$$

$$= (1-e)^2 \times \sum_{i=0}^{\infty} i\, e^i$$

$$= e \text{ transmissions/frame} \qquad ...(10)$$

For a noise-free channel, e = 0 and the average number of retransmissions is also 0. This indicates that a frame is sent once for a successful transmission.

For a typical channel, $e \approx \in n << 1$ and the average number of transmissions can be approximated as

$$N_t \approx \in n \qquad ...(11)$$

We define the efficiency of the SW ARQ protocol as the inverse of the total number of transmissions which includes the first transmission plus the average number of retransmissions. In that case, η is given by

$$\eta = \frac{1}{1+N_t} = \frac{1}{1+e} \qquad ...(12)$$

For an error-free channel, $N_t = 0$ and $\eta = 100\%$. For a typical channel, $e \approx \in n << 1$ and the efficiency is given by

$$\eta \approx 1 - \in n \qquad ...(13)$$

This indicates that the efficiency decrease with increase in bit error rate or frame size. Thus, we see that the system performance will degrade gradually with any increase in the number of bits in the frame or any increase in the frame error probability.

The throughput of the transmitter can be expressed as

$$Th = \eta = 1 - e \text{ frame/time step} \quad ...(14)$$

Thus, for an error-free channel, $\eta = 1$ and arriving frames are guaranteed to be transmitted on the first try. We could have obtained the above expression for the throughput by estimating the number of frames that are successfully transmitted in each transmitter state :

$$Th = (1-e)\sum_{i=0}^{\infty} s_i = 1 - e \quad ...(15)$$

When errors are present in the channel, then $\eta < 1$ and so is the system throughput.

Piggy backing Method : In two-way communication, whenever a data frame is received, the receiver waits and does not send the control frame (acknowledgement or ACK) back to the sender immediately. The receiver waits until its network layer passes in the next data packet. The delayed acknowledgement is then attached to this outgoing data frame. This technique of temporarily delaying the acknowledgement so that it can be hooked with next outgoing data frame is known as piggy backing.

Whenever party A wants to send data to party B, it will send the data along with this ACK field. Considering the sliding window here of size 8 bits, if A has send frames up to 5 correctly (from B), and wants to send frames starting from frame 6, it will send ACK6 with the data. Three rules govern the piggy backing data transfer.

- If station A wants to send both data and an acknowledgement, it keeps both fields there.
- If station A wants to send just the acknowledgement, then a separate ACK frame is sent.
- If station A wants to send just the data, then the last acknowledgement field is sent along with the data. Station B simply ignores this duplicate ACK frame upon receiving.

Q.24 ***Explain linear codes and how error you detect and correct using linear code techniques.***

[R.T.U. Dec. 2013]

Ans. Linear Codes : A code is called as linear if any two code words in the code can be added in modulo arithmetic to generate a third code word in the code.

Structure of Linear Code

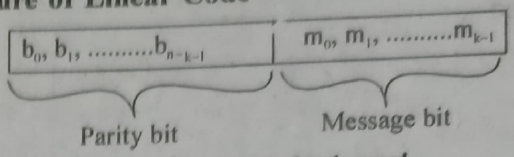


Fig. : Structure of code word

Let us consider an (n, k) linear code in which k bits of the *n* code bits are always identical to the message sequence to be transmitted. The (n – k) bits in the remaining portion are computed from the message bits in accordance with a prescribed encoding rule which determines the mathematical structure of the code. Accordingly these (n - k) bits are reffered as generalized parity check bits or simply parity bits.

Technique of Linear Code to Detect and Correct Errors: Let $b_0, b_1, \ldots b_{n-k-1}$ denote the (n–k) parity bits in the code word. Let the $m_0, m_1, \ldots m_{k-1}$ constitute a block of k arbitary messages bits. For the code to process a systematic structure, a code word is divided into two parts, one of which is occupied by the message bits and the other bits the parity bits.

We may therefore write

$$c_i = \begin{cases} b_i, & i = 0,1...n-k-1 \\ m_{i+k-n}, & i = n-k,\ n-k+1...n-1 \end{cases} \quad ... (1)$$

The (n – k) parity bits are linear sum of the k message bits, as shown by the following generalized relation

$$b_i = p_{oi}m_0 + p_{1i}m_1 + ...p_{k-1}, i_{k-1}^m \quad ... (2)$$

Where the coefficients are defined as follows

$$p_{ij} = \begin{cases} 1 & \text{if } b_i \text{ depends on } m_j \\ 0 & \text{elsewhere} \end{cases} \quad ... (3)$$

The coefficients p_{ij} are chossen in such a way that the rows of the generator matrix are linearly independent and the parity equation are unique.

The system of equation (1) and (2) defines the mathematical structure of the (n,k) linear code. This system of equation may be rewritten in a compact form using matrix notation. To proceed with bits reformulation, we define the 1 by k message vector m, the 1 by (n-k) parity vector b, and 1 by n code vector c as follows.

$$m = [m_0, m_1, ...m_{k-1}] \quad ... (4)$$

$$b = [b_0, b_1, ...b_{n-k-1}] \quad ... (5)$$

$$c = [c_0, c_1, ...c_{n-1}] \quad ... (6)$$

we may thus rewrite the set of simultaneous equation defining the parity bits in the compact matrix form :

$$b = mP \quad ... (7)$$

where P is the k by (n-k) coefficients matrix and is denoted as

$$P+=\begin{bmatrix} P_{00} & P_{01} & P_{0,n-k-1} \\ P_{10} & P_{11} & P_{1,n-k-1} \\ \vdots & \vdots & \vdots \\ P_{k-1,0} & P_{k-1,1} & P_{k-1,n-k-1} \end{bmatrix} \quad \ldots (8)$$

where p_{ij} is 0 or 1. From equation (4) and (6), it may be observed that c may be expressed as a partitioned row vector in terms of the vectors m and b us under

$$c = [b: m] \quad \ldots (9)$$

Therefore, substituting equation (7) in equation (9) and factoring out the common message vector m, we have

$$c_1 = [I_k : P] \quad \ldots (10)$$

where I_k is the k by identity matrix and is given by

$$I_k = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix} \quad \ldots (11)$$

Now let us define the k by n generator matrix as follows

$$G = [I_k : P] \quad \ldots (12)$$

Similarly $c = mG \quad \ldots (13)$

There is another way of expressing the relationship between the message bit and parity check bits of a linear code. Let H denote an (n-k) by n matrix, which is defined as

$$H = [P^T : I_{n-k}] \quad \ldots (14)$$

where P^T is an (n-k) by k matrix representing the transpose of the coefficients matrix P and I_{n-k} is the (n-k) identity matrix.

Example Given $= \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right]$

Find out all the possible code vectors.

Solution: $G = [I_k : P]_{k \times n}$

so k = 3 are message bits and $2^3 = 8$ possible combination of code words which is given as

(000) (001) (010) (011) (100) (101) (110) (111)

we know that

$$C = mG$$

for $m = 111$

$$(111)\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right] = (111000)$$

Like wise

m(message)	Code	Vectors
000	000	000
001	0001	110
010	010	101
011	011	011
100	100	011
101	101	101
110	110	110
111	111	000

Parity check matrix H is a matrix and it is associated with each (n,k) block code. H can be written as

$$H = \left[\begin{array}{ccc|ccc} P_{11} & P_{21} & P_{k1} & 1 & 0 & 0 \\ P_{12} & P_{22} & P_{k2} & 0 & 1 & 0 \\ P_{1(n-k)} & P_{2(n-k)} & P_{k(n-k)} & 0 & 0 & 1 \end{array}\right]_{(n-k)\times n}$$

The parity check matrix is used to varity the code word c generated by generator matrix. G = [I_k: P]. This verification can be done by following steps.

Step 1. c is a correct code word in (n,k) block code, generated by G is and only if

$$cH^T = 0$$

where H^T is transfer of parity check matrix. Thus generator matrix is used in encoding operation where as the parity check matrix is used for decoding operation.

Step 2. If c be the codeword that was transmitted over a noisy channel and let R be the noise corrupted vector that was received. The vector R is the combination of code vector c and error vector E.

$$R = c + E$$

Step 3. The receiver does the decoding operation by determining an (n-k) vectors. The networks is called error syndrome

$$S = RH^T$$
$$S = (c+E)H^T$$
$$S = EH^T$$

Thus we can say, syndrom of received vector is zero, if R is a valid code word.

Step 4. If error occur then S of received H^T will be non zero. If $S \neq 0$, than S is compared with H^T. The row with this syndrome indicates the position where error is present.

Q.25 ***(a) What is CSMA/CD? Explain.***

(b) Compare throughput of CSMA & CSMA/CD.

[R.T.U. 2013]

Ans.(a) CSMA with Collision Detection (CSMA/CD) : Persistent and nonpersistent CSMA protocols are

clearly an improvemen. over ALOHA because they ensure that no station begins to transmit when it senses the channel busy. But when two frames collide, the medium remains unusable for the duration of transmission of both damaged frames. For long frames, compared to propagation time, the amount of waste capacity can be considerable. In CSMA with collision detection (CSMA/CD) this wastage of capacity is also removed by aborting the transmission as soon as stations detect a collision. In CSMA/CD, stations continuously listen to the medium while transmitting. This leads to the following rules for CSMA/CD :

1. If the medium is idle, transmit; otherwise, go to step 2.
2. If the medium is busy, continue to listen until the channel is idle, then transmit immediately.
3. If a collision is detected during transmission, transmit a brief jamming signal to assure that all stations know that there has been a collision and then cease transmissions.
4. After transmitting the jamming signal, wait a random amount of time, then attempt to transmit again. (Repeat from step 1)

Figure 1, illustrates the technique for a baseband bus. At time t_0, station A begins transmitting a packet addressed to D. At t_1, both B and C are ready to transmit. B senses a transmission and so defers. C, however, is still unaware of A's transmission and begins its own transmission. When A's transmission reaches C, at t_2, C detects the collision and ceases transmission. The effect of the collision propagates back to A, where it is detected some time later, t_3, at which time A ceases transmission.

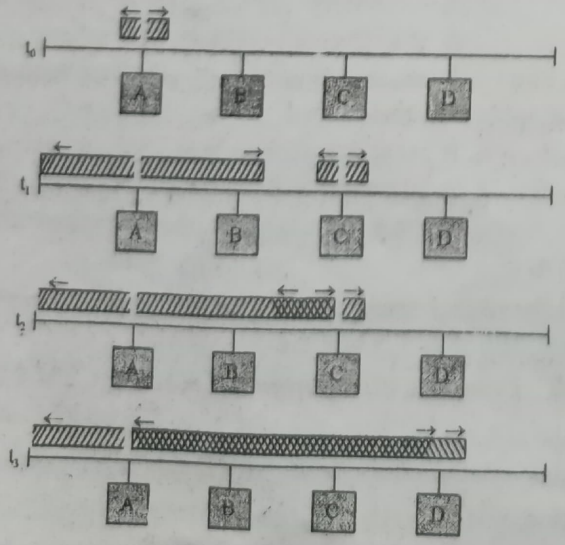


Fig. 1 : CSMA/CD Operation

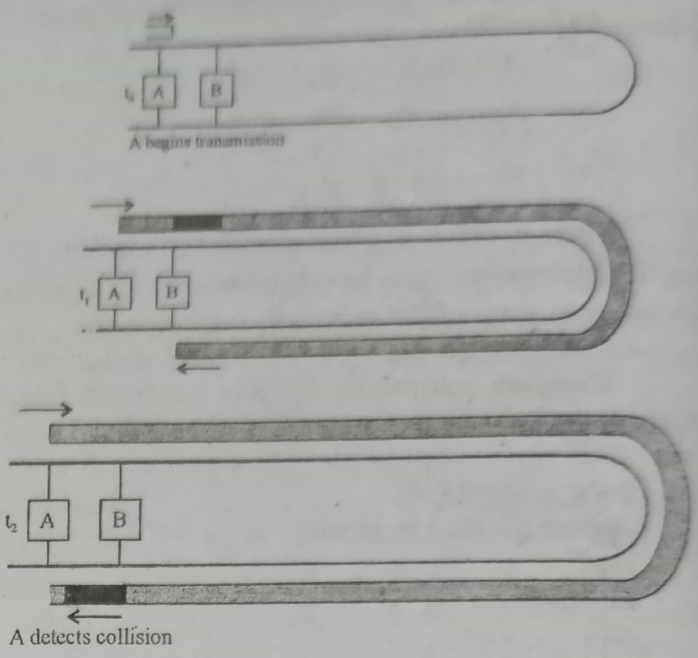


Fig. 2 : Broadband collision detection timing

With CSMA/CD, the amount of wasted capacity is reduced and it depends on the time it takes to detect a collision. How long does it takes? Let us first consider the case of a baseband bus and consider two stations as far as possible. For example in fig.2 suppose that station A begins a transmission and that just before that transmission reaches D, D is ready to transmit. Because D is not yet aware of A's transmission, it begins to transmit. A collision occurs almost immediately and is recognized by D. However, the collision must propagate all the way back to A's before A is aware of the collision. By this line of reasoning, we conclude that the amount of time that it takes to detect a collision is no greater than twice the end to end propagation delay. For a broadband bus, the delay is even longer. Fig.2 shows a dualcable system. This time, the worse case occurs for two stations as close together as possible and as far as possible from the headend. In this case, the maximum time to detect a collision is four times the propagation delay from one end of the cable to the head end.

B begins transmission just before leading edge of A's packet arrives at B's receiver, B almost immediately detects A's transmission and ceases its own transmission.

Ans.(b) Throughput of CSMA : All of the variations of CSMA are sensitive to the end-to-end propagation delay of the medium that constitutes the vulnerable period. An analysis of the throughput S versus load G for CSMA is beyond the scope of this text. Figure 1 shows the throughput S versus G for 1-Persistent and Non-Persistent CSMA for three values of a.

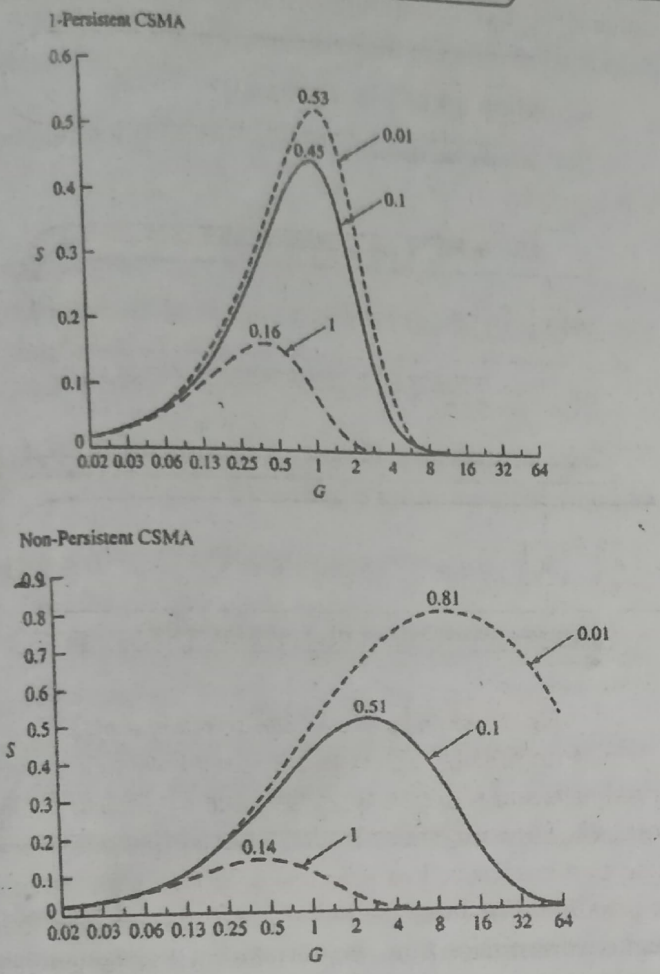


Fig. 1 : Throughput S versus load for 1-Persistent and Non-Persistent CSMA. The curves are for different values of a.

It can be seen that the throughput of 1-Persistent CSMA drops off much more sharply with increased G. It can also be seen that the normalized propagation delay $a = t_{prop}/X$ has a significant impact on the maximum achievable throughput.

Non-Persistent CSMA achieves a higher throughput than 1-Persistent CSMA does over a broader range of load values G. For very small values of a, Non-Persistent CSMA has a relatively high maximum achievable throughput. However, as a approaches 1, both 1-Persistent and Non-Persistent CSMA have maximum achievable throughput that are even lower than the ALOHA schemes.

Through of CSMA/CD

The maximum throughput in the CSMA-CD systems occurs when all of the channel time is spent in frame transmissions followed by contention intervals. Each frame transmission time X is followed by a period t_{prop} during which stations find out that the frame transmission is completed and then a contention interval of average duration $2e\ t_{prop}$ and, so the maximum throughput is then

$$\rho_{Max} = \frac{X}{X + t_{prop} + 2et_{prop}}$$

$$= \frac{1}{1+(2e+1)a} = \frac{1}{1+(2e+1)Rd/vL} \quad ...(i)$$

where $a = t_{prop}/X$ is the propagation delay normalized to the frame transmission time. The rightmost term in the above expression is in terms of the bit rate of the medium R, the diameter of the medium d, the propagation speed over the medium v, and the frame length L. The expression shows that CSMA-CD can achieve throughput that are close to 1 when a is much smaller than 1. For example, if $a = 0.01$, then CSMA-CD has a maximum throughput of 94 percent. The CSMA-CD scheme provides the basis for the Ethernet LAN protocol.

As the load approaches the maximum throughput in equation (i), the average transfer delay increases as collisions occur more frequently. It should be emphasized that CSMA-CD does not provide an orderly transfer of frames. The random backoff mechanism and the random occurrence of collisions imply that frames need not be transmitted in the order that they arrived. Indeed once a frame is involved in a collision, it is quite possible for other later arrivals to be transferred ahead of it. This behavior implies that in CSMA-CD frame delays exhibit greater variability than in first-in-first-out statistical multiplexers.

Figure 2 shows a comparison of the maximum throughput of the main random access MAC techniques discussed in this section. For small values of a, CSMA-CD has the highest maximum throughput followed by CSMA. As a increases, the maximum throughput of CSMA-CD and CSMA decrease since the reaction times are approaching the frame transmission times. As a approaches 1, the throughput of these schemes becomes less than that of ALOHA and slotted ALOHA. As we know, ALOHA and slotted ALOHA are not sensitive to a since their operation does not depend on the reaction time.

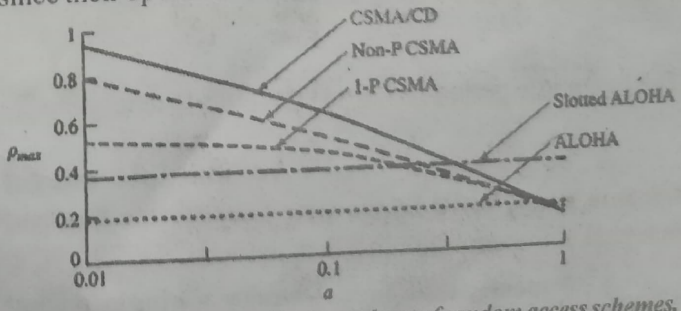


Fig. 2 : Maximum achievable throughput of random access schemes.

Q.26 ***Derive the expression for throughtput of an unslotted CSMA carrier. Enumerate all the assumptions.*** *[R.T.U. 2013]*

Ans. For fixed packet lengths, the packet transmission time is one slot.

The propagation delay between two users is t seconds, and the normalized propagation delay, a is given by

$$a = \frac{\tau}{l/C} = \frac{\tau C}{l}$$

where C is the channel capacity in bit/s and l is the packet length in bits. The status of the channel alternates with busy and ideal periods.

Let

B_n be the busy period in the n^{th} cycle,

I_n be the idle period in the n^{th} cycle, and

u be the time that the channel is used without collision.

Let $\overline{X}$ denotes the mean value of X. The success rate S is then given by

$$S = \frac{\overline{u}}{\overline{B} + \overline{I}} \quad \ldots (1)$$

The parameters $\overline{u}$, $\overline{I}$ and $\overline{B}$ are determined as follows:

Let

P_s = P[busy period has a single packet transmission]

= P[no transmission in normalized interval a]

or $P_s = e^{-aG}$. ... (2)

The mean time that the channel is used without collision is given by

$$\overline{u} = \text{packet transmission time} \times P_s$$

or $$\overline{u} = 1 \times P_s$$

The mean idle period is $\overline{I} = \frac{1}{G}$. It remins to determine $\overline{B}$.

A busy period can have multiple packet transmissions. Let Y be the time between the first and last packet transmission in a busy period.

Y=0 means that there is only a single packet transmission in the busy period so that the transmission is successful. If Y is equal to or longer than the normalized propagation delay a, there is no collision.

Collision takes place when $0 < Y < a$.

For this case, the cumulative distribution function of Y is given by

$$F_Y(y) = P[Y \le y]$$

or $F_Y(y)$ = P[no packet arrives in (y, a)]

or $$F_Y(y) = e^{-(a-y)G} \qquad 0 \le y \le a \ \ldots (3)$$

The probability density function of Y, $F_Y(y)$, is obtained differentiating equation (3)

$$F_Y(y) = e^{-aG}\,\delta(y) + Ge^{-(a-y)G}, \quad 0 \le y \le a$$

Hence, mean value of Y is given by

$$\overline{Y} = \int_0^\infty y F_y(y)\,dy$$

or $$\overline{Y} = a - \frac{1 - e^{-aG}}{G}$$

The mean busy period is given by the sum of the packet transmission time, the normalized propagation delay, and the mean interval between the first and last packet transmission instants.

Mathematically, this means that

$$\overline{B} = 1 + a + Y$$

The throughput for unsoltted non-persistent CSMA is obtained by substitution of $\overline{u}$, $\overline{I}$ and $\overline{B}$ in equation (1), yielding

$$S_{np-CSMA} = \frac{Ge^{-aG}}{G(1+2a) + e^{-aG}}$$

In the limit when $a \to 0, S \to \frac{G}{(1+G)}$

The throughput for unsoltted 1-persistent CSMA can be derived in a similar manner as under :

$$S_{1,-CSMA} = \frac{Ge^{-G(1+2a)}\left[1 + G + aG\left(1 + G + aG/2\right)\right]}{G(1+2a) - \left(1 - e^{-aG}\right) + (1+aG)e^{-G(1+a)}}$$

In the limit as $a \to 0, S \to \frac{Ge^{-aG}(1+G)}{G + e^{-aG}}$

Also, the mean packet delay of CSMA is aproximately given by

$$\overline{D}_{CSMA} = R + \left(\frac{G}{S} - 1\right)\left(R + \overline{D}_r\right)$$

Q.27 ***The parity check matrix of a particular (7, 4) linear block code is given by***

$$H = \begin{bmatrix} 1110100 \\ 1101010 \\ 1011001 \end{bmatrix}$$

(a) Find the generator matrix G .

(b) List all the code vectors.

(c) What is the minimum distance between the code vectors ?

(d) How many errors can be detected and how many can be corrected ? ***[R.T.U. 2012]***

Ans.
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & : & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & : & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & : & 0 & 0 & 1 \end{bmatrix} \quad \ldots (1)$$

$$H = [P^T : I_3] \quad \ldots(2)$$

Comparing matrix (1) & (2)

$$P^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$P = \left[P^T\right]^T$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{4\times3}$$

(a) Generator Matrix (G)

$$G = [I_K : P_{k\times q}]_{k\times n}$$

but $k = 4, q = 7, n = 7$

$$G = [I_4 : P_{4\times3}]_{4\times7}$$

so

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & : & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & : & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & : & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & : & 0 & 1 & 1 \end{bmatrix}_{4\times7}$$

(b) Code Word

$$C = MP$$

$$[C_1\ C_2\ C_3], = [M_1, M_2, M_3, M_4] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$C_1 = M_1 \oplus M_2 \oplus M_3$$

$$C_2 = M_1 \oplus M_2 \oplus M_4$$

$$C_3 = M_1 \oplus M_3 \oplus M_4$$

Code table

5.3.2/5-20.

(c) Minimum distance: It is equal to minimum weight of any non-zero code vector. So from the table

$$d_{min} = [w(x)]_{min.}$$

$$\boxed{d_{min} = 3}$$

(d) Error Detection & Correction

$$d_{min} \geq s + 1$$

$$3 \geq s + 1$$

$$s \leq 2$$

two errors will be detected.

$$d_{min} \geq 2t + 1$$

$$3 \geq 2t + 1$$

$$t \leq 1$$

only one error will be corrected.

❑❑❑

NETWORK LAYER 3

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 ***Differentiate between IPv4 address and IPv6 address.*** *[R.T.U. 2019]*

Ans. Differences between IPv4 and IPv6

IPv4	IPv6
IPv4 addresses are 32 bit length.	IPv6 addresses are 128 bit length.
IPv4 addresses are represented in decimal format	IPv6 addresses are represented in hexadecimal format.
IPSec support is optional.	Inbuilt IPSec support.
Checksum filed is available in IPv4 header.	No Checksum field in IPv6 header.
Fragmentation is done by hosts and routers.	Fragmentation is done by sender hosts only.
Packet size is 576 bytes	Packet size is 1280 bytes
Router discovery is optional in IPv4.	Router discovery is required in IPv6.
IPv4 uses broadcast.	IPv6 do not use broadcast

Q.2 ***What is address mapping.***

Ans. The delivery of a packet to a host or a router requires two levels of addressing : logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. This can be done by using either static or dynamic mapping.

Q.3 ***Explain ARQ in data transmission.***

Ans. Automatic repeat request (ARQ) is a protocol for error control in data transmission. When the receiver defeats an error in a packet it automatically requests the transmitter to resend the packets. This process is repeated until the packet is error free or the error continues beyond a predetermined number of transmission.

PART-B

Q.4 ***Explain the services provided by network security.*** *[R.T.U. 2019]*

Ans. Network Security Services

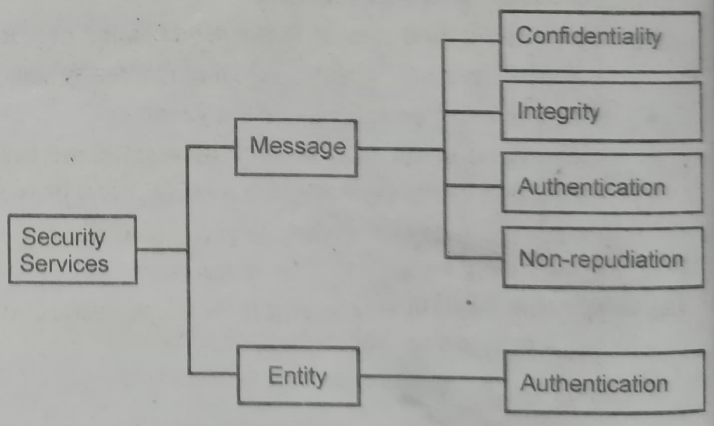


Fig.

1. Message confidentiality

- It means that the content of a message when transmitted across a network must remain confidential, i.e. only the intended receiver and no one else should be able to read the message.

- The users, therefore, want to encrypt the message they send so that an eavesdropper on the network will not be able to read the contents of the message.

2. Message Integrity

- It means the data must reach the destination without any adulteration i.e. exactly as it was sent.
- There must be no changes during transmission, neither accidentally nor maliciously.
- Integrity of a message is ensured by attaching a checksum to the message.
- The algorithm for generating the checksum ensures that an intruder cannot alter the checksum or the message.

3. Message Authentication

- In message authentication the receiver needs to be sure of the sender's identity i.e. the receiver has to make sure that the actual sender is the same as claimed to be.
- There are different methods to check the genuineness of the sender :

1. The two parties share a common secret code word. A party is required to show the secret code word to the other for authentication.
2. Authentication can be done by sending digital signature.
3. A trusted third party verifies the authenticity. One such way is to use digital certificates issued by a recognized certification authority.

4. Message non-repudiation

- Non-repudiation means that a sender must not be able to deny sending a message that it actually sent.
- The burden of proof falls on the receiver.
- Non-repudiation is not only in respect of the ownership of the message; the receiver must prove that the contents of the message are also the same as the sender sent.
- Non-repudiation is achieved by authentication and integrity mechanisms.

5. Entity Authentication

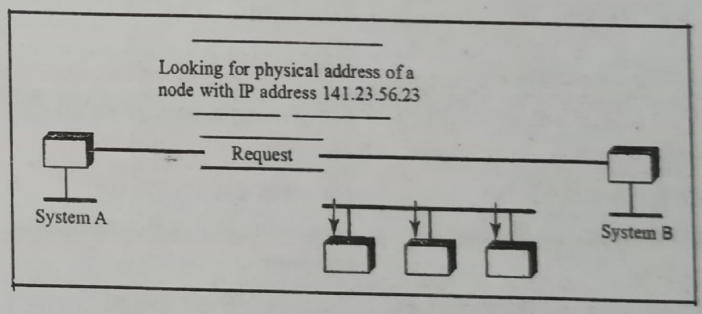
- In entity authentication (or user identification) the entity or user is verified prior to access to the system resources .

Q.5 ***Explain ARP and RARP address mapping protocols.***

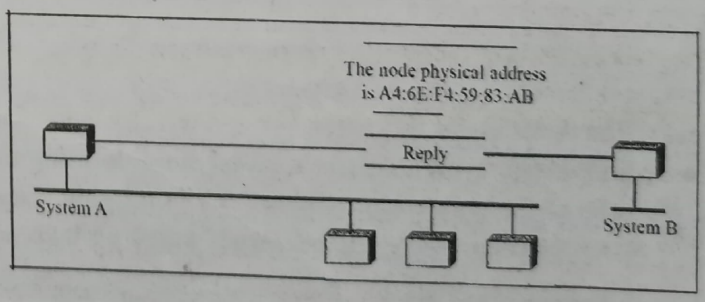
[R.T.U. 2019]

Ans. ARP : Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network (see Figure 1).

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses. The packet is unicast directly to the inquirer by using the physical address received in the query packet.



(a) ARP request is broadcast



(b) ARP reply is unicast

Fig. 1 : ARP operation

In Figure 1(a), the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IF address of 141.23.56.23.

This packet is received by every system on the physical network, but only system B will answer it, as shown in Figure1(b). System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination by using the physical address it received.

Cache Memory : Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B. It could have broadcast the IP packet itself. ARP can be useful if the ARP reply is cached (kept in cache memory for a while) because a system normally sends several packets to the same destination. A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted. Before sending an ARP request, the system first checks its cache to see if it can find the mapping.

Packet Format

Figure 2 shows the format of an ARP packet.

32 bits

8 bits	8 bits	16 bits
Hardware Type		Protocol Type
Hardware length	Protocol length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 6 bytes for Ethernet) (It is not filled in a request)		
Target protocol address (For example, 4 bytes for IP)		

Fig. 2 : ARP packet

The fields are as follows:

- **Hardware type :** This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network.
- **Protocol type :** This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 0800_{16}, ARP can be used with any higher-level protocol.
- **Hardware length :** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- **Protocol length :** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
- **Operation :** This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).
- **Sender hardware address :** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- **Sender protocol address :** This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
- **Target hardware address :** This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is allOs because the sender does not know the physical address of the target.
- **Target protocol address :** This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

RARP : The **Reverse Address Resolution Protocol (RARP)** is a computer networking protocol used by a host computer to request its Internet Protocol (IPv4) address from an administrative host, when it has available its Link Layer or hardware address, such as a MAC address.

RARP allows a physical machine in a local area network to request its IP address from a gateway server's ARP table or cache. A network administrator creates a table in a local area network's gateway router that maps the physical machine (or Media Access Control - MAC address) addresses to corresponding Internet Protocol addresses (IP address). When a new machine is set up, its RARP client program requests from the RARP server on the router to be sent its IP address. Assuming that an entry has been set up in the router table, the RARP server will return the IP address to the machine which can store it for future use. RARP was limited to serving only IP addresses. RARP requires one or more server hosts to maintain a database of mappings of link layer addresses to their respective protocol addresses. MAC addresses needed to be individually configured on the servers by an administrator.

Q.6 ***Write short note on Mobile IP.*** ***[R.T.U. 2016]***

Ans. Mobile IP is an open standard that allows users to move from one network to another without loosing connectivity. Mobile devices have IP addresses that are associated with one network and moving to another network that means

changing the IP address. Using the mobile IP system will allow users to achieve this and at the same time make the underlying process transparent for a user. It has been foreseen that mobile computing devices will become more pervasive, more useful, and more powerful in the future. In IP networks, routing is based on stationary IP addresses, similar to how a postal letter is delivered to the fixed address on the envelope. A device on a network is reachable through normal IP routing by the IP address it is assigned on the network. The problem occurs when a device roams away from its home network and is no longer reachable using normal IP routing. This results in the active sessions of the device being terminated.

Mobile IP was created to enable users to keep the same IP address while travelling to a different network (which may even be on a different wireless operator), thus ensuring that a roaming individual could continue communication without sessions or connections being dropped. Because the mobility functions of Mobile IP are performed at the network layer rather than the physical layer, the mobile device can span different types of wireless and wireline networks while maintaining connections and ongoing applications.

Remote login, remote printing, and file transfers are examples of applications where it is desirable not to interrupt communications while an individual roams across network boundaries. Also, certain network services, such as software licenses and access privileges, are based on IP addresses.

A device that can roam while appearing to a user to be at its home network is called a mobile node. Examples of mobile nodes include: a personal digital assistant, a laptop computer, or a data-ready cellular phone—that can change its point of attachment from one network or subnet to another. This mobile node can travel from link to link and maintain communications using the same IP address. There is no need for any changes to applications, because the solution is at the network layer, which provides the transparent network mobility.

Mobile IP has the following three components:

1. **Mobile Node:** The mobile node is a device such as a cell phone, personal digital assistant, or laptop whose software enables network roaming capabilities.
2. **Home Agent:** The home agent is a router on the home network serving as the anchor point for communication with the mobile node; it tunnels packets from a device on the Internet, called a correspondent node, to the roaming mobile node.
3. **Foreign Agent:** The foreign agent is a router that may function as the point of attachment for the mobile node when it roams to a foreign network, delivering packets from the home agent to the mobile node.

The mobile access router functions similarly to the mobile node with one key difference—the mobile access router allows entire networks to roam. For example, an airplane with a mobile access router can fly around the world while passengers stay connected to the Internet. This communication is accomplished by Mobile IP aware routers tunneling packets, which are destined to hosts on the mobile networks, to the location where the mobile access router is visiting. The mobile access router then forwards the packets to the destination device.

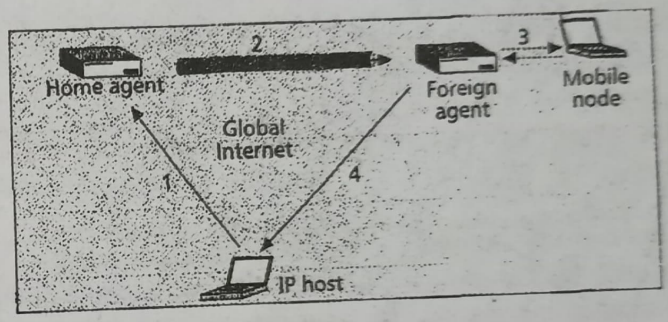


Fig. : Mobile IP datagram flow

Q.7 ***Explain the following protocols :***

(i) RARP Vs BOOTP

(ii) POP3 Vs IMAP *[R.T.U. 2015, 13]*

Ans. (i) RARP : *Refer to Q.5.*

BOOTP : Bootstrap protocol (BOOTP) is a client-server protocol designed to provide the four pieces of information :

- IP address of a computer
- Its subnet mask
- IP address of a router
- IP address of a name server

BOOTP provides this information for a diskless computer or a computer that is booted for the first time. RARP provides only the IP address for a diskless computer, not the other information. If we use BOOTP, we do not need RARP.

BOOTP is a not a dynamic configuration protocol configuration protocol. When a client requests its IP address, the BOOTP server searches a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client should already exist. The binding is predetermined.

However, what if a host moves from one physical network to another? What if a host wants a temporary IP address? BOOTP cannot handle these problems because the

binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. BOOTP is a static configuration protocol.

Ans. (ii) POP3 Vs IMAP : Both POP3 and IMAP are email assuming protocols but they differ in following manner :

POP3 (Post Office Protocol Version 3)	IMAP (Internet Message Accessing Protocol)
Message stores at user's (client) machine.	Message stores at server machine.
Message stores at client machine so it provides offline reading facility.	Online message reading facility due to message stores at server.
Because message stores at client machine so user takes backup.	Backup facility provided by email service provider.
Memory needed at clients machine to store the message.	Memory not needed at client side due to message stores at server.
User can access email through his/her own computer only.	User can access email through any machine in the world *i.e.* it provides portability.

Q.8 How are IP addresses assigned? Describe this with suitable example for internet. *[R.T.U. 2015]*

OR

How are IP addresses assigned ? Describe this with suitable example for internet currently in use. *[Raj. Univ. 2004]*

Ans. Every host and router on the internet has an IP address, which encodes its network number and host number. The combination is unique, in principle, no two machines on the internet have the same IP address. All IP addresses are 32 bits long and are used in the source address and destination address fields of IP packets. An IP address does not actually refer to a host. It really refers to a network interface, so if a host is on two networks, it must have two IP addresses. However, in practice, most hosts are on one network and thus have one IP address. IP addresses were divided into the five categories listed in the fig. This allocation has come to be called classful addressing. The class A, B, C and D formats allow for upto 128 networks with 16 million host each, 16384 networks with up to 64K hosts and 2 million networks (e.g. LANs) with up to 256 hosts each (although a few of these are special). Also supported is multicast in which a datagram is directed to multiple hosts. Addresses beginning with 1111 are reserved for future use. Over 500,000 networks are now connected to the internet.

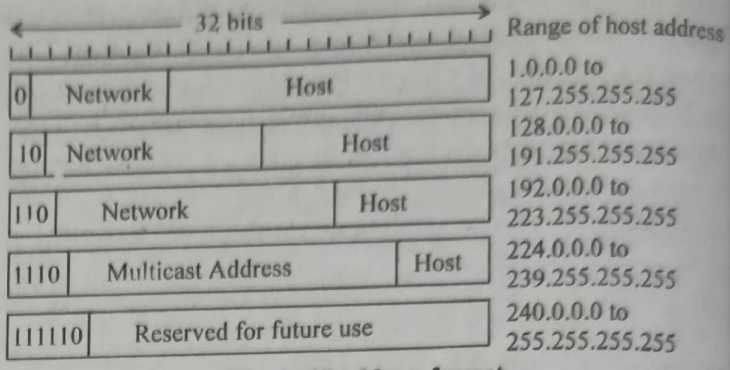


Fig. 1 : IP address format

Network addresses, which are 32 bit numbers, are usually written in dotted decimal notation. In this format, each of the 4 bytes are written in decimal, from 0 to 255. E.g. the 32 bit hexadecimal address C0290614 is written as 192.41.6.20. The lowest IP address is 0.0.0.0 and the highest 255.255.255.255.

The value 0 and 1(all 1's) have special meanings, as shown in fig. 2. The value 0 means this network or this host. The value of 1 is used as a broadcast address to mean all hosts on the indicated network.

The IP address 0.0.0.0 is used by hosts when they are being booted IP addresses with 0 as network number refer to the current network. These addresses allow machines to refer to their own network without knowing its number (but they have to know its class to know how many 0's to include). The address consisting of all 1's allows broadcasting on the local network typically a LAN. The addresses with a proper network number and all 1's in the host field allow machines to send broadcast packets to distant LANs anywhere in the internet (although many network administrators disable this feature). Finally, all addresses of the form 127.xx.yy.zz are reserved for loopback testing. Packets sent to that address are not putout onto the wire, they are processed locally and treated as incoming packets. This allows packets to be sent to the local network without sender knowing its number.

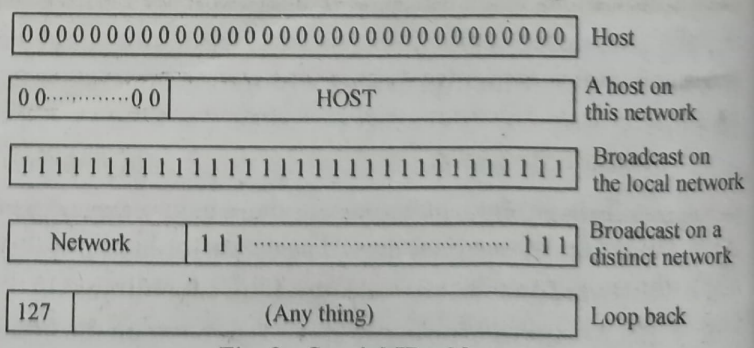


Fig. 2 : Special IP addresses

Q.9 IP, TCP and UDP all discard a packet that arrives with a checksum error and do not attempt to notify the source. Why ? *[R.T.U. 2009]*

Ans. In the protocols, acknowledgements of frames received and permission to send new frames were tied together. This was a consequence of a fixed window size for each protocol. In TCP, acknowledgements and permission to send additional data are completely decoupled. In effect, a receiver can say I have received bytes up through k but I do not want any more just now. This decoupling gives additional flexibility. A checksum is also provided for extra reliability. It checksums the header, the data and the coneptual pseudoheader. When performing this computation, the TCP checksum field is set to zero and the data field is paded out with an additional zero byte if its length is an add number. The checksum algorithm is simply to add up all the 16 bit words in one's complement and then to take the one's complement of the sum. As a consequence, when the receiver performs the calculation on the entire segment, including the checksum field, the result should be zero.

The pseudoheader contains the 32 bit IP addresses of the source and destination machines, the protocol number for TCP (6), and the byte count for the TCP segment (including the header). Including the pseudoheader in the TCP checksum computation helps detect misdelivered packets, but including it also violates the protocol hierarhcy since the IP addresses in its belong to the IP layer, not to the TCP layer. UDP uses the same pseudoheader for its checksum.

This is the reason due to which, IP, TCP and UDP all discard a packet that arrives with a checksum error and do not attempt to notify the source. If checksum field shows zero ('0') as a result, this means packet is received successfully. If there is an errror than checksum field does not show zero ('0') and that packet will be discarded automatically. After sometime, that data will be retransmitted.

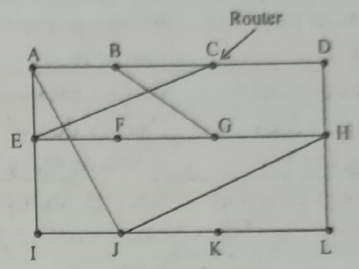
PART-C

Q.10 ***Explain distance vector routing protocol with suitable example.*** *[R.T.U. 2019]*

Ans. Distance Vector Routing : The distance vector routing algorithms operated by having each router maintain a table (*i.e.* a vector) giving the best known distance to each destination and which line to use to get there. These tables are updated by exchanging information with the neighbours.

The distance vector routing algorithm is sometimes called by other names, most commonly the distributed Bellman–Ford routing algorithm and the Ford–Fulkerson algorithm. It was the original ARPANET routing algorithm and was also used in the internet under the name RIP.

In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the subnet. This entry contains two parts : the preferred outgoing line to use for that destination and an estimate of the time or distance to that destination. The metric used might be number of hops, time delay in milliseconds, total number of packets queued along the path, or something similar.



(a) A subnet

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

delay is 8 | delay is 10 | delay is 12 | delay is 6 | New routing table for J

Vectors received from J's four neighbors

(b) Input from A, I, H K and the new routing table for J

Fig. : Distance Vector Routing Algorithm

The router is a assumed to know the 'distance' to each of its neighbors. If the metric is in hops, the distance is just one hop. If the metric is in queue length, the router simply examines each queue. If the metric is delay, the router can measure it directly with special ECHO packets that the receiver just time stamps and sends back as fast as it can.

As an example, assume that delay is used as a metric and that the router knows the delay to each of its neighbors. Once energy T μ/sec, each router sends to each neighbor a list of its estimated delays to each destination. It also receives a similar list from each neighbor. Imagine that one of these tables has just come from neighbor X, with X_i being X's estimate of how long it takes to get to router *i*.

If the router knows that the delay to X is m/sec., it also knows that it can reach i via X in X_i + m m/sec. By performing this calculation for each neighbor, a router can find out, which estimate seems the best and use that estimate best and the corresponding line in its new routing table. Note that the old routing table is not used in the calculation.

This updating process is illustrated in the above figure shows a subnet. The first four columns of part (b) shows the delay vectors received from the neighbors of router. A claims to have a 12 m/sec. delay to B, a 25 m/sec. delay to C, a 40 m/sec. delay to D etc. Suppose that J has measured or estimated its delay to its neighbors, A, I, H and K as 8, 10, 12 and 6 m/sec. respectively. Consider how J computes its new router to router G. It knows that it can get to A in 8 m/sec. and A claims to be able to get to G in 18 m/sec. so J knows it can count on a delay of 26 m/sec. to G if it forwards packets bound for G to A. Similarly, it computes the delay to G via I, H and K as 41(31 + 10), 18 (6 + 12), and 37 (31 + 6) m/sec. respectively. The best of these values is 18, so it makes an entry in its routing table that the delay to G is 18 m/sec. and that the route to use is via H. The same calculation is performed for all other destinations, with new routing table shown in the last column of the figure.

Advantages

- **Simpler :** Distance vector–based routing algorithms are easy to understand.
- Easy to configure

Disadvantages

- **Large routing tables :** Multiple routes to a given network ID can be reflected as multiple entries in the routing table. In a large internetwork with multiple paths, the routing table can have hundreds or thousands of entries.
- **High network traffic overhead :** Route advertising is done periodically even after the internetwork has converged.
- **Does not scale :** Between the size of the routing table and the high overhead, distance vector–based routing do not scale well to large and very large internetworks.
- **High convergence time :** Due to the unsynchronized and unacknowledged way by which distance vector information is exchanged, convergence of the internetwork can take several minutes. While converging, routing loops and black holes can occur.

Q.11 ***Discuss classes of IPV4. Also explain provision of multicast and broadcast support in IPV4.***

[R.T.U. 2016]

Ans. Classes of IPV4 : IP version 4 (IPv4) defines five address classes. Three of the classes, Classes A, B, and C, consist of unicast IP addresses. Unicast addresses identify a single host or interface, so that the address uniquely identifies the device. Class D addresses serve as multicast addresses, so that one packet sent to a Class D multicast IPv4 address may actually be delivered to multiple hosts. Finally, Class E addresses are experimental.

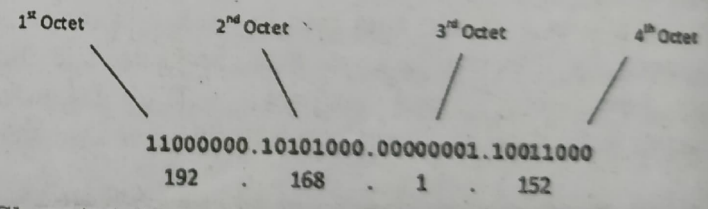
The classes can be identified on the basis of value of the first octet of the address as shown in the table –

Table 1: IPv4 Address Classes Based on First Octet Values

First Octet Values	Class	Purpose
1-127	A	Unicast (large networks)
128-191	B	Unicast (medium-sized networks)
192-223	C	Unicast (small networks)
224-239	D	Multicast
240-255	E	Experimental

CCENT and CCNA focus mostly on the unicast classes (A, B, and C) rather than Classes D and E. After you identify the class as either A, B, or C, many other related facts can be derived just through memorization.

The first octet referred here is the left most of all. The octets numbered as follc ws depicting dotted decimal notation of IP Address :



Class A Address

The first bit of the first octet is always set to 0 (zero). Thus the first octet ranges from 1 – 127, i.e.

00000001 – 01111111

1 - 127

Class A addresses only include IP starting from 1.x.x.x to 126.x.x.x only. The IP range 127.x.x.x is reserved for loopback IP addresses. The default subnet mask for Class A IP address is 255.0.0.0 which implies that Class A addressing can have 126 networks (2^7 -2) and 16777214 hosts (2^{24}-2).

Class A IP address format is thus:

0nnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh

Class B Address

An IP address which belongs to class B has the first two bits in the first octet set to 10, i.e.

10000000 – 10111111

128 - 191

Class B IP Addresses range from 128.0.x.x to 191.255.x.x. The default subnet mask for Class B is 255.255.x.x. Class B has 16384 (2^{14}) Network addresses and 65534 (2^{16}-2) Host addresses.

Class B IP address format is:

10nnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh

Class C Address

The first octet of Class C IP address has its first 3 bits set to 110, that is:

11000000 – 11011111

192 – 223

Class C IP addresses range from 192.0.0.x to 192.255.255.x. The default subnet mask for Class C is 255.255.255.x. Class C gives 2097152 (2^{21}) Network addresses and 254 (2^8 -2) Host addresses.

Class C IP address format is:

110nnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

Class D Address

Very first four bits of the first octet in Class D IP addresses are set to 1110, giving a range of:

11100000 – 11101111

224 – 239

Class D has IP address range from 224.0.0.0 to 239.255.255.255. Class D is reserved for multicasting. In multicasting data is not destined for a particular host, that is why there is no need to extract host address from the IP address, and Class D does not have any subnet mask.

Class E Address

This IP Class is reserved for experimental purposes only for R research purpose D or study. IP addresses in this class ranges from 240.0.0.0 to 255.255.255.254. Like Class D, this class too is not equipped with any subnet mask.

Provision of multicast and broadcast support in IPv4 –

The IPv4 version of the Internet Protocol (IP) discusses two types of point-to-multipoint communications:

Broadcast communications, in which a host sends information address to the entire network. IPv4 requires support for broadcast. The destination IP address is the subnet address with all the host ID bits set. For example, if the subnet is 192.168.240.xxx, then the broadcast address is 192.168.240.255—this is called directed broadcast. However, the official IP broadcast address has all bits set 255.255.255.255, which is what NDK uses for broadcast communications. Both these addresses are translated to the link layer address (MAC address) as FF:FF:FF:FF:FF:FF, and are received by every Ethernet adapter in the subnet.

Multicast communications are those in which a host sends information to a specific group of hosts. IPv4 makes support for multicast optional. Multicast communications are more complex—they require virtual groups of hosts to have an associated unique ID (or address). To differentiate groups from ordinary hosts, the IP standard assigns the address range from 224.0.0.0 to 239.255.255.255 (called class D addresses) to hold groups in the subnet. The lower 28 bits form the multicast group ID. The full 32-bit address is called the group address. Only 23 bits (of the full 28 bits) of the group ID are translated to the link layer, followed by a zero in the 24th bit. The remaining 24 bits are always 01:00:5E.

These two communication methods are enabled by assigning specific address numbers at both the protocol layer (IP) and the link layer (Ethernet). At the transport layer, both broadcast and multicast communications require UDP (User Datagram Protocol) or raw IP sockets to perform communications.

Q.12*(a) What do you understand by routing? Explain the classification of routing algorithms.*

(b) Discuss shortest path routing algorithm with help of suitable example. ***[R.T.U. 2016]***

Ans.(a) Routing algorithm as the network layer protocol that guides packets through the communication subnet to their correct destination. The times at which routing decisions are made depend on whether the network uses datagrams or virtual circuits. Routing is the act of moving information across an inter-network from a source to a destination. Along the way, at least one intermediate node typically is encountered. It is also referred to as the process of choosing a path over which to send the packets. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer.

Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of measurement; such as path bandwidth, reliability, delay, current load on that path etc; that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information.

Route information varies depending on the routing algorithm used. Routing algorithms fill routing tables with a variety of information. Mainly Destination/Next hop associations tell a router that a particular destination can be reached optimally by sending the packet to a particular node representing the "next hop" on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop. Some of the routing algorithm allows a router to have multiple "next hop" for a single destination depending upon best with regard to different metrics.

Routing Algorithm : Routing algorithms can be grouped into two major classes :Non–adaptive and adaptive. Non–adaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off–line, and down loaded to the router when the network is booted.

This procedure is called as Non–adaptive or fixed routing. In adaptive routing decisions that are made change as conditions on the network change. The principal conditions that influence routing decisions are :

- **Failure :** When a node or trunk fails, it can no longer be used as part of a route.
- **Congestion :** When a particular portion of the network is heavily congested, it is desirable to route packets around rather than through the area of congestion.

For adaptive routing to be possible, information about the state of the network must be exchanged among the nodes. There are several drawbacks associated with the use of adaptive routing, compared to non–adaptive routing :

(1) The routing decision is more complex, therefore the processing burden on network nodes increases.

(2) In most cases, adaptive strategies depend on status information that is collected at one place but used at another. There is a trade off here between the quality of the information and the amount of overhead. The more information that is exchanged, and the more frequently it is exchanged, the better will be the routing decisions that each node makes. On the other hand, this information is itself a load on the constituent networks, causing a performance degradation.

(3) An adaptive strategy may react too quickly, causing congestion producing oscillation, or too slowly, being irrelevant.

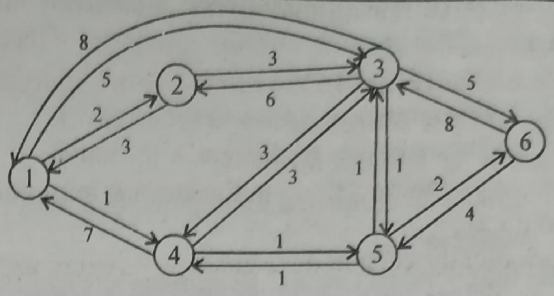
Despite of all demerits, adaptive routing are by for the most prevalent for two reasons.

(1) An adaptive routing strategy can improve performance, as seen by the network user.

(2) An adaptive routing strategy can add in congestion control because an adaptive routing strategy tends to balance loads, it can delay the onset of severe congestion.

Fixed Routing (Non–adaptive strategy)

For the fixed routing, a single, permanent route is configured for each source–destination pair of nodes in the network. Either of the least cost routing algorithm could be used. The routes are fixed, at least only change when there is a change in the topology of the network. Thus, the link costs used in designing routes cannot be based on any dynamic variable such as traffic.



Central Routing Directory from Node

	1	2	3	4	5	6
1	–	1	5	2	4	5
2	2	–	5	2	4	5
To Node 3	4	3	–	5	3	5
4	4	4	5	–	4	5
5	4	4	5	5	–	5
6	4	4	5	5	6	–

Node 1 Directory

Dest.	Next Node
2	2
3	4
4	4
5	4
6	4

Node 2 Directory

Dest.	Next Node
1	1
3	3
4	4
5	4
6	4

Node 3 Directory

Dest.	Next Node
1	5
2	5
4	5
5	5
6	5

Node 4 Directory

Dest.	Next Node
1	2
2	2
3	5
5	5
6	5

Node 5 Directory

Dest.	Next Node
1	4
2	4
3	3
4	4
6	6

Node 6 Directory

Dest.	Next Node
1	5
2	5
3	5
4	5
5	5

Fig. : Fixed Routing

The figure shows how fixed routing might be implemented. A central routing matrix is created, to be stored perhaps at a network control center. The matrix shows, for each source–destination pair of nodes, the identity of the next node on the route.

It is not necessary to store the complete route for each possible pair of nodes. Rather, it is sufficient to know, for each pair of nodes, the identity of the first node on the route. In above example, the route from node 1 to node 6 begins by going through node 4. Again consulting the matrix, the route from node 4 to node 6 goes through node 5. Finally, the route from node 5 to node 6 is a direct link to node 6. Thus, the complete route from node 1 to node 6 is 1–4–5–6.

Adaptive Routing Example

Bellman–Ford Algorithm : For this algorithm, each node maintain two vectors :

$$D_i = \begin{bmatrix} d_{i1} \\ \vdots \\ d_{iN} \end{bmatrix} \quad S_i = \begin{bmatrix} S_{i1} \\ \vdots \\ S_{iN} \end{bmatrix}$$

where D_i = delay vector for node i

d_{ij} = current estimate of minimum delay from node i to node j

($d_{ij} = 0$)

N = number of nodes in the network

S_i = successor node vector for node i

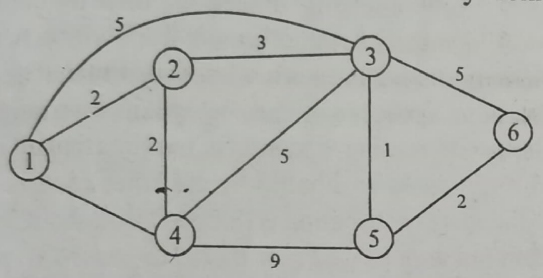
s_{ij} = the next node in the current minimum delay route from i to j.

Periodically, each node exchanges its delay vector with all of its neighbors. On the basis of all incoming delay vectors, a node updates both of its vectors as follows : $d_{kj} = \min_{i \in A}[d_{ij} + l_{ki}]$

where, s_{kj} = i using i that minimizes the preceding expression

A = set of neighbor nodes of k

l_{ki} = current estimate of delay from k to i.



(a) Network Example

Dest.	Delay	Next Node	D_1	S_1	D_2	D_3	D_4	Dest.	Delay	Next Node
1	0	–	3	7	5	1	0	–		
2	2	2	0	4	2	2	2	2		
3	3	3	3	0	2	3	3	4		
4	1	4	2	2	0	4	1	4		
5	6	3	3	1	1	5	2	4		
6	8	3	5	3	3	6	4	4		

$I_{1,2} = 2$

$I_{1,3} = 5$

$I_{1,4} = 1$

(b) Node 1's routing table before update (c) Delay vectors sent from node 1 neighbor nodes in update (d) Node 1's routing table after update and link costs used to update

Fig.

The above figure, the network with some of the links costs having different values (and assuming the same cost in both directions).

The routing table for node 1 at an instant in time that reflects the link costs of the network (above case). For each destination, a delay is specified, and the next node on the route that produces that delay. Assume that node 1's neighbors (node 2, 3 and 4) learn of the change before node 1. Each of these nodes updates its delay vector and sends a copy to all of its neighbors, including node 1. Node 1 discards its current routing table and builds a new one, based solely on the incoming delay vector and is own estimate of link delay to each of its neighbors. The result is shown in fig.

Ans.(b) Dijkstra Algorithm : The **link-state routing** algorithm given here is known as Dijkstra's algorithm. Dijkstra algorithm can be described as – Find the shortest paths from a given source node to all other nodes, by developing the paths in order of increasing path length. The algorithm proceeds in stages. By path stage, the shortest paths to the p nodes closest to (least cost away from) the source node have been determined, these nodes are in a set S. At stage (p + 1), the node not in S that has the shortest path from the source node is added to S. As each node is added to S, its path from the source is defined. The algorithm can be formally described as follows. Define :

N = set of nodes in the network

x = source node

S = set of nodes so far incorporated by the algorithm

w(i, j) = link cost from node i to node j, $w(i, j) = 0$,

$w(i, j) = \infty$ = if the two nodes are not directly connected, $w(i, j) \geq 0$

y = the two nodes are directly connected

L(n) = cost of the least cost path from node x to node n that is currently known to the algorithm, at termination, this is the cost of the least cost path in the graph from x to n.

The algorithm has three, steps and step 2 and 3 are repeated until S = N, *i.e.* step 2 and 3 are repeated until final paths have been assigned to all nodes in the network.

Step 1 : Initialization

S = {x} *i.e.* the set of nodes so far incorporated consists of only the source node.

L(n) = w(x, n), for $n \neq x$ *i.e.* the initial path costs to neighboring nodes are simply the link costs.

Step 2 : Get Next Node

Find the neighboring node not in S that has the least cost path from node x and incorporate that node in S : Also incorporate the edge that is incident on that node and a node in S that contributes to the path. This can be expressed as :

Find $y \notin S$ such that $L(y) = \min_{j \notin T} L(j)$

Add y to S, add to S the edge that is incident on y and that contributes the least cost component to L(y), *i.e.* the last hop in the path.

Step 3 : Update Least Cost Path

$L(n) = \min[L(n), L(y) + w(y, n)]$, for all $n \notin S$

If the later term is the minimum, the path from x to n is now the path from x to y concatenated with the edge from y to n.

The algorithm terminates when all nodes have been added to S. At termination, the value L(y) associated with each node y is the cost (length) of the least cost path from x to y. In addition, S defines the least cost path from S to each other node.

One iteration of step 2 and 3 adds one new node to S and defines the least cost path from x to that node. That path passes only through nodes that are in S. To see this, consider the following line of reasoning. After P iterations, there are p nodes in S, and the least cost path from x to each of the there nodes has been defined. Now consider all possible paths from

x to nodes not in S. Among those paths, there is one of least cost that passes exclusively through nodes in S, ending with a direct link from some node in S to a node not in S. This node is added to S and the associated path is defined as the least cost path for that node.

Example :

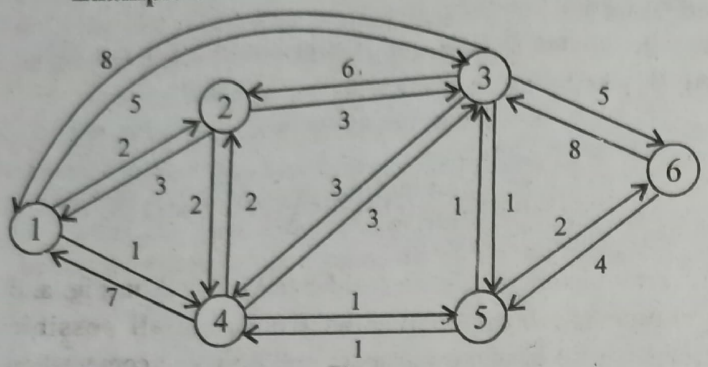
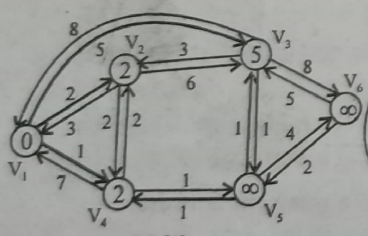


Fig.

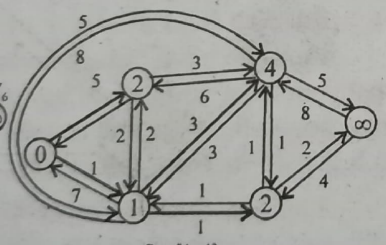
For (x = 1)

Iteration	S	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	[1]	2	1–2	5	1–3	1	1–4	∞	–	∞	–
2	[1,4]	2	1–2	4	1–4–3	1	1–4	2	1–4–5	∞	–
3	[1,2,4]	2	1–2	4	1–4–3	1	1–4	2	1–4–5	∞	–
4	[1,2,4,5]	2	1–2	3	1–4–5–3	1	1–4	2	1–4–5	4	1–4–5–6
5	[1,2,3,4,5]	2	1–2	3	1–4–5–3	1	1–4	2	1–4–5	4	1–4–5–6
6	[1,2,3,4,5,6]	2	1–2	3	1–4–5–3	1	1–4	2	1–4–5	4	1–4–5–6



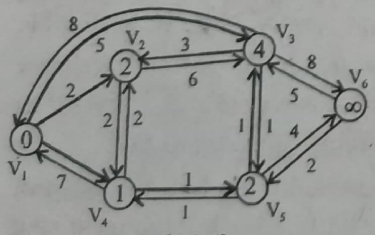
S = [1]

(a)



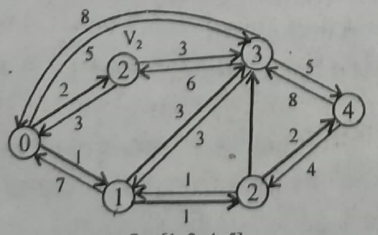
S = [1, 4]

(b)



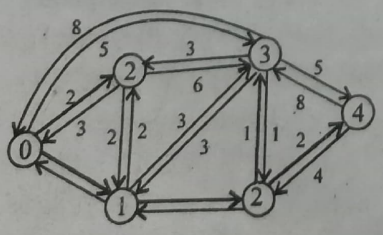
S = [1, 2, 4]

(c)



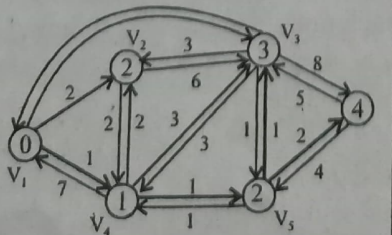
S = [1, 2, 4, 5]

(d)



S = [1, 2, 3, 4, 5]

(e)



S = [1, 2, 3, 4, 5, 6]

(f)

Fig.

The dark edges define the spanning tree for the graph. The value in each circle are the current estimates of L(y) for each node y. A node is shaded when it is added to S. It is to be noticed that at each step the path to each node plus the total cost of that path is generated. After the final iteration, the least cost path to each node and the cost of the path have been developed. The same procedure can be used with node 2 as source node, and so on.

Q.13 ***Contrast between distance vector and link state routing after discussing both.*** *[R.T.U. 2016]*

Ans. Contrast between distance vector routing and link-state routing: In distance vector routing the routing information is only exchanged between directly connected neighbors. This means a router knows from which neighbor a route was learned, but it does not know where that neighbor learned the route; a router can't see beyond its own neighbors. While in link-state routing, in contrast, requires that all routers know about the paths reachable by all other routers in the network. Link-state information is flooded throughout the link-state domain (an area in OSPF or IS-IS) to ensure all routers posses a synchronized copy of the area's link-state database.

Distance Vector exchanges the routing updates periodically whether the topology is change or not, this will maximize the convergence time which increases the chance of routing loops while the Link State routing protocols send triggered change based updates when there is a topology change. After initial flood, pass small event based triggered link state updates to all other routers.

Distance Vector routing is less scalable such as RIP supports 16 hops and IGRP has a maximum of 100 hops. Link State routing is very much scalable and supports infinite hops.

Distance vector routing : *Refer to Q.10.*

Flow Based Routing

- This is a static algorithm which uses topology and load condition (traffic) for deciding a route.
- For example in fig. there is always a huge from A to B. Then the traffic from A to C should not be routed through B.

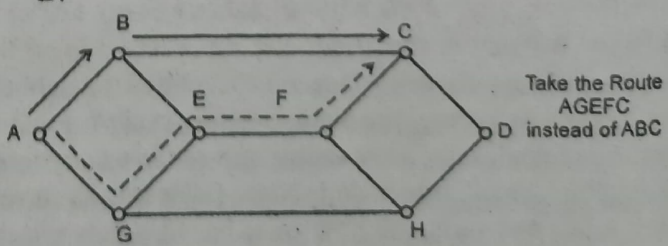


Fig. : Flow Based Routing

- Instead route it through AGEFC even through it is a longer path than ABC. This is called as a flow based routing.

- It is possible to optimize the routing by analyzing the data flow mathematically. This is possible if the average traffic from one node to the other is known in advance and it is constant in time.
- The Mathematical analysis is based on idea that for a given line if the capacity and average data flow are known, then it is possible to calculate the mean packet delay using the Queueing Theory.
- From the mean delays on all the lines it is possible to calculate the mean packet delay for the whole subnet.
- To use the technique of flow based routing, the following information should be known in advance.

(1) Subnet topology

(2) Traffic matrix

(3) Line capacity matrix which specifies capacity of each line.

Link state routing : The link state algorithm presented below is known as Dijkstra's algorithm named after its inventor. A closely related algorithm if Prim's algorithm, for a general discussion of graph algorithms. Dijkstra's algorithm computes the least–cost path from one node to all other nodes in the network. Dijkstra's algorithm is iterative and has the property that after the k^{th} iteration of algorithm, the least–cost paths are known to destination nodes, and among the least–cost paths to all destination nodes, these k paths will have the k smallest costs. We define the following notation :

1. c(i, j) : Link cost from node i to node j. If nodes i and j are not directly connected then $c(i, j) = \infty$. We will assume that c(i, j) equals c(j, i) in order to simplify off notation and examples but stress that the algorithm works for the case that c(i, j) and c(j, i) are not equal.

2. D(v) : Cost of path from the source node to destination v that has currently (as of this iteration of the algorithm) the least cost.

3. P(v) : Previous node (neighbor of v) along the current least–cost path from the source to v.

4. N : Subset of nodes whose least–cost path from the source to v is definitely known.

The link state algorithm consists of an initialization step followed by a loop. The number of times the loop is executed is equal to the number of nodes in the network. Upon termination, the algorithm will have calculated the shortest paths from the source node (A) to every other node in the network. Link State (LS) Algorithm for Source Node A :

1. Initialization :
2. N = (A)
3. for all nodes v
4. if v adjacent to A
5. then D(B) = c (A, B)
6. else D(B) = ∞
7. Loop
8. Find C not in N such that D(C) is a minimum
9. add C to N
10. update D(B) for all B adjacent to C and not in N :
11. D(B) = min (D(B), D(C) + c(C, B))
    u → A, v → B, w → c x → D, y → E, z → F
12. /* new cost to B is either old cost to B or known shortest path cost to w plus cost from C to B */
13. Until all nodes in N

As an example, let's consider the network in Fig. and compute the least cost paths from A to all possible destinations. A tabular summary of the algorithm's computation is shown in table , where each line in the table gives the values of the algorithm's variables at the end of the iteration. Let's consider the few first steps in detail.

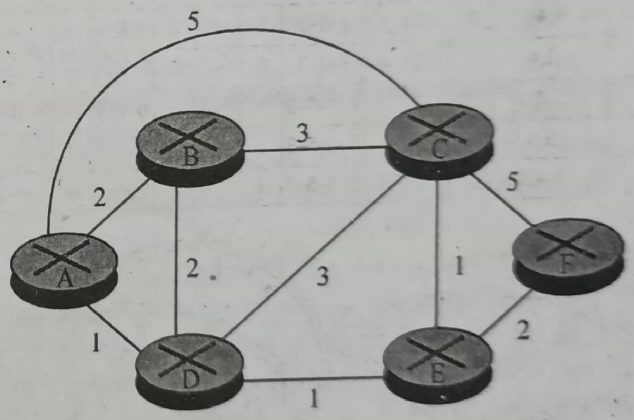


Fig. : Abstract model of a computer network

- In the initialization step, the currently known least–cost paths from A to its directly attached neighbors, B, C and D, are initialized to 2,5 and 1, respectively. Note in particular that the cost to C is set to 5 since this is the cost of the direct (one hop) link from A to C. The costs to E and F are set to infinity because they are not directly connected to A.
- In the first iteration, we look among those nodes not yet added to the set N and find that node with the least cost as of the end of the previous iteration. That node is D, with a cost of 1, and thus D is added to the set N. Line 12 of the LS algorithm is then performed to update D(B) for all nodes B, yielding the results shown in the second line (step 1) in Table. The cost of the path to B is unchanged. The cost of the path to C (which was 5 at the end of the initialization) through node D is found to have a cost of 4. Hence, this lower cost path is selected as C's predecessor along the shortest path from A is set to D. Similarly, the cost to E (through D) is computed to be 2, and the table is updated accordingly.

Table : Running the link state algorithm on the network in fig.

Step	N	D(B), P(B)	D(C), P(C)	D(D), P(D)	D(E), P(E)	D(F) F(F)
0	A	2,A	5,A	1,A	∞	∞
1	AD	2,A	4,D		2,D	∞
2	ADE	2,A	3,E			4,E
3	ADEB		3E			4,E
4	ADEBC					4,E
5	ADEBCF					

- In the second iteration, nodes B and E are found to have the least–cost paths (2) and we break the tie arbitrarily and add E to the set N so that N now contains A, D and E. The cost to the remaining nodes not yet in N, that is, nodes B, C, and F are updated via line 12 of the LS algorithm, yielding the results shown in the third row in the above table.
- And so on.

Q.14 ***State the concept of tunneling. Under what practical circumstances it is used? Explain by suitable example.***

[R.T.U. 2016]

OR

What is tunneling and fragmentation? Explain it.

[R.T.U. 2015, 2010]

Ans. Tunneling : Handling the general case of making twc different networks interwork is exceedingly difficult. However, there is a common special case that is manageable. This case is where the source and destination hosts are on the same type of network, but there is a different network in between. As an example, we think of an international bank with a TCP/IP-based Ethernet in Paris, a TCP/IP-based Ethernet in London, and a non-IP wide area network (e.g. ATM) in between, as shown in Fig.1.

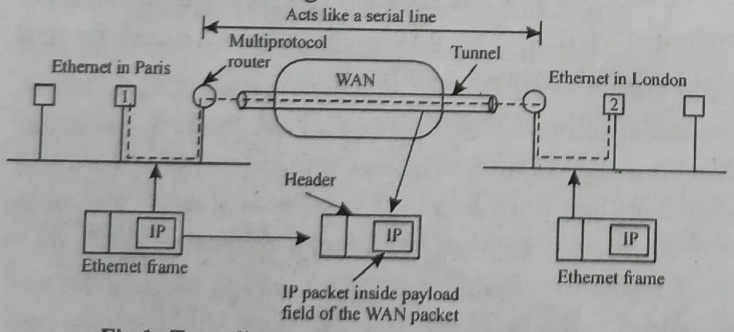


Fig.1: Tunneling a Packet from Paris to London

The solution to this problem is a technique called **tunneling**. To send an IP packet to host 2, host 1 constructs the packet containing the IP address of host 2, inserts it into an Ethernet frame addressed to the Paris multiprotocol router, and puts it on the Ethernet. When the multiprotocol router gets the frame, it removes the IP packet, inserts it in the payload field of the WAN network layer packet, and addresses the latter to the WAN address of the London multiprotocol router. When it gets there, the London router removes the IP packet and sends it to host 2 inside an Ethernet frame.

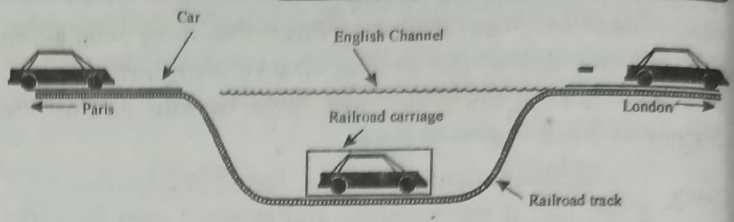


Fig.2: Tunneling a Car from France to England

The WAN can be seen as a big tunnel extending from one multiprotocol router to the other. The IP packet just travels from one end of the tunnel to the other, snug in its nice box. It does not have to worry about dealing with the WAN at all. Neither do the hosts on either Ethernet. Only the multiprotocol router has to understand IP and WAN packets. In effect, the entire distance from the middle of one multiprotocol router to the middle of the other acts like a serial line.

An analogy may make tunneling clearer. Consider a person driving her car from Paris to London. Within France, the car moves under its own power, but when it hits the English Channel, it is loaded into a high-speed train and transported to England through the Channel (cars are not permitted to drive through the Channel). Effectively, the car is being carried as freight, as depicted in Fig.2. At the far end, the car is let loose on the English roads and once again continues to move under its own power. Tunneling of packets through a foreign network works the same way.

Fragmentation : Each network imposes some maximum size on its packets. These limits have various causes, among them :

1. Hardware (e.g., the size of an Ethernet frame)
2. Operating system (e.g., all buffers are 512 bytes)
3. Protocols (e.g., the number of bits in the packet length field).
4. Compliance with some (inter) national standard.
5. Desire to reduce error-induced retransmissions to some level.
6. Desire to prevent one packet from occupying the channel too long.

The result of all these factors is that the network designers are not free to choose any maximum packet size they wish. Maximum payloads range from 48 bytes (ATM) cells to 65,515 bytes (IP packets), although the payload size in higher layers is often larger.

An obvious problem appears when a large packet wants to travel through a network whose maximum packet size is too small. If the original source packet is too large to be handled by the destination network, the routing algorithm can hardly by pass the destination. Basically, the only solution to the problem is to allow gateways to break up packets into fragments, sending each fragment as a separate internet packet. However, as every parent of a small child knows, converting a large object into small fragments is considerably

easier than the reverse process. (Physicists have even given this effect a name: *the second law of thermodynamics*). Packet-switching networks, too, have trouble putting the fragments back together again.

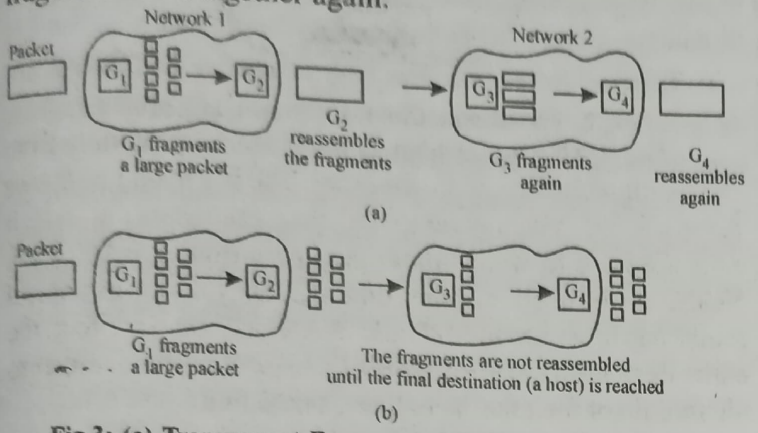


Fig.3: (a) Transparent Fragmentation (b) Non-transparent Fragmentation

Two opposing strategies exist for recombining the fragments back into the original packet. The first strategy is to make fragmentation caused by a "small-packet" network transparent to any subsequent networks through which the packet must pass on its way to the ultimate destination. This option is shown in Fig.3(a). In this approach, the small-packet network has gateways (most likely, specialized routers) that interface to other networks. When an over sized packet arrives at a gateway, the gateway breaks it up into fragments. Each fragement is addressed to the same exit gateway, where the pieces are recombined. In this way passage through the small-packet network has been made transparent. Subsequent networks are not even aware that fragmentation has occurred. ATM networks, for example, have special hardware to provide transparent fragmentation of packets into cells and then reassembly of cells into packets. In the ATM world, fragmentation is called **segmentation**, the concept is the same, but some of the details are different.

Transparent fragmentation is straight forward but has some problems. For one thing, the exit gateway must know when it has received all the pieces, so either a count field or an "end of packet" bit must be provided. For another thing, all packets must exit via the same gateway. By not allowing some fragments to follow one route to the ultimate destination and other fragments a disjoint route, some performance may be lost. A last problem is the overhead required to repeatedly reassemble and then refragment a large packet passing through a series of small-packet networks. ATM requires transparent fragmentation.

The other fragmentation strategy is to refrain from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet.All fragments are passed through the exit gateway (or gateways), as shown in Fig.3(b). Recombination occurs only at the destination host. IP works this way.

Non-transparent fragmentation also has some problems. For example, it requires every host to be able to do reassembly. Yet another problem is that when a large packet is fragmented, the total overhead increased because each fragment must have a header. Whereas in the first method this overhead disappears as soon as the small-packet network is exited, in this method the overhead remains for the rest of the journey. An advantage of non-transparent fragmentation, however, is that multiple exit gateways can now be used and higher performance can be achieved. Of course, if the concatenated virtual circuit model is being used, this advantage is of no use.

When a packet is fragmented, the fragments must be numbered in such a way that the original data stream can be reconstructed. One way of numbering the fragments is to use a tree. If packet 0 must be split up, the pieces are called 0.0, 0.1, 0.2, etc. If these fragments themselves must be fragmented later on, the pieces are numbered 0.0.0, 0.0.1, 0.0.2, ..., 0.1.0, 0.1.1, 0.1.2, etc. If enough fields have been reserved in the header for the worst case and no duplicates are generated anywhere, this scheme is sufficient to ensure that all the pieces can be correctly reassembled at the destination, no matter what order they arrive in.

However, if even one network loses or discards packets, end-to-end retransmissions are needed, with unfortunate effects for the numbering system. Suppose that a 1024-bit packet is initially fragmented into four equal-sized fragments, 0.0, 0.1, 0.2, and 0.3. Fragment 0.1 is lost, but the other parts arrive at the destination. Eventually, the source times out and retransmits the original packet again. Only this time Murphy's law strikes and the route taken passes through a network with a 512-bit limit, so two fragments are generated. When the new fragment 0.1 arrives at the destination, the receiver will think that all four pieces are now accounted for and reconstruct the packet incorrectly.

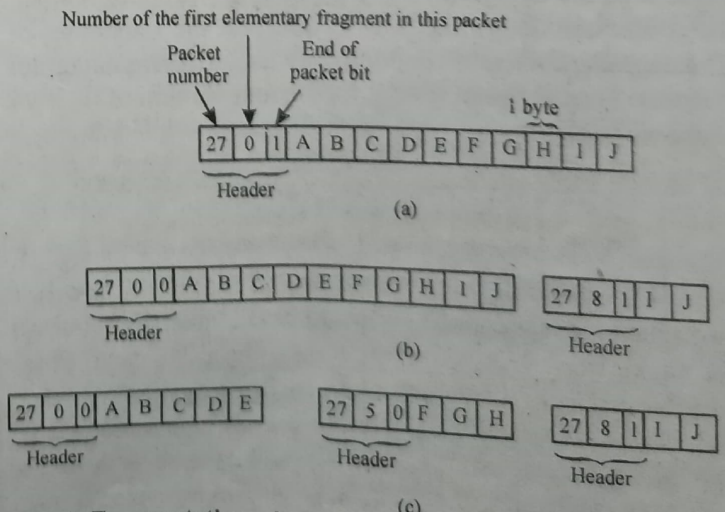


Fig.4: Fragmentation when the elementary data size is 1 byte. (a) Original packet, containing 10 data bytes. (b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header. (c) Fragments after passing through a size 5 gateway.

A completely different (and better) numbering system is for the internetwork protocol to define an elementary fragment size small enough that the elementary fragment can pass through every network. When a packet is fragmented, all the pieces are equal to the elementary fragment size except the last one, which may be shorter. An internet packet may contain several fragments, for efficiency reasons. The internet header must provide the original packet number and the number of the (first) elementary fragment contained in the packet. As usual, there must also be a bit indicating that the last elementary fragment contained within the internet packet is the last one of the original packet.

This approach requires two sequence fields in the internet header, the original packet number and the fragment number. There is clearly a trade-off between the size of the elementary fragment and the number of bits in the fragment number. Because the elementary fragment size is presumed to be acceptable to every network, subsequent fragmentation of an internet packet containing several fragments causes no problem. The ultimate limit here is to have the elementary fragment be a single bit or byte, with the fragment number then being the bit or byte offset within the original packet, as shown in Fig.4.

Q.15 ***Write short note on IPv4 and IPv6 packet format.*** *[R.T.U. 2015, 2010, Raj. Univ. 2007, 2004]*

OR

Draw header details of IPv4 protocol and describe each one of them briefly. *[Raj. Univ. 2005]*

OR

Write short note IPv6 Packet Format. *[Raj. Univ. 2004]*

Ans. IPv4 : The IPv4 datagram format is shown in Fig. The key fields in the IPv4 datagram are given below :

Version Number : These 4 bits specify the IP protocol version of the datagarm. By looking at the version number, the router can determine how to interpret the remainder of IP datagram. Different versions of IP use different datagram formats. The datagram format for current version of IP, IPv4 is shown in figure.

32-bits

<table>
<tr><td>Version</td><td>Header Length</td><td>Type of Service</td><td colspan="2">Datagram length (bytes)</td></tr>
<tr><td colspan="3">16 - bit identifier</td><td>Flags</td><td>13 - bit Fragmentation offset</td></tr>
<tr><td>Time to live</td><td colspan="2">Upper layer protocol</td><td colspan="2">Header checksum</td></tr>
<tr><td colspan="5">32- bit Source IP Address</td></tr>
<tr><td colspan="5">32 - bit Destination IP Address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

Fig. : IPv4 datagram format

Header Length : Because an IPv4 datagram can contain a variable number of option these 4 bits are needed to determine where in the IP datagram the data actually begins. Most IP datagrams do not contain options so the typical IP datagram has a 20 byte header.

Type of Service : The type of service (TOS) bits are included in IPv4 header to allow different types of IP datagram to be distinguish from each other.

Datagram Length : This is the total length of IP datagram, measured in bytes. Since this filed is 16-bit long, the theoretical maximum size of IP datagram is 65, 535 bytes.

Identifier, Flags Fragmentation Offset : These three fields have to do with so called IP fragmentation. The new version of IP, IPv6 doesn't allow fragmentation at routers.

Time to Live (TTL) : This field is included to ensure that datagrams do not circulate forever in network. This field is decremented by one each item the datagram is processed by a router. If TTL field reaches 0, the datagram must be dropped.

Protocol : This field is used only when an IP datagram reaches its final destination. The value for this field indicates the specific transport layer protocol to which the data portion of this IP datagram should be passed.

Header Checksum : The header checksum aids a router in detecting but errors in a received IP datagram. The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using 1's complement arithmetic.

Source and Destination IP Addresses : When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field.

Options : The option field allows an IP header to be extended. Header options were meant to be used rarely hence the decision to save overhead by not including the information in options fields in every datagram header.

Data (Payload) : The data field of IP datagram contains the transport layer segment (TCP or UDP) to be delivered to destination. However , the data field can carry other types of data, such as ICMP messages.

IPv6 : Internet protocol is the host–to–host network layer delivery protocol for the Internet. IP is an unreliable and connectionless datagram protocol. IP provides no error control or flow control. IP uses only an error detection mechanism and discard the packet if it is corrupted.

If reliability is important, IP must be paired with a reliable protocol such as TCP.

IPv6 is an updated version of the internet protocol based on IPv4. IPv4 and IPv6 are demultiplexed at the media layer. The most important changes introduced in IPv6 are evident in the datagram format :

Expanded Addressing Capabilities : IPv6 increases the size of the IP address from 32 to 128 bits. In addition to unicast and multicast address, IPv6 has introduced a new type of address, called as any cast address, which allows a datagram addressed to an any cast address to be desired to any one of a group of hosts.

A stream Lined 40–byte Header : A number of IPv4 fields have been dropped or made optimal. The resulting 40–byte fixed–length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.

Flow Labeling and Priority : IPv6 has an exclusive definition of a "flow".

This allows "Labeling of packets belonging to particular flows for which the sender requests special handling, such as a non default quality of service or real–time service".

The IPv6 header also has an 8–bit traffic classfield. This field can be used to give priority to certain packets within a flow.

<table>
<tr><td>Version(4 bit)</td><td>Traffic class (4 bit)</td><td colspan="2">Flow Label (24 bit)</td></tr>
<tr><td colspan="2">Payload Length (16 bit)</td><td>Next Header (8 bit)</td><td>Hop Limit (8 bit)</td></tr>
<tr><td colspan="4">Source Address (128 bits)</td></tr>
<tr><td colspan="4">Destination Address (128 bits)</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

Fig. : IPv6 Datagram Format

The following fields are defined in IPv6 :

Version : This four–bit field identifies the IP version number. IPv6 carries a value of "6" in this field.

Traffic Class : Enables a source to identify the desired delivery priority of the packets. Priority values are divided into ranges: traffic where the source provides congestion control and non–congestion control traffic.

Flow Label : Used by a source to label those products for which it requests special handling by the IPv6 router. The flow is uniquely identified by the combination of a source address and non-zero flow label.

Payload Length : This 16–bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed–length, 40–byte datagram header.

Next Header : Identifies the type of header immediately following the IPv6 header.

Hop Limit : 8–bit integer that is decremented by 1, by each node that forwards the packet. The packet is discarded if the hop limit is decremented to zero.

Source Address : 128–bit address of the originator of the packet.

Destination Address : 128–bit address of the intended recipient of the packet.

Data : This is the payload portion of the IPv6 datagram. When the datagram reaches its destination, the payload will be removed from the IP datagram and passes on to the protocol specified in the next header field.

The discussion above identified the purpose of the fields that are included in the IPv6 datagram. The several fields appearing in the IPV4 datagram are no longer present in IPv6 datagram :

Fragmentation/Reassembly : IPv6 does not allow for fragmentation and reassembly at intermediate routers, these operations can be performed only by the source and destination. If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a "Packet Too Big" ICMP error message back to the sender.

The sender can then resend the data, using a smaller IP datagram size. Fragmentation and reassembly is a time-consuming operation, removing this functionality from the routers and placing it squarely in the end systems considerably speeds up IP forwarding within the network.

Header Checksum : Because the transport layer (for example, TCP and UDP) and data link (for example, Ethernet) protocols in the Internet layers perform checksumming, the designers of IP probably felt that this functionality was sufficiently redundant in the network layer that it could be removed. Once again, fast processing of IP packets was a central concern. The IPv4 header contains a TTL field (similar to the hop limit field in IPv6), the IPv4 header checksum needed to be recomputed at every router. As with fragmentation and reassembly, this too was a costly operation in IPv4.

Options : An options field is no longer a part of the standard IP header. However, it has not gone away. Instead, the options field is one of the possible "next headers" pointed to from within the IPv6 header. That is, just as TCP or UDP protocol headers can be the next header within an IP packet, so too can an options field. The removal of the options field results in a fixed–length, 40–byte IP header.

Q.16 ***(a) Describe Routing principle. Explain distance vector routing algorithm.***

[R.T.U. 2013, Raj. Univ. 2006, 2005, 2004]

(b) Explain flooding and shortest path routing and describe how they are used in link state routing.

[R.T.U. 2013]

OR

Using suitable example, explain the following routing strategy :

(i) Flooding

(ii) Dijkstra's Algorithm ***[R.T.U. 2009]***

OR

Describe the Dijkstra algorithm used in packet switching network. ***[Raj. Univ. 2007]***

OR

Describe Dijkstra's congestion control routing algorithm with suitable example. ***[Raj. Univ. 2006, 2005]***

Ans.(a) Routing Principle : A routing protocol is used by routers to dynamically find all the networks in the internetwork and to ensure that all routers have the same routing table. Basically, routing protocols determine the path of a packet through an internetwork.

Example : RIP, IGRP, EIGRP and OSPF.

Once all routers know about all networks, a routed protocol can be used to send user data (Packets) through the established enterprise. Routed protocols are assigned to an interface and determine the method of packet delivery. **Example :** IP and IPX. There are three classes of routing protocols :

Distance Vector Routing : *Refer to Q.10.*

Ans.(b) (i) Flooding : It is a static algorithm. In this routing scheme, at each node an incoming packet is retransmitted on all outgoing links except for the link on which it arrived. Eventually a number of copies of the desired packets will reach the destination but the destination picks the first correct one and discards the other based on their identification. It is obvious among all tried nodes one shortest path will also exist and one copy of the packet will definitely reach the destination.

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

An alternative technique for damming the flood is to keep track of which packets have been flooded, to avoid sending them out a second time.

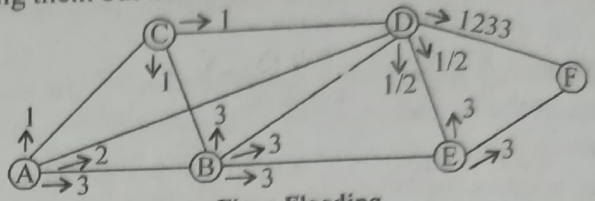


Fig. : Flooding

Achieve this goal is to have the source router put a sequence number in each packet it receives from its hosts. Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.

To prevent the list from growing without bound, each list should be augmented by a counter, *k*, meaning that all sequence numbers through *k* have been seen. When a packet comes in, it is easy to check that packet is duplicate, if so, it is discarded. Furthermore, the full list below *k* is not needed, since *k* effectively summarizes it.

A variation flooding that is slightly more practical is selective flooding. In this algorithm the routers do not send every incoming packet out on every line, only on those lines that are going approximately in the right direction. There is usually little point in sending a westbound packet on an eastbound line unless the topology is extremely peculiar and the router is sure of this fact.

Dijkstra's algorithm to compute the shortest path through a graph.

```
#define MAX_NODES 1024    /*maximum number of
                            nodes */
#define INFINITY 1000000000
        /*a number larger than every maximum path */
int n, dist[MAX_NODES][MAX–NODES]

/* dist[MAX_NODES][MAX–NODES] is the distance
                                  from i to j */
void shortest_path(int s, int t, int path[])

{
    struct state
        {           /* the path being worked on */
        int predecessor;
                    /*previous node */
        enum (permanent, tentative) label;
                    /* label state */
        }
    state [MAX–NODES];
    int i, k, min;
    struct state *p;
```

```
for (p = &state [0]; p < & state[n]; p++)
                    /* initialize state */
{
    p->predecessor = –1;
    p->length = INFINITY;
    p->label = tentative;
}
state [t].length = 0; state [t].label = permanent;
k = t;                  /* k is the initial working node */
do
{       /* is there a better path from k ? */
for (i = 0; i < n; i ++)  /* this graph has n nodes */
    if(dist[k][i] ! = 0 && state [i].label = tentative)
    {
    if (state [k].length + dist[k][i] < state [i].length)
    {
    state [i].predecessor = k;
    state [i].length = state [k].length + dist[k][i];
        }
    }
/* Find the tentatively labeled node with the smallest
label. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
    if (state [i].label = tentative && state [i].length
    <min)
    {
        min = state [i].length;
        k = i;
    }
    state [k].label = permanent;
}
while (k! = s);  /* Copy the path into the output array. */
i = 0; k = s;
do
    {
        path [i++] = k; k = state [k].predecessor;
    }
while (k>=0);
}
```

Applications

Flooding is not practical in most applications, but it does have some uses.

- In military applications, where large numbers of routers may be blown to bits at any instant, the tremendous robustness of flooding is highly desirable.
- In distributed database applications, it is sometimes necessary to update all the databases concurrently, in which case flooding can be useful.
- In wireless networks, all messages transmitted by a station can be received by all other stations within its radio range, which is, in fact, flooding, and some algorithms utilize this property.
- A fourth possible use of flooding is a metric against which other routing algorithms can be compared.
- Flooding always chooses the shortest path because it chooses every possible path in parallel. Consequently, no other algorithm can produce a shorter delay (if we ignore the overhead generated by the flooding process itself).

(ii) Dijkstra Algorithm : *Refer to Q.12(b).*

Q.17 ***(a) What is the difference between open-loop congestion control and closed loop congestion control? (b) What is four general techniques to improve the quality of service.*** *[R.T.U. 2013]*

Ans.(a) Open-Loop Congestion Control

In open-loop congestion control, policies are applied to prevent congestion before it happens. In these mechanisms, congestion control is handled by either the source or the destination. A brief list of policies that can prevent congestion is given below :

(i) Retransmission Policy

Retransmission is sometimes unavoidable. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. Retransmission in general may increase congestion in the network. However, a good retransmission policy can prevent congestion. The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion. For example, the retransmission policy used by TCP is designed to prevent or alleviate congestion.

(ii) Window Policy

The type of window at the sender may also affect congestion. The Selective Repeat window is better than the Go-Back-N window for congestion control. In the Go-Back-N window, when the timer for a packet times out, several packets may be re-sent, although some may have arrived safe and sound at the receiver. This duplication may make the congestion worse. The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

(iii) Acknowledgment Policy

The acknowledgment policy imposed by the receiver may also affect congestion. If the receiver does not acknowledge every packet it receives, it may slow down the sender and help to prevent congestion. Several approaches are used in this case. A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires. A

receiver may decide to acknowledge only N packets at a time. We need to know that the acknowledgments are also part of the load in a network. Sending fewer acknowledgments means imposing fewer loads on the network.

(iv) Discarding Policy

A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission. For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

(v) Admission Policy

An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual-circuit networks. 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 or if there is a possibility of future congestion.

Closed-Loop Congestion Control

Closed-loop congestion control mechanisms try to alleviate congestion after it happens. Several mechanisms have been used by different protocols. Few of them are described here.

(i) Backpressure

The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes. This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream nodes or nodes and so on. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source.

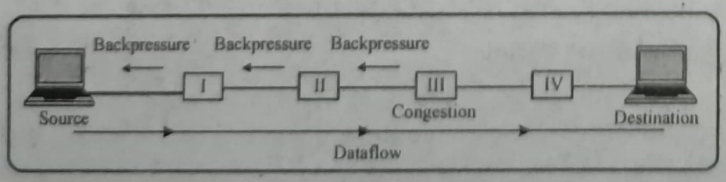


Fig. 1 : Backpressure

The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is corning. Figure 1 shows the idea of backpressure.

Node III in the figure has more input data than it can handle. It drops some packets in its input buffer and informs node II to slow down. Node II, in turn, may be congested because it is slowing down the output flow of data. If node II is congested, it informs node I to slow down, which in turn may create congestion. If so, node I inform the source of data to slow down. This, in time, alleviates the congestion. Note that the pressure on node III is moved backward to the source to remove the congestion.

(ii) Choke Packet

A choke packet is a packet sent by a node to the source to inform it of congestion. In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station. In the choke packet method, the warning is from the router, which has encountered congestion, to the source station directly. The intermediate nodes through which the packet has traveled are not warned. When a router in the Internet is overwhelmed with IP datagrams, it may discard some of them; but it informs the source host, using a source quench ICMP message. The warning message goes directly to the source station; the intermediate routers, and does not take any action. Figure 2 shows the idea of a choke packet.

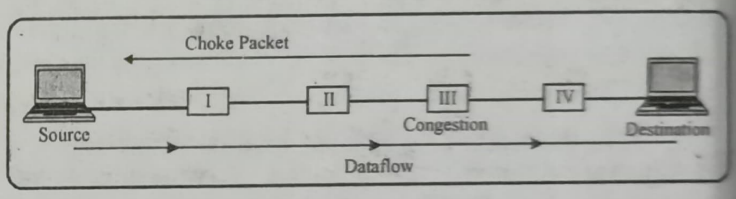


Fig. 2 : Choke Packet

(iii) Implicit Signaling

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

(iv) Explicit Signaling

The node that experiences congestion can explicitly send a signal to the source or destination. The explicit signaling method, however, is different from the choke packet method. In the choke packet method, a separate packet is used for this purpose; in the explicit signaling method, the signal is included in the packets that carry data. Explicit signaling can occur in either the forward or the backward direction.

- **Backward Signaling :** A bit can be set in a packet moving in the direction opposite to the congestion. This bit can warn the source that there is congestion and that it needs to slow down to avoid the discarding of packets.
- **Forward Signaling :** A bit can be set in a packet moving in the direction of the congestion. This bit can warn the destination that there is congestion. The receiver in this case can use policies, such as slowing down the acknowledgments, to alleviate the congestion.

Ans.(b) Techniques to Improve QoS

Some techniques that can be used to improve the quality of service are described here. We briefly discuss four common methods : scheduling, traffic shaping, and admission control and resource reservation.

1. Scheduling

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Several scheduling techniques are designed to improve the quality of service. We discuss three of them here : FIFO queuing, priority queuing and weighted fair queuing.

2. Traffic Shaping

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic : **leaky bucket** and **token bucket.**

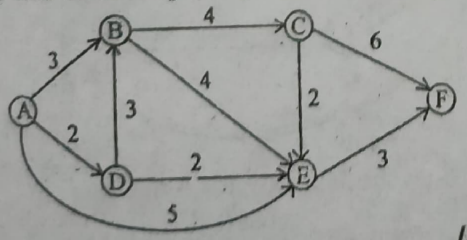
3. Resource Reservation

A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand.

4. Admission Control

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a low based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

Q.18 ***(a) Describe the working principles, advantages and disadvantages of the following routing algorithms: (i) Distance vector routing, (ii) Reverse path forwarding. (b) Consider the following network in figure with the indicated link cost. Use Dijkstra's shortest path algorithm to compute the shortest paths from A to C and F.***



[R.T.U. 2011]

Ans.(a) (i) Distance Vector Routing : *Refer to Q.10.*

(ii) Reverse Path Forwarding

In some applications, hosts need to send messages to many or all other hosts. For example, a service distributing weather reports, stock market updates, or live radio programs might work best by broadcasting to all machines and letting those that are interested read the data. Sending a packet to all destinations simultaneously is called **broadcasting**.

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

An example of reverse path forwarding is shown in Fig. part (a) shows a subnet, part (b) shows a sink tree for router I of that subnet, and part (c) shows how the reverse path algorithm works. On the first hop, I sends packets to F, H, J, and N, as indicated by the second row of the tree. Each of these packets arrives on the preferred path to I (assuming that the preferred path falls along the sink tree) and is so indicated by a circle around the letter. On the second hop, eight packets are generated, two by each of the routers that receives a packet on the first hop. As it turns out, all eight of these arrive at previously unvisited routers, and five of these arrive along the preferred line. Of the six packets generated on the third hop, only three arrive on the preferred path (at *C*, *E*, and *K*); the others are duplicates. After five hops and 24 packets, the broadcasting terminates, compared with four hops and 14 packets had the sink tree been followed exactly.

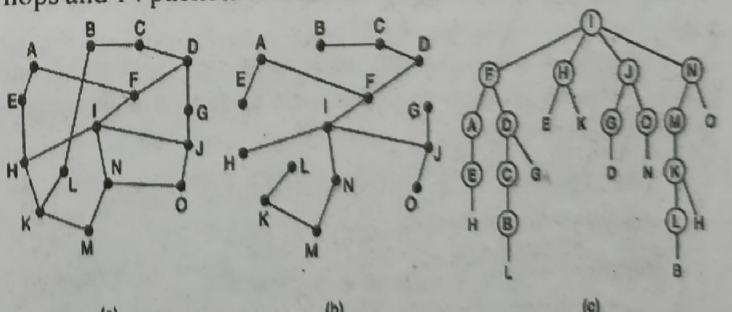


Fig. : Reverse path forwarding. (a) A subnet. (b) A sink tree. (c) The tree built by reverse path forwarding.

Advantage

The principal advantage of reverse path forwarding is that it is both reasonably efficient and easy to implement. It does not require routers to know about spanning trees, nor does it have the overhead of a destination list or bit map in each broadcast packet as does multidestination addressing. Nor does it require any special mechanism to stop the process, as flooding does (either a hop counter in each packet and a priori knowledge of the subnet diameter, or a list of packets already seen per source).

Disadvantages

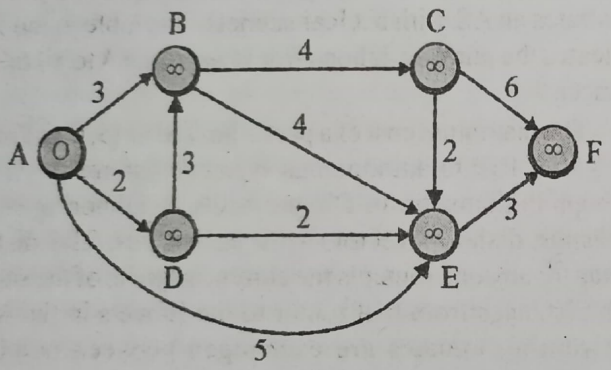
(i) The foremost disadvantage is that each broker in the network simultaneously belongs to multiple event dissemination trees, so the routing information in each broker can hardly be replaced by that of other brokers. Once a broker fails, the routing reconfiguration of the whole system will be very complex.

(ii) As it has link state routing dissemination strategy, recognized link changes cause node to flood control packets across the entire network which taxes network resources.

(iii) It generates a large number of duplicate packets while broadcasting.

Ans.(b) In Dijkstra algorithm first find out source that is A and destination is C and R

First define node weight ∞ excluding source. Source has 0 node weight.



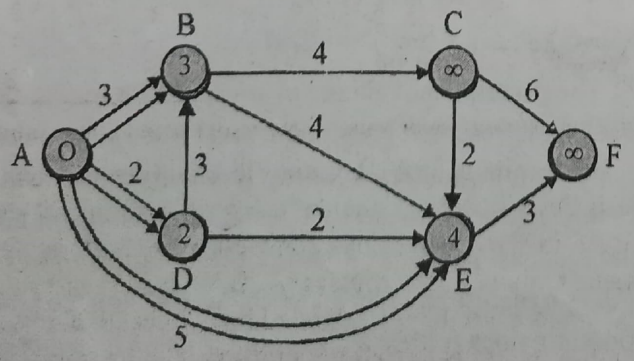
Find out the adjacency node of A, *i.e.*, B, D and E and also find the minimum value between nodeweight and edge + source node weight

i.e. for B $\min(\infty, 3+0) = 3$

for D $\min(\infty, 2+0) = 2$

for E $\min(\infty, 5+0) = 5$

So prepare a path between two so



Now find out minimum node from the list

B(3), D(2), E(5), C(∞), F(∞)

Minimum is D.

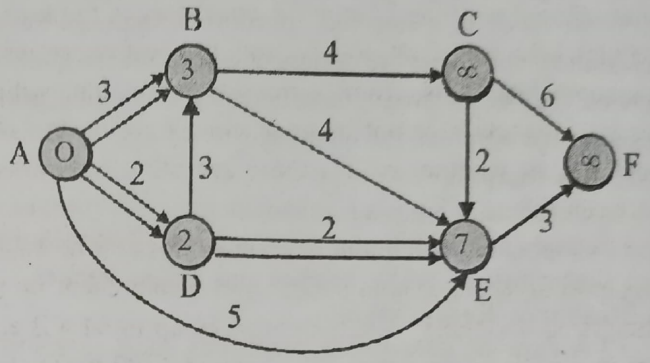
Adjacency node of D is B and E.

Find minimum value

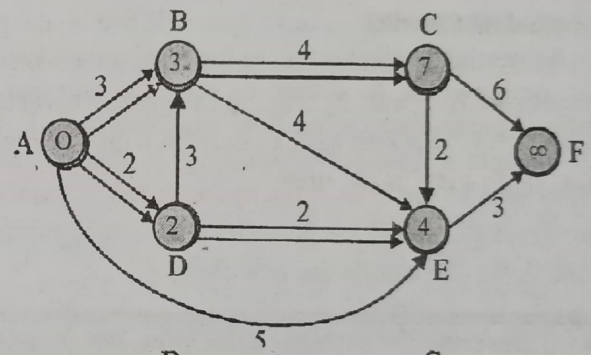
For B $\min(3, 2+3) = 3$

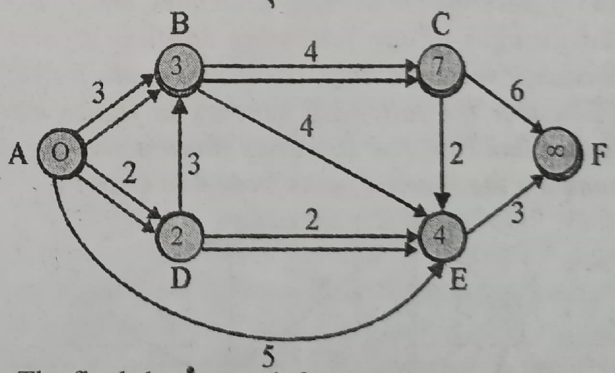
For E $\min(5, 2+2) = 4$

So for E, A to E path is replaced or cancelled and a new path is decided for E (D to E).



Repeat the process for rest vertex.





The final shortest path for A to C is

A – B – C cost is 7

A – D – E – F cost is 7

Q.19 ***Write a short note on Network layer in Internet.***

[R.T.U. 2009, Raj. Univ. (Suppl. 2004)]

Ans. At the network layer, the internet can be viewed as a collection of subnetworks or Autonomous Systems (ASes) that are connected together.

There is no real structure, but several major backbones exist. These are constructed from high bandwidth lines and

fast routers attached to the backbone are regional (midlevel) networks, and attached to these regional networks are the LANs at many universities, companies, and internet service providers. A sketch of this **quasihierarchical** organization is given in figure.

The glue that holds the internet together is the network layer protocol, IP (Internet Protocol). Unlike most older network layer protocols, it was designed from the beginning with internet working in mind. A good way to think of the network layer is this, its job is to provide a best efforts way to transport datagrams, from source to destination, without regard to whether or not these machines are on the same network, or whether or not there are other networks in between them.

Communication in the internet works as follows. The transport layer takes data stream and breaks them up into datagrams. In theory, datagrams can be up to 64 KB each, but in practice they are usually around 1500 bytes. Each datagram is transmitted through the internet, possibly being fragmented into smaller units as it goes. When all the pieces finally get to the destination machine, they are reassembled by the network layer into the original datagram. This datagram is then handed to the transport layer, which inserts it into the "receiving process" input stream.

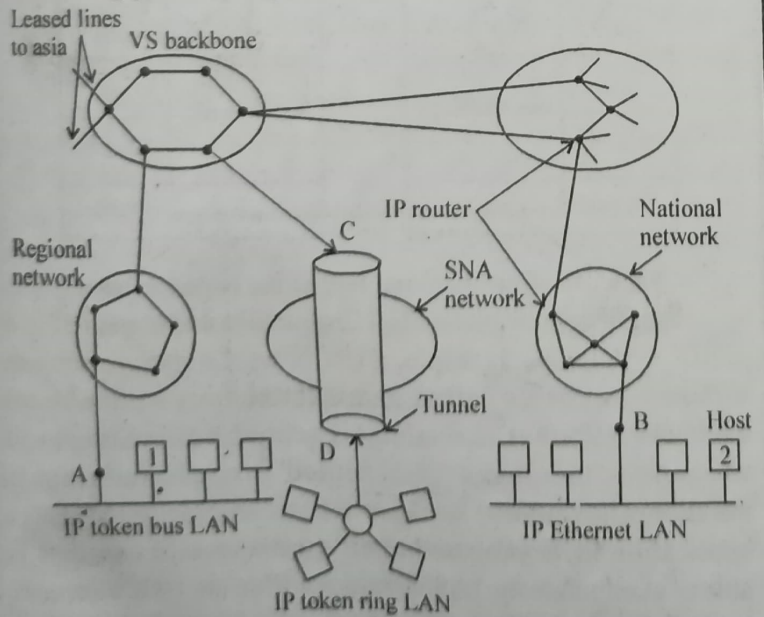


Fig. : The internet is an interconnected collection of many networks

Q.20 (a) Explain the Intra-Autonomous System Routing in the Internet.

OR

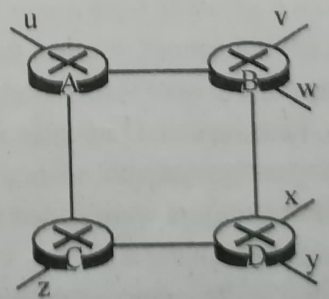
Explain OSPF and types of links defined by OSPF. *[Raj. Univ. 2006]*

(b) What are the roles played by IGMP protocol and a wide area multicast routing protocol. *[Raj. Univ. 2007]*

Ans.(a) Intra-Autonomous System Routing : AS routing protocol is used to determine how routing is performed within an Autonomous System (AS). Intra-AS routing protocols are also known as interior gateway protocol. Historically, two routing protocols have been used extensively for routing with an autonomous system in the Internet, the Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). A routing protocol closely related to OSPF is the IS - IS protocol.

1. RIP : RIP is a distance vector protocol that operates in a manner very close to the idealized DV protocol. The version of RIP specified in RFC 1058 uses hop count as a cost metric, that is , each link has a cost of 1. In the DV algorithm, costs were defined between pairs of routers. In RIP (and also in OSPF), costs are actually from source router to destination subject. RIP uses the term hop, which is the number of subnets traversed along the shortest path from source router to destination subnet, including the destination subnet. RIP uses the term hop, which is the number of subnets traversed along the shortest path from source router to destination subnet, including the destination subnet. Fig. 1 illustrates an AS with six leaf subnets. The table in the fig. 1 indicates the number of hops from the source A to all the leaf subnets.

The maximum cost of a part is limited to 15, thus limiting the use of RIP to autonomous systems that are fewer than 15 hops in diameter. In DV protocols, neighboring routers exchange distance vectors with each other. The distance vector for any one router is the current estimate of the shortest path distances from that router to the subnets in the AS. In RIP routing updates are exchanged between neighbors approximately every 30 seconds using a RIP response message. The response message sent by a router or host contains a list of upto 25 destination subnets within the AS, as well as the sender's distance to each of those subnets. Response messages are also known as RIP advertisements.



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

Fig. 1: Number of hops from source router A to various subnets

Let's take a look at a simple example of how RIP advertisements work. Consider the portion of an AS shown in fig. 2. In this figure, lines connecting the routers, denote subnets. Only selected routers (A, B, C and D) and subnets (w, x, y and z) are labeled. Dotted lines indicate that the AS continues on l thus this autonomous system has many more routers and links than are shown.

Each router maintains a RIP table known as a routing table. A router's routing table includes both the router's distance vector and the router's forwarding table. Fig. 3 shows the routing table for router D. Note that the routing table has three columns. The first column is for the destination subnet, the second column indicates the identity of the next router along the shortest path to the destination subnet, and the third column indicates, the number of hops (that is, the number of subnets that have to be traversed, including the destination subnet) to get to the destination subnet along the shortest path. The table in fig.3 and the subsequent tables to come, are only partially complete.

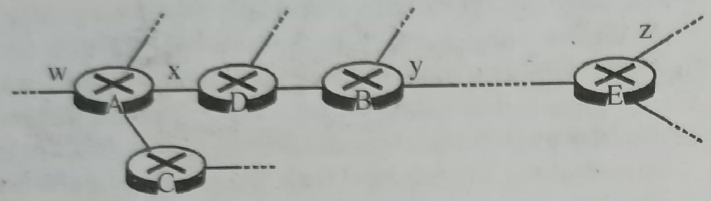


Fig. 2: A portion of an autonomous system

Destination Subnet	Next Router	Number of Hops to Destination
w	A	2
y	B	2
z	B	7
x	-	1
.....	.....	.....

Fig. 3 : Routing table in router D before receiving advertisement from router A

Now suppose that 30 seconds later, router D receives from router A the advertisement shown in fig. 4. Note that this advertisement is nothing other than the routing table information from router A. This information indicates, in particular that subnet z is only four hops away from router A. Router D, upon receiving this advertisement, merges that advertisement with the old routing A to subnet z that is shorter than the path thought router B. Thus, router D updates its routing table to account for the shorter shortest path, as shown in fig. 5 or perhaps new links and/or routers were added to the AS, thus changing the shortest paths in the AS.

Let's next consider a few of the implementation aspects of RIP. RIP routers exchange advertisements approximately every 30 seconds. If a router does not hear from its neighbor atleast once every 180 seconds, that neighbor is considered to be no longer reachable, that is either the neighbor has died or the connecting line has gone down. When this happens, RIP modifies the local routing table and then propagates this information by sending advertisements to its neighboring routers (the ones that are still reachable).

Destination Subnet	Next Router	Number of Hops to Destination
z	c	4
w	-	1
x	-	1
.....	.....	.....

Fig. 4 : Advertisement from router A

Destination Subnet	Next Router	Number of Hops to Destination
w	A	2
y	B	2
x	A	5
.....	.....	.....

Fig. 5 : Routing table in router D after receiving advertisement

A router can also request message. Routers send RIP request and response messages to each other over UDP using port number 520. The UDP segment is carried between routers in a standard IP datagram. The fact that RIP uses a transport layer protocol (UDP) on top of a network layer protocol (IP) to implement network layer functionality (a routing algorithm) may seem rather convoluted.

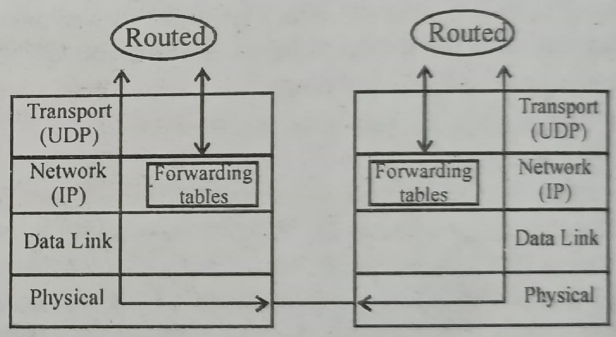


Fig.6 : Implementation of RIP as the routed daemon

Fig. 6 sketches how RIP is typically implemented in a UNIX system, for example, a UNIX workstation serving as a router. A process called routed (pronounced route deep) executes RIP, that is, maintains routing information and exchanges messages with routed process running in neighboring routers. Because RIP is implemented as an application layer process (albeit a very special one that is able to manipulate the routing tables within the UNIX kernel), it can send and receive messages over a standard, socket and use a standard transport protocol. RIP is implemented as an application layer protocol running over UDP

2. OSPF : Like RIP, OSPF routing is widely used for intra-AS routing in the Internet. OSPF and its closely related cousing, IS-IS are typically deployed in upper- tier ISP whereas RIP is deployed in lower-tier ISPs and enterprise networks. The open in OSPF indicates that the routing protocol specification is publicly available. The most recent version of OSPF version 2, is defined in RFC2328, a public document.

OSPF was conceived as the successor to RIP and as such has a number of advanced features. At its heart, however OSPF is a link-state protocol that uses flooding of link-state information and a Dijkstra least cost path algorithm. With OSPF, a router constructs a complete topological map (that is, a graph) of the entire autonomous system. The router then locally runs Dijkstra's shortest-path algorithm to determine a shortest-path tree to all subnets, with itself as the root node. Individual link costs are configured by the network administrator. The administrator might choose to set all link costs to 1, thus achieving minimum-hop routing, or might choose to set the link weights to be inversely proportional to link capacity in order to discourage traffic from using low-bandwidth links. OSPF does not mandate a policy for how link weights are set (that is the job of the network administrator), but instead provides the mechanisms (protocol) for determining least cost path routing for the given set of link weights.

With OSPF, a router broadcasts routing information to all other routers in the autonomous system, not just to its neighboring routers. A router broadcasts link state information whenever there is a change in a link's state (for example a change in cost or a change in up/down status). It also broadcasts a link's state periodically (at least once every 30 minutes), even if the link's state has not changed RFC 2328 notes that "this periodic updating of link state advertisements adds robustness to the link state algorithm "OSPF advertisements are contained in OSPF. Thus, the OSPF protocol must itself implement functionality such as reliable message transfer and link state broadcast. The OSPF protocol also checks that links are operational (via a HELLO message that is sent to an attached neighbour) and allows an OSPF router to obtain a neighboring router's database of network wide link state.

Some of the advance embodied in OSPF include the following :

Security : Exchanges between OSPF routers (for example, link-state updates) are authenticated. With aunthentication, only trusted routers can participate in the OSPF protocol within an AS, thus preventing malicious intruders (or networking students taking their new found knowledge out for a joyride) from injecting incorrect information into router tables. By default, OSPF packets between routers are not authenticated and could be forged. Two types of authentication can be configured simple and MD5. With simple authentiction. The same passwords is configured on each router. When a router sends an OSPF packet, it includes the password in plain text. Clearly, simple authentication is not secure. MD5 authentication is based on shared secret keys that are configured in all the routers. Each router computes an MD5 hash for each OSPF packet based on the content of the packet and the configured secret key. Then it includes the resulting hash value in the OSPF packet, The receiving router, using the preconstructed secret key, will compute an MD5 hash of the packet and compare it with the hash value the packet carries, thus verifying the packet's authenticity . Sequence numbers are also used with MD5 authentication to protect against replay attacks.

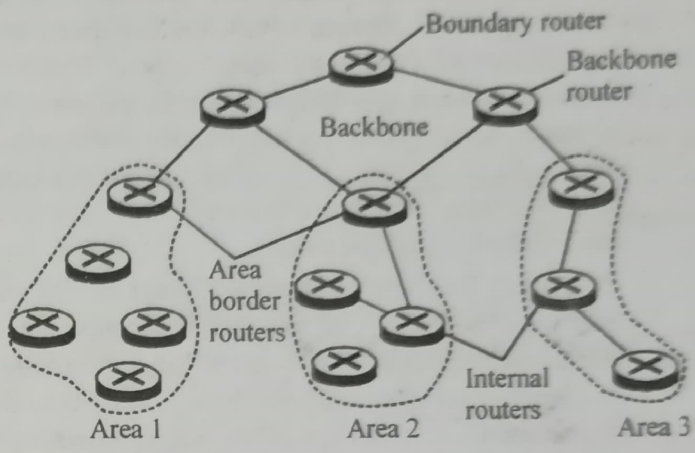


Fig.7 : Hierarchically structured OSPF AS

Multiple Same-cost Paths : When multiple paths to a destination have the same cost, OSPF allows multiple paths to be used (that is a, single path need not be chosen for carrying all traffic when multiple equal cost paths exits).

Integrated Support for Unicast and Multicast Routing : Multicast OSPF (MOSPF) provides simple extension to OSPF to provide for multicast routing MOSPF uses the existing OSPF link database and adds a new type of link-state advertisement to the existing OSPF link-state broadcast mechanism.

Support for Hierarchy within a Single Routing Domain : Perhaps the most significant advance in OSPF is the ability to structure an autonomous a system hierarchically.

An **OSPF Autonomous System** can be configured into areas. Each area runs its own OSPF link-state routing algorithm, with each router in an area broadcasting its link state to all other routers in that area. The internal details of an area thus remain invisible to all routers outside the area. Intra-area routing involves only those routers within the same area.

Within each area, one or more area border routers are responsible for routing packets outside the area. Exactly one OSPF area in the AS is configured to be the backbone area. The primary role of the backbone area is to route. Traffic between the other area in the AS. The backbone always contains all area border routers in the AS and may contain nonborder routers as well. Inter area routing within the AS requires that the packet be first routed to an area border (intra-area routing), then routed through the backbone to the area border router that is in the destination area and then routed to the final destination.

We can identify four types of OSPF router as follows :

- **Internal Routers :** These routers are in nonbackbone areas and perform only intra-AS routing
- **Area Border Routers :** These routers belong to both an area and the backbone.
- **Backbone Routers (nonborder routers) :** These routers perform routing within the backbone but themselves are not area border routers. Within a nonbackbone area, internal routers learn of the existence of routers to other areas from information (essentially a link-state advertisement, but advertising the cost of a route to another area, rather than a link cost) broadcast within the area by its backbone routers.
- **Boundary Routers :** A boundary router exchanges routing information with routers belonging to other autonomous systems. This router might, for example, use BGP to perform inter-AS routing. It is through such a boundary router that other routers learn about paths to external networks.

Ans.(b) Internet Group Management Protocol (IGMP): The IGMP protocol version3 operates between a host and its directly attached router (informally, we can think of the directly attached router as the first hop router that a host would see on a path to any other host outside its own local network or the last-hop router on any path to that host), as shown in fig. 1. Fig. 1 shows three first-hop multicast routers, each connected to its attached hosts via one outgoing local interface. This local interface is attached to a LAN in this example, and while each LAN has multiple attached hosts, at most a few of these hosts will typically belong to given multicast group at any given time.

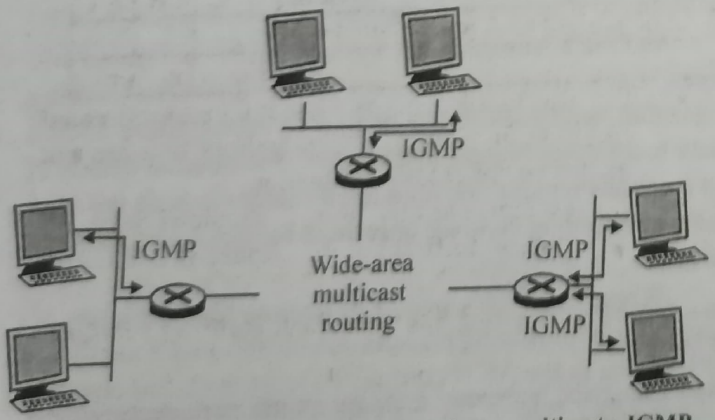


Fig.1 : The two components of network -layer multicast : IGMP and multicast routing protocols.

IGMP provides the means for host to inform its attached router that an application running on the host wants to join a specific multicast group. Given that the scope of IGMP interaction is limited to a host and its attached router, another protocol is clearly required to co-ordinate the multicast routers (including the attached routers) throughout the Internet, so that multicast datagrams are routed to their final destinations. This latter functionality is accomplished by network-layer multicast in the Internet thus consists of two complementary components : IGMP and multicast routing protocols.

Although IGMP's name suggests and manages the group of hosts joined to a multicast group, the name is a bit misleading since IGMP operates locally, between a host and an attached router. Despite its name IGMP is not a protocol that operates among all the host that have joined a multicast group. Indeed, there is no network layer multicast group membership protocol that operates among all the Internet hosts in a group. There is no network-layer protocol, for example, that allows a host to determine the identities of all of the other hosts, network-wide, that have joined the multicast group.

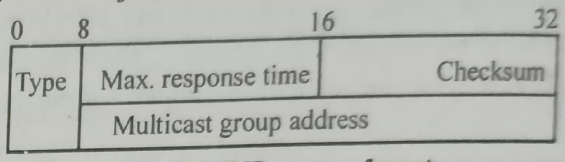


Fig.2: IGMP message format

IGMP has only three message types. The IGMP message format is summarized in above fig. Like ICMP, IGMP messages are carried encapsulated within an IP datagram, with an IP protocol number of 2. A general membership_query message is sent by a router to all hosts on that interface. A router can also determine whether a specific multicast group has been joined by hosts on an attached interface using a specific membership_query. The specific query includes the multicast address of the group being queried in the multicast group address field of the IGMP membership_query message, as shown in fig. Hosts respond to memersbership_query message with an IGMP membership_report message, as illustrated in figure. Memebrship_report messages can also be generated by a host when an application first joins a multicast group without waiting for a membership_query message from the router cast group without waiting for a membership_query message from the router.

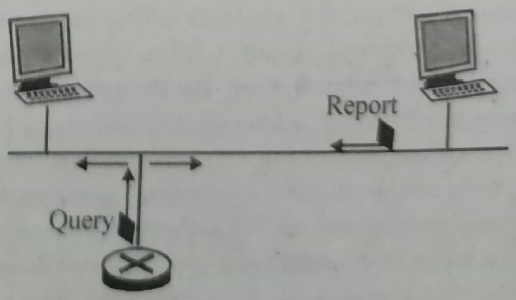


Fig.3: IGMP member query and membership report

The final type of IGMP message is the leave_group message. Interestingly, this message is optional, but if it is optional, how does a router detect that there are no longer

any hosts on an attached interface that are joined to a given multicast group. The answer to this question lies in the use of the IGMP membership_query message. The router infers that no hosts are joined into a given multicast group when no host responds to membership_query message with the given group address. This is an example of what is sometimes called soft-state in an Internet protocol. In a soft-sate protocol, the state (in this case of IGMP, the fact that there are hosts joined to a given multicast group) is removed via time-out event (in this case, via periodic membership_query message from the router) if it is not explicitly refreshed (in this case, by a membership_report message from an attached host). It has been argued that soft-state protocols result in simpler control than hard-state protocols which not only require state to be explicitly added and removed, but also require mechanisms to recover form the situation where the entity responsible for removing state has terminated prematurely or failed.

Q.21 ***(a) What is Distance Vector Routing? Explain count to infinity problem and give its solution?***

(b) Explain Flooding and shortest path routing an describe how and when they are used in link state routing.

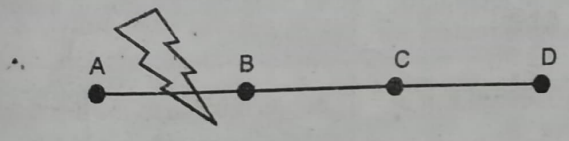
Ans. (a) Distance Vector Routing : *Refer to Q.10.*

Count to Infinity Problem and its Solution :

1. One of the important issue in Distance Vector Routing is County of Infinity Problem.
2. Counting to infinity is just another name for a routing loop.
3. In distance vector routing, routing loops usually occur when an interface goes down.
4. It can also occur when two routers send updates to each other at the same time.

Example

Link Between A and B is Broken



	A	B	C	D
A	0, -	1, A	2, B	3, C
B	1, B	0, -	2, C	3, D
C	2, B	1, C	0, -	1, C
D	3, B	2, C	1, D	0,-

- Imagine a network with a graph as shown above in figure.
- As you see in this graph, there is only one link between A and the other parts of the network.
- Now imagine that the link between A and B is cut.
- At this time, B corrects its table.
- After a specific amount of time, routers exchange their tables, and so B receives C's routing table.
- Since C doesn't know what has happened to the link between A and B, it says that it has a link to A with the weight of 2 (1 for C to B, and 1 for B to A it doesn't know B has no link to A).
- B receives this table and thinks there is a separate link between C and A, so it correct its table and changes infinity to 3 (1 for B to C, and 2 for C to A, as C said).
- Once again, routers exchange their tables,
- When C receives B's routing table, it sees that B has changed the weight of its link to A from 1 to 3, so C updates its table and changes the weight of the link to A to 4 (1 for C to B, and 3 for B to A, as B said).
- This process loops until all nodes find out that the weight of link to A is infinity.
- This situation is shown in the table below.
- In this way, Distance Vector Algorithms have a slow convergence rate.

	B	C	D
Sum of Weight to A after link cut	∞, A	2, B	3 C
Sum of Weight to A after 1st updating	3,C	2,B	3,C
Sum of Weight to A after 2nd updating	3, C	4, B	3, C
Sum of Weight to A after 3rd updating	5, C	4, B	5 C
Sum of Weight to A after 4th updating	5, C	6, B	5, C
Sum of Weight to A after 5th updating	7. C	6 B	7, C
Sum of Weight to A after nth updating	....	....	...
∞	∞	∞	∞

- One way to solve this problem is for routers to send information only to the neighbors that are not exclusive links to the destination.
- For example, in this case, C shouldn't send any information to B about A, because B is the only way to A.

Ans. (b) Flooding and Shortest Path Routing : *Refer to Q.16(b).*

Q.22 ***(a) What are the basic design issues of a Network layer? Also explain what are the services provided by the network layer to the transport layer?***

(b) Differentiate between the static and dynamic routing with their pros and cons. Give examples of some routing protocol used in both type of routing.

Ans. (a) Design Issues Faced by Transport Layer are:

- **Induced traffic:** The traffic at any given link (or path) due to the traffic through neighboring links (or paths) is referred to as induced traffic. This is due to the broadcast nature of the channel and the location-dependent contention on the channel. This induced traffic affects the throughput achieved by the transport layer protocol.
- **Induced Throughput Unfairness:** This refers to the throughput unfairness at the transport layer due to the throughput/delay unfairness existing at the lower layers such as the network and MAC layers.
- **Separation of congestion control, reliability, and flow control:** A transport layer protocol can provide better performance if end-to-end reliability, flow control, and congestion control are handled separately. Reliability and flow control are end-to-end activities, whereas congestion can at times be a local activity. The transport layer flow can experience congestion with just one intermediate link under congestion.
- **Power and Bandwidth Constraints:** Nodes in ad hoc wireless networks face resource constraints including the two most important resources: (i) power source and (ii) bandwidth. The performance of a transport layer protocol is significantly affected by these constraints.
- **Misinterpretation of Congestion:** Traditional mechanisms of detecting congestion in networks, such as packet loss and retransmission timeout, are not suitable for detecting the network congestion in ad hoc wireless networks.
- **Completely Decoupled Transport Layer:** Another challenge faced by a transport layer protocol is the interaction with the lower layers. Wired network transport layer protocols are almost completely decoupled from the lower layers.
- **Dynamic Topology:** Some of the deployment scenarios of ad hoc wireless networks experience rapidly changing network topology due to the mobility of nodes. This can lead to frequent path breaks, partitioning and remerging of networks, and high delay in reestablishment of paths.

Services provided by network layer to transport layer are:

There are two types of service that can be provided by the network layer:

- an unreliable connectionless service
- a connection-oriented, reliable or unreliable, service

Connection-oriented services have been popular with technologies such as X.25 and ATM or frame-relay, but nowadays most networks use an unreliable connectionless service.

Ans. (b) Differences between Static Routing and Dynamic Routing:

Static routing	Dynamic Routing
It is ideal for small networks.	It is suitable for large networks.
Configurations of static routes involve less cost. It can be easily maintained by the network administrator.	Dynamic routing involves cost in terms of CPU processes and bandwidth used on the network links.
Routes cannot be changed until authorized by the administrator.	Routing protocols find the best routes for traversing the packets to their destination.
Routes are not updated dynamically in the routing table and hence cannot detect inactive routes.	Routing protocols update the routing table with the update of the latest update on the routes.

Static Routing Advantages and Disadvantages:

Advantages	Disadvantages
Easy to implement in a small network.	Suitable for simple topologies or for special purposes such as a default static route.
Very secure. No advertisements are sent, unlike with dynamic routing protocols.	Configuration complexity increases dramatically as the network grows. Managing the static configurations in large networks can become time consuming.
It is very predictable, as the route to the destination is always the same.	If a link fails, a static route cannot reroute traffic. Therefore, manual intervention is required to re-route traffic.
No routing algorithm or update mechanisms are required. Therefore, extra resources (CPU and memory) are not required.	

Dynamic Routing Advantages and Disadvantages:

Advantages	Disadvantages
Suitable in all topologies where multiple routers are required.	Can be more complex to initially implement.
Generally independent of the network size.	Less secure due to the broadcast and multicast routing updates. Additional configuration settings such as passive interfaces and routing protocol authentication are required to increase security. Route depends on the current topology.
Automatically adapts topology to reroute traffic if possible.	Requires additional resources such as CPU, memory, and link bandwidth.

Examples of Some Routing Protocol used in both Type of Routing :

- Interior Gateway Protocols (IGP): It is used for routing within an AS. It is also referred to as intra-AS routing. Companies, organizations, and even service providers use an IOP on their internal networks. IGPs include RIP, EIGRP, OSPF, and IS-IS.
- Exterior Gateway Protocols (EGP): It is used for routing between autonomous systems. It is also referred to as inter-AS routing. Service providers and large companies may interconnect using an EGR. The Border Gateway Protocol (BGP) is the only currently viable EGP and is the oficial routing protocol used by the Internet.

A router using a distance vector routing protocol does not have the knowledge of the entire path to a destination network. Distance vector protocols use routers as sign posts along the path to the final destination. The only information a router knows about a remote network is the distance or metric to reach that network and which path or interlace to use to get there. Distance vector routing protocols do not have an actual map of the network topology.

There are four distance vector IPv4 IOPs:

- **RIPv1 :** First generation legacy protocol
- **RIPv2 :** Simple distance vector routing protocol
- **IGRP :** First generation Cisco proprietary protocol (obsolete and replaced by EIORP)
- **EIGRP :** Advanced version of distance vector routing

Link-state protocols work best in situations where:

- The network design is hierarchical, usually occurring in large networks
- Fast convergence of the network is crucial
- The administrators have good knowledge of the implemented link-state routing protocol

There are two link-state IPv4 IOPs:

- OSPF Popular standards-based routing protocol
- IS-IS: Popular in provider networks

Q.23 *Explain Address Mapping in detail.*

Ans. Address Mapping : An internet is made of a combination of physical networks connected by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host. The hosts and routers are recognized at the network level by their logical (IP) addresses.

However, packets pass through physical networks to reach these hosts and routers. At the physical level, the hosts and routers are recognized by their physical addresses.

A physical address is a local address. Its jurisdiction is a local network. It must be unique locally, but is not necessarily unique universally. It is called a *physical* address because it is usually (but not always) implemented in hardware. An example of a physical address is the 48-bit MAC address in the Ethernet protocol, which is imprinted on the NIC installed in the host or router.

The physical address and the logical address are two different identifiers. We need both because a physical network such as Ethernet can have two different protocols at the network layer such as IP and IPX (Novell) at the same time. Likewise, a packet at a network layer such as IP may pass through different physical networks such as Ethernet and LocalTalk (Apple).

This means that delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done by using either static or dynamic mapping.

Static mapping involves in the creation of a table that associates a logical address with a physical address. This table is stored in each machine on the network. Each machine that knows, for example, the IP address of another machine but not its physical address can look it up in the table. This has some limitations because physical addresses may change in the following ways:

1. A machine could change its NIC, resulting in a new physical address.
2. In some LANs, such as LocalTalk, the physical address changes every time the computer is turned on.
3. A mobile computer can move from one physical network to another, resulting in a change in its physical address.

To implement these changes, a static mapping table must be updated periodically. This overhead could affect network performance.

In dynamic mapping each time a machine knows one of the two addresses (logical or physical), it can use a protocol to find the other one.

Mapping Logical to Physical Address: ARP : *Refer to Q.5.*

Encapsulation : An ARP packet is encapsulated directly into a data link frame. For example, in Figure 1 an ARP packet is encapsulated in an Ethernet frame. Note that the type field indicates that the data carried by the frame are an ARP packet.

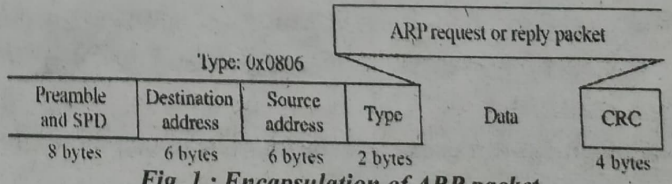


Fig. 1 : Encapsulation of ARP packet

Operation : Let us see how ARP functions on a typical internet. First we describe the steps involved. Then we discuss the four cases in which a host or router needs to use ARP. These are the steps involved in an ARP process :

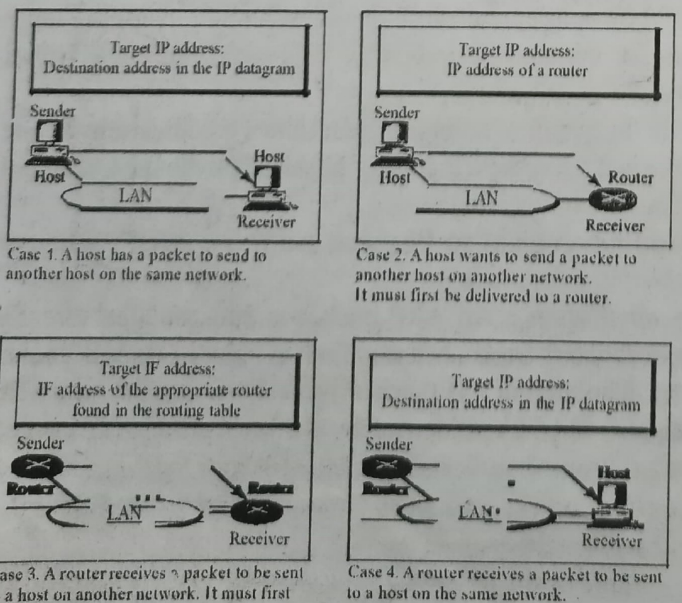
1. The sender knows the IP address of the target. We will see how the sender obtains this shortly.
2. IP asks ARP to create an ARP request message, filling in the sender physical address, the sender IP address, and the target IP address. The target physical address field is filled with Os.
3. The message is passed to the data link layer where it is encapsulated in a frame by using the physical address of the sender as the source address and the physical broadcast address as the destination address.
4. Every host or router receives the frame. Because the frame contains a broadcast destination address, all stations remove the message and pass it to ARP. All machines except the one targeted drop the packet. The target machine recognizes its IP address.
5. The target machine replies with an ARP reply message that contains its physical address. The message is unicast.
6. The sender receives the reply message. It now knows the physical address of the target machine.
7. The IP datagram, which carries data for the target machine, is now encapsulated in a frame and is unicast to the destination.

Four Different Cases

The following are four different cases in which the services of ARP can be used (see Figure 2).

1. The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header.
2. The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table, it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.
3. The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router. The IP address of the next router becomes the logical address that must be mapped to a physical address.
4. The sender is a router that has received a datagram destined for a host on the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.

An ARP request is broadcast; an ARP reply is unicast.



Case 1. A host has a packet to send to another host on the same network.

Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.

Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.

Case 4. A router receives a packet to be sent to a host on the same network.

Fig. 2 : Four cases using ARP

Example : A host with IP address 130.23.43.20 and physical address B2:34:55: 10:22: 10 has a packet to send to another host with IP address 130.23.43.25 and physical address A4:6E:F4:59:83:AB (which is unknown to the first host). The two hosts are on the same Ethernet network. Show the ARP request and reply packets encapsulated in Ethernet frames.

Solution : Figure 3 shows the ARP request and reply packets. Note that the ARP data field in this case is 28 bytes, and that the individual addresses do not fit in the 4-byte boundary. That is why we do not show the regular 4-byte boundaries for these addresses.

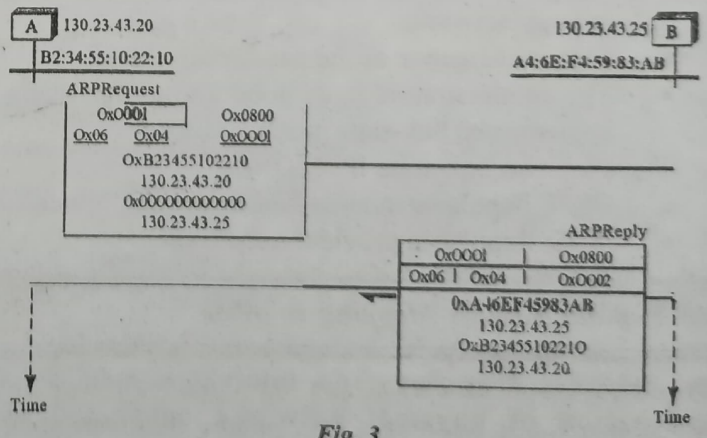


Fig. 3

ProxyARP : A technique called *proxy* ARP is used to create a subnetting effect. A proxy ARP is an ARP that acts on behalf of a set of hosts. Whenever a router running a proxy ARP receives an ARP request looking for the IP address of one of these hosts, the router sends an ARP reply announcing its own hardware (physical) address. After the router receives the actual IP packet, it sends the packet to the appropriate host or router.

Let us give an example. In Figure 4 the ARP installed on the right-hand host will answer only to anARP request with a target IP address of 141.23.56.23.

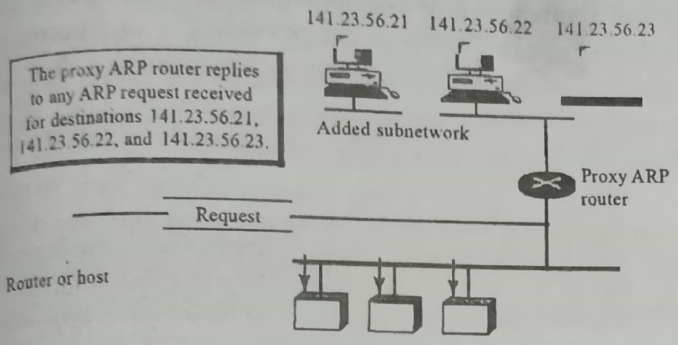


Fig. 4 : Proxy ARP

However, the administrator may need to create a subnet without changing the whole system to recognize subnetted addresses. One solution is to add a router running a proxy ARP. In this case, the router acts on behalf of all the hosts installed on the subnet. When it receives an ARP request with a target IP address that matches the address of one of its proteges (141.23.56.21, 141.23.56.22, or 141.23.56.23), it sends an ARP reply and announces its hardware address as the target hardware address. When the router receives the IP packet, it sends the packet to the appropriate host.

Mapping Physical to Logical Address: RARP, BOOTP, and DHCP

There are occasions in which a host knows its physical address, but needs to know its logical address. This may happen in two cases:

1. A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
2. An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

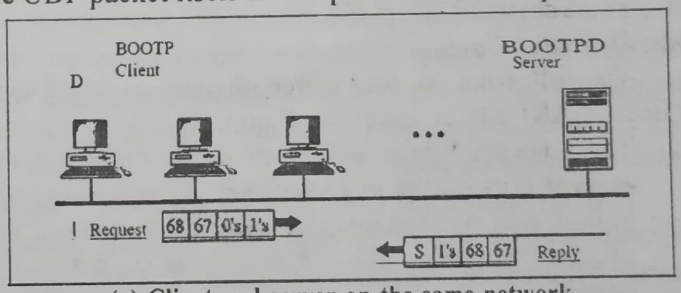
RARP: Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine. To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.

However, a diskless machine is usually booted from ROM, which has minimum booting information. The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator.

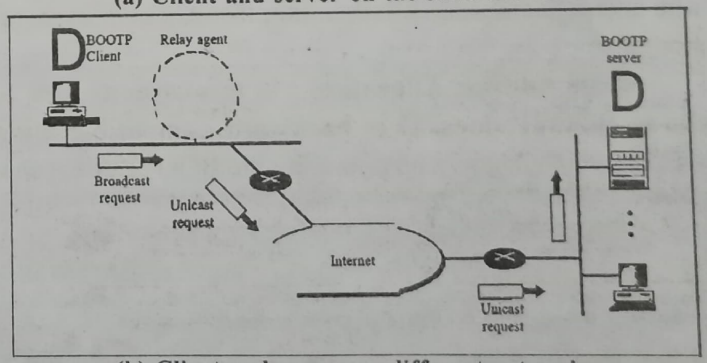
The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol. A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.

There is a serious problem with RARP: Broadcasting is done at the data link layer. The physical broadcast address, all is in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This is the reason that RARP is almost obsolete. Two protocols, BOOTP and DHCP, are replacing RARP.

BOOTP: The Bootstrap Protocol (BOOTP) is a client/server protocol designed to provide physical address to logical address mapping. BOOTP is an application layer protocol. The administrator may put the client and the server on the same network or on different networks, as shown in Figure 5. BOOTP messages are encapsulated in a UDP packet, and the UDP packet itself is encapsulated in an IP packet.



(a) Client and server on the same network



(b) Client and server on different networks

Fig. 5 : BOOTP client and server on the same and different network

The reader may ask how a client can send an IP datagram when it knows neither its own IP address (the source address) nor the server's IP address (the destination address). The client simply uses all as the source address and all as the destination address.

One of the advantages of BOOTP over RARP is that the client and server are application-layer processes. As in other application-layer processes, a client can be in one

network and the server in another, separated by several other networks. However, there is one problem that must be solved. The BOOTP request is broadcast because the client does not know the IP address of the server. A broadcast IP datagram cannot pass through any router. To solve the problem, there is a need for an intermediary. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay. The host in this case is called a relay agent. The relay agent knows the unicast address of a BOOTP server. When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the BOOTP server. The packet, carrying a unicast destination address, is routed by any router and reaches th1e BOOTP server. The BOOTP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent. The relay agent, after receiving the reply, sends it to the BOOTP client.

DHCP : BOOTP is not a dynamic configuration protocol. When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client already exists. The binding is predetermined.

However, what if a host moves from one physical network to another? What if a host wants a temporary IP address? BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. BOOTP is a static configuration protocol.

The Dynamic Host Configuration Protocol (DHCP) has been devised to provide static and dynamic address allocation that can be manual or automatic. DHCP provides static and dynamic address allocation that can be manual or automatic.

Static Address Allocation : In this capacity DHCP acts as BOOTP does. It is backward compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

Dynamic Address Allocation : DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.

When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.

The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (as is a subscriber to a service provider). DHCP provides temporary IP addresses for a limited time.

The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

Manual and Automatic Configuration One major problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes. DHCP, on the other hand, allows both manual and automatic configurations. Static addresses are created manually~ dynamic addresses are created automatically.

❑❑❑

TRANSPORT LAYER 4

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 ***Differentiaate between connectionless and connection-oriented service.*** ***[R.T.U. 2019]***

Ans. Difference between Connection oriented and Connectionless service

1. In connection oriented service authentication is needed, while connectionless service does not need any authentication.
2. Connection oriented protocol makes a connection and checks whether message is received or not and sends again if an error occurs, while connectionless service protocol does not guarantees a message delivery.
3. Connection oriented service is more reliable than connectionless service.
4. Connection oriented service interface is stream based and connectionless is message based.

Q.2 ***What do you mean by connectionless transport in transport layer.***

Ans. Connectionless Transport (UDP) : UDP is a connectionless, unreliable protocol that has no flow and error control. It performs very limited error checking. It uses port numbers to multiplex data from the application layer.

Q.3 ***What is segment structure of UDP.***

Ans. Segment Structure : UDP packets, called user datagram, have a fixed–size header of 8 bytes.

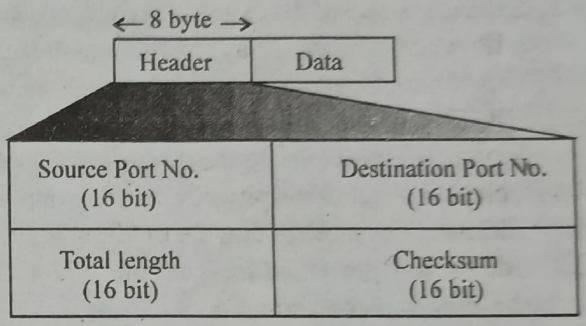


Fig.

- **Source Port Number :** This is the port number used by the process running on the source host.
- **Destination Port Number :** This is the port number used by the process running on the destination host.
- **Length :** This is a 16 bit field that defines the total length of the user datagram (Header + Data)
- **Checksum :** This field is used to detect errors over the entire user datagram (Header + Data).

Q.4 ***Write two difference between Token Bucket & Leaky Bucket.***

Ans. Difference between Token Bucket and Leaky Bucket Algorithms

S. No.	Token Bucket Algorithm	Leaky Bucket Algorithm
1.	Token bucket throws away tokens when the bucket is full but never discards packet.	Leaky bucket discards packets when the bucket is full.
2.	Token bucket allows saving upto a maximum size of bucket n. This means that bursts of upto n packets can be sent at once, giving faster response to sudden bursts of input.	Leaky bucket also gives response to sudden bursts of input.

PART-B

Q.5 ***Briefly discuss the transport layer services.***

[R.T.U. 2015, Raj. Univ. 2004]

Ans. Transport Layer Services : The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective service to its users, normally processes in the application layer. To achieve this goal, the transport layer makes use of the services provided by the network layer. The hardware and/or software within the transport layer that does the work are called the transport entity.

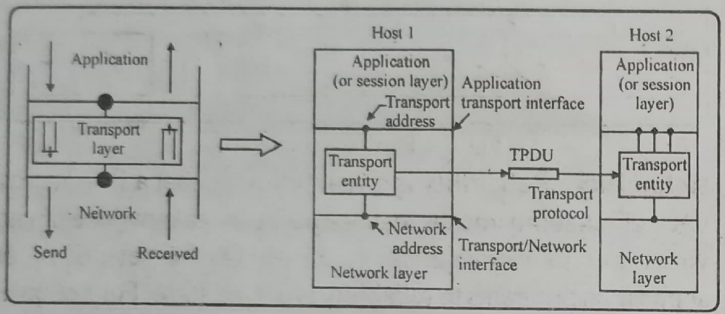


Fig. : Relationship between Network, Transport and Application layer

Transport entity can be located in the operating system kernel, in a separate user process, in a library package bound into the network applications, or conceivably on the Network Interface Card (NIC). The logical relationship of the network, transport and application layers is illustrated in fig.

Transport layer services are implemented by a transport protocol used between two transport entities. The services provided by transport layer protocols are divided into following categories :

1. Addressing : At the same time, computer often runs many programs. For example, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process on one computer to a specific process on the other. The transport layer header must therefore include a type of address called service-point address or port address. That's why; this addressing is also called as Service Point Addressing. The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.

2. Segmentation and Reassembly : Message is divided into transmittable segments at the source machine, where each segment contains a sequence number. These sequence numbers enable transport layer at the receiving machine to re-assemble message correctly at the destination and to identify and replace lost packets.

3. Connection Control : Transport layer protocol services may be divided into following two categories :

(a) **Connection-oriented Service :** A connection-oriented transport protocol establishes a connection *i.e.* virtual path between source and destination. Over this path, each packet is delivered to the transport layer of the destination machine. And after sending the complete message, connection is released. **TCP** is the transport layer protocol which provides connection-oriented service in the Internet.

(b) **Connectionless Service :** A connectionless transport protocol will treat each packet independently because there is no connection between source and destination. Each packet can take its own different route. **UDP** is the transport layer protocol which provides connectionless service in the Internet.

4. Flow Control : Like data link layer, transport layer is also responsible for flow control. Flow control is performed end-to-end, rather than across a single link.

5. Error Control : Like data link layer, transport layer is also responsible for error control. Error control is performed end-to-end. The sending transport layer makes sure that entire message is transmit without any damage, loss or duplication. If any error occurs, error correction is usually achieved through retransmission.

Q.6 ***Draw the format of the UDP header and explain in brief the various fields.*** *[R.T.U. 2015, 11]*

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

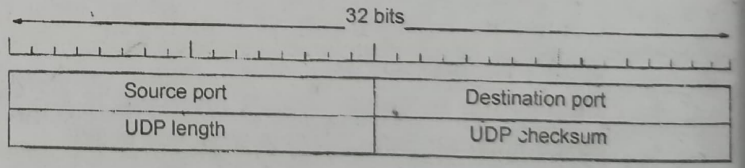


Fig. : The UDP header

UDP transmits segments consisting of an 8 byte header followed by the payload. The header is shown in figure. The two ports serve to identify the end points within the source and destination machines. When a UDP packet arrives, its payload is handed to the process attached to the destination port. This attachment occurs when BIND primitive or something similar is used. In fact, the main value of having UDP over just using raw IP is the addition of the source and destination ports. Without the port fields, the transport layer would not know what to do with the packet. With them, it delivers segments correctly.

The *source port* is primarily needed when a reply must be sent back to the source. By copying the *source port* field

from the incoming segment into the *destination port* field of the outgoing segment, the process sending the reply can specify which process on the sending machine is to get it.

The *UDP length* field includes the 8-byte header and the data. The *UDP checksum* is optional and stored as 0 if not computed (a true computed 0 is stored as all 1s). Turning it off is foolish unless the quality of the data does not matter (e.g., digitized speech).

It is probably worth mentioning explicitly some of the things that UDP does not do. It does not do flow control, error control, or retransmission upon receipt of a bad segment. All of that is upto the user processes. What it does do is provide an interface to the IP protocol with the added feature of demultiplexing multiple processes using the ports. That is all it does. For applications that need to have precise control over the packet flow, error control, or timing.

One area where UDP is especailly useful is in client-server situations. Often, the client sends a short request to the server and expects a short reply back. If either the request or reply is lost, the client can just time out and try again. Not only is the code simple, but fewer messages are required (one in each direction) than with a protocol requiring an initial setup.

An application that uses UDP this way is DNS (the Domain Name System), a program that needs to look up the IP address of some host name to a DNS server. The server replies with a UDP packet containing the host's IP address. No setup is needed in advance and no release is needed afterward. Just two messages go over the network.

Q.7 *Explain Quality of service for transport layer.*

[R.T.U. 2015, 13]

Ans. Quality of Service (QoS) : In the field of computer networking and other packet-switched telecommunication networks, the traffic engineering term quality of service (QoS) refers to resource reservation control mechanisms rather than the achieved service quality. Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. For example, a required bit rate, delay, jitter, packet dropping probability and/or bit error rate may be guaranteed. Quality of service guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IPTV, since these often require fixed bit rate and are delay sensitive, and in networks where the capacity is a limited resource, for example in cellular data communication.

QoS is sometimes used as a quality measure, with many alternative definitions, rather than referring to the ability to reserve resources. Quality of service sometimes refers to the level of quality of service, i.e. the guaranteed service quality. High QoS is often confused with a high level of performance or achieved service quality, for example high bit rate, low latency and low bit error probability.

Quality of service (QoS) is an internetworking issue that has been discussed more than defined. We can informally define quality of service as something a flow seeks to attain.

Flow Characteristics : Traditionally, four types of characteristics are attributed to a flow : reliability, delay, jitter, and bandwidth, as shown in fig. 1.

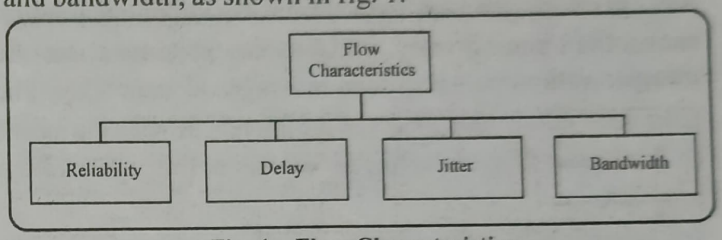


Fig. 1 : Flow Characteristics

Reliability : Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledegment, which entails retransmission. However, the sensitivity of application programs to reliability is not the same. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

Delay : Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

Jitter : Jitter is the variation in delay for packets belonging to the same flow. For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time. On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays : 21,22, 19, and 24.

For applications such as audio and video, the first case is completely acceptable; the second case is not. For these applications, it does not matter if the packets arrive with a short or long delay as long as the delay is the same for all packets. For this application, the second case is not acceptable.

Jitter is defined as the variation in the packet delay. High jitter means the difference between delays is large; low jitter means the variation is small. If the jitter is high, some action is needed in order to use the received data.

Bandwidth : Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.

Q.8 ***Explain the difference between connection oriented and connection less services.***

[R.T.U. 2013, Raj. Univ. 2006]

OR

Describe how the internet connection oriented services provides reliable transport. *[Raj. Univ. 2007]*

OR

How connection oriented and connection less services are implemented in network layer ? Discuss.[R.T.U. 2014]

Ans. Each system uses the internet to communicate with each other. Specifically, end–system programs use the services of the internet to send messages to each other. The links, routers and other pieces of the interent provide the means to transport these messages between the end system programs.

Internet or TCP/IP network, provide two types of services to end–system applications : Connection–less services and Connection–oriented services. A developer creating an internet application must design the application to use one of these two services.

Connection–Oriented Service : When an application uses the connection–oriented service, the client program and the server program sends control packets to each other before sending packets with real data. This so–called handshaking procedure alerts the client and server, allowing them to prepare for an onslaught of packets. Once the handshaking procedure is finished, a connection is said to be established between the two end–systems.

The internet's connection–oriented service comes bundled with several other services, including reliable data transfer, flow control and congestion control.

The internet's connection–oriended service has a name–TCP (Transmission Control Protocol). The services that TCP provides to an application include reliable transport, flow control and congestion control.

By **reliable data transfer**, we mean that an application can rely on the connection to deliver all of its data with outer or and in the proper order. Reliability in the internet is achieved through these of acknowledgements and retransmission.

Flow Control makes sure that neither side of a connection overwhelms the other side by sending two many packets too fast. The flow control service forces the sending end system to reduce its rate whenever there is such a risk. The internet implements the flow–control service by using sender and receiver buffers in the communicating end–systems.

The internet's **Congestion Control** service helps to prevent the internet from entering a state of gridlock. When a router becomes congested, its buffers can overflow and packet loss can occur. In such circumstances, if every pair of communicating end–system continues to pump packets into the network as fast as they can, gridlock sets in and few packets are delivered to their destinations.

Connection-less Service : There is no–handshaking with the internet's connectionless service. When one side of an application wants to send packets to the other side of the application, the sending program simply sends the packets. Connectionless services are faster than connection–oriented, because no–handshaking is required.

But there is no reliable data transfer either, so a source never knows for sure which packets have arrived at the destination. Moreover, the internet's connectionless service makes no provision for flow control or congestion control.

The Internet's connectionless service is called UDP (User Datagram Protocol). Most of the more familiar internet applications use TCP, the internet connection–oriented service.

PART-C

Q.9 ***Explain the leaky bucket algorithm with the help of suitable diagrams.*** *[R.T.U. 2019]*

Ans. Leaky Bucket Algorithm : Leaky bucket algorithm is used to control the transmission rate in a network. It can be used to check data transmissions, in the form of packets to define limits on bandwidth. By setting the bucket size and output flow rate, the code can be refined to slow, medium and fast.

Imagine a bucket with a small hole in the bottom, as illustrated in Fig. (a). No matter at what rate water enters the bucket, the outflow is at a constant rate, ρ, when there is any water in the bucket and zero when the bucket is empty. Also, once the bucket is full, any additional water entering it spills over the sides and is lost (i.e., does not appear in the output stream under the hole).

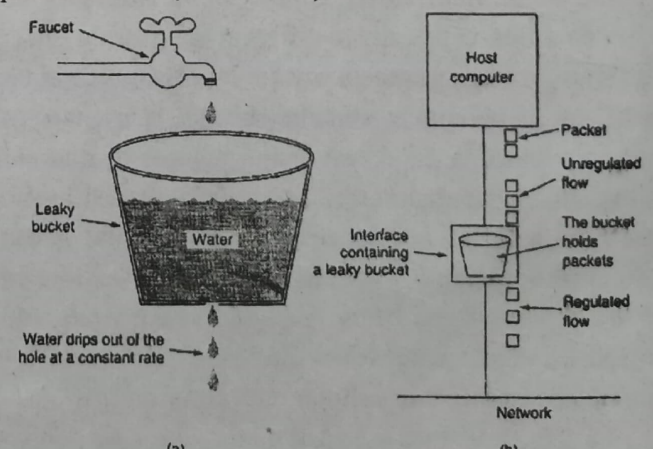


Fig. : (a) A leaky bucket with water; (b) A leaky bucket with packet.

The same idea can be applied to packets, as shown in Fig. (b). Conceptually, each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more processes within the host try to send a packet when the maximum number are already queued, the new packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulated by the host operating system. It was first proposed by Turner (1986) and is called the **leaky bucket algorithm**. In fact, it is nothing other than a single-server queuing system with constant service time.

The host is allowed to put one packet per clock tick onto the network. Again, this can be enforced by the interface card or by the operating system. This mechanism turns an uneven flow of packets from the user processes inside the host into an even flow of packets onto the network, smoothing out bursts and greatly reducing the chances of congestion.

When the packets are all the same size (e.g.. ATM cells), this algorithm can be used as described. However, when variable-sized packets are being used, it is often better to allow a fixed number of bytes per tick, rather than just one packet. Thus if the rule is 1024 bytes per tick, a single 1024-byte packet can be admitted on a tick, two 512-byte packets, four 256-byte packets, and so on. If the residual byte count is too low, the next packet must wait until the next tick. Implementing the original leaky bucket algorithm is easy. The leaky bucket consists of a finite queue. When a packet arrives, if there is room on the queue it is appended to the queue; otherwise, it is discarded. At every clock tick, one packet is transmitted (unless the queue is empty).

The byte-counting leaky bucket is implemented almost the same way. At each tick, a counter is initialized to n. If the first packet on the queue has fewer bytes than the current value of the counter, it is transmitted, and the counter is decremented by that number of bytes. Additional packets may also be sent, as long as the counter is high enough. When the counter drops below the length of the next packet on the queue, transmission stops until the next tick, at which time the residual byte count is overwritten and lost.

Q.10 ***Discuss the reason of congestion in a network. Also discuss Leaky bucket and Token bucket algorithms in detail.*** ***[R.T.U. 2016]***

Ans. Congestion Control Algorithms : Congestion in a network may occur if the load on the network (the number of packets sent to the network) is greater than the capacity of the network (the number of packets a network can handle). Congestion in a network occurs because routers and switches have queue buffers that hold the packets before and after processing.

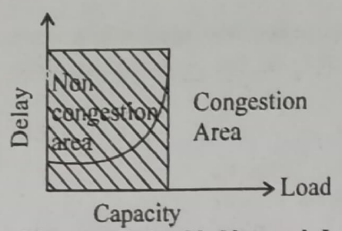


Fig. (a) : Delay Vs Network Load

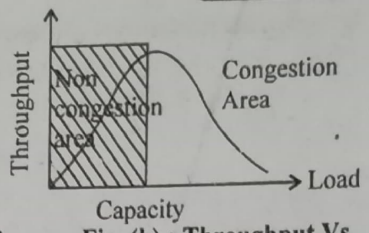


Fig. (b) : Throughput Vs Network Load

Congestion control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened. Congestion control can be divided into open loop congestion control (prevention) and closed loop control.

Token Bucket Algorithm

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. All of these have in common the fact that they make decisions without regard to the current state of the network. The closed loop control is based on the concept of a feedback loop.

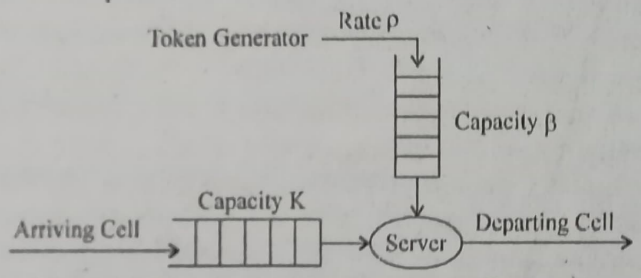


Fig. (c) : Token bucket algorithm for traffic shaping

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 place where action can be taken.

(3) Adjust system operation to correct the problem.

Congestion control involves two factors that measure the performance of a network : throughput and delay

The figure illustrates the basic principle of the token bucket. A token generator produces tokens at a rate of ρ tokens per second and places these in the token bucket, which has a maximum capacity of β tokens. Cell arriving from the source are placed in a buffer with a maximum capacity of K cells. To transmit a cell through the server, one token must be removed from the bucket. If the token bucket is empty, the cell is queued waiting for the next token.

The result of this scheme is that if there is a back log or cells and an empty bucket, then cells are emitted at a smooth flow of ρ cells per second with no cell delay variation until the back log is cleared. Thus, the token bucket smooths out bursts of cells.

Leaky Bucket Algorithm : *Refer to Q.9.*

Q.11 ***Describe UDP protocol and its application in DNS.***

[R.T.U. 2016]

OR

Explain connectionless transport protocol.

Ans. Connectionless Transport : UDP

UDP does just about as little as a transport protocol can do. Aside from the multiplexing/demultiplexing function and some light error checking, it adds nothing to IP. In fact, the application developer chooses UDP instead of TCP, then the application is almost directly talking with IP. UDP takes messages from the application process, attaches source and destination port number fields for the multiplexing/ demultiplexing service, adds two other small fields, and passes the resulting segment to the network layer. The network layer encapsulates the segment into an IP datagram and then makes a best–effort attempt to deliver the segment to the receiving host. If the segment arrives at the receiving host, UDP uses the destination port number to deliver the segment's data to correct application process. Note that with UDP there is no handshaking between sending and receiving transport–layer entities before sending a segment. For this reason, UDP is said to be connectionless.

DNS is an example of an application–layer protocol that typically uses UDP. When the DNS application in a host wants to make a query, it constructs a DNS query message and passes the message to UDP. Without performing any handshaking with the UDP entity running on the destination and system. UDP adds header fields to the message and passes the resulting segment to the network layer. The network layer encapsulates the UDP segment into a datagram and sends the datagram to a name server. The DNS application at the querying host then waits for a reply to its query. If it doesn't receive a reply (possibly because the underlying network lost the query or the reply), either it tries sending the query to another name server, or it informs the invoking application that it can't get a reply. An application developer would ever choose to build an application over UDP rather than over TCP. TCP is not always preferable, since TCP provides a reliable data transfer service. Many applications are better suited for UDP for the following reasons :

1. No Connection Establishment : TCP uses a three–way handshake before it starts to transfer data. UDP just blasts away without any formal preliminaries. Thus UDP does not introduce any delay to establish a connection. This is probably the principal reason why DNS runs over UDP rather than TCP–DNS would be much slower if it ran over TCP. HTTP uses TCP rather than UDP, since reliability is critical for Web pages with text. The TCP connection–establishment delay in HTTP is an important contributor to the "world wide web".

2. No Connection State : TCP maintains connection state in the end systems. This connection state includes receive and send buffers, congestion–control parameter and sequence and acknowledgement number parameters. This state information is needed to implement TCP's reliable data transfer service and to provide congestion control. UDP, on the other hand, does not maintain connection state and does not track any of these parameters. For this reason, a server devoted to a particular application can typically support many more active clients when the application runs over UDP rather than TCP.

3. Small Packet Header Overhead : The TCP segment has 20 bytes of header overhead in every segment, whereas UDP has only eight bytes of overhead.

4. Finer Application : Level control over what data is sent and when. Under UDP as soon an application process passes data to UDP, UDP will package the data inside a UDP segment and immediately pass the segment to the network layer. TCP on the other hand, has a congestion–control mechanism that throttles the transport layer TCP sender when one or more links between the source and destination hosts become excessively congested. TCP will also continue to send a segment of data until receipt of the segment has been acknowledged by the destination, regardless of how long reliable delivery takes.

Real–time applications often require minimum sending rate, do not want to : overly delay segment transmission and can tolerate some data loss, TCP's service model is not particularly well matched to these application's needs, these applications can use UDP and implement, as part of the application, any additional functionality that is needed beyond UDP's on–frills process–to–process service segment–delivery.

Fig. lists popular Internet applications and the transport protocols that they use. As we expect, e–mail, remote terminal access, the Web and file transfer run over TCP–all these applications need the reliable data transfer service of TCP. Nevertheless, many important applications run over UDP rather than TCP. UDP is used for RIP routing table updates (on the network layer), because the updates are sent periodically (typically every five minutes), so that last updates are replaced by more recent updates. UDP is also used to carry network management (SNMP) data.

Application layer protocol	Application–transport protocol	Underlying
Electronic mail	SMTP	TCP
Remote terminal access	Telnet	TCP
Web	HTTP	TCP

File transfer	FTP	TCP
Remote file server	NFS	typically UDP
Streaming multimedia	Proprietary	typically UDP
Internet telephony	Proprietary	typically UDP
Network management	SNMP	typically UDP
Routing protocol	RIP	typically UDP
Name translation	DNS	typically UDP

Fig. : Popular internet applications and their underlying transport protocols

UDP is preferred to TCP in this case, since network management applications must often run when the network is in a stressed state–precisely when reliable, congestion–controlled data transfer is difficult to achieve. Also, DNS runs over UDP, thereby avoiding TCP connection–establishment delays.

UDP has no congestion control. But congestion control is needed to prevent the network from entering a state in which very little useful work is done. If ever one were to start streaming high–bit–rate video without using any congestion control, there would be so much packet overflow at routers that no one would see anything. Thus, the lack of congestion control in UDP is a potentially serious problem. Many researchers have proposed new mechanisms to force all sources, including UDP sources, to perform adaptive congestion control.

UDP protocol applications in DNS : There are various key internet applications that uses UDP protocol, one of them is the Domain Name System (DNS), where queries must be fast and only consist of a single request followed by a single reply packet. To map a name onto an IP address, an application program calls a library procedure known as resolver. The resolver sends a UDP packet to a local DNS server, which searches for the name in its database. If the name is found, it returns the IP address to the resolver, which in turn informs it to the client. After having the IP address, the client then establishes a TCP connection with a destination node.

Q.12 *(a) Write a technical note on flow control and buffering.* *[R.T.U. 2016]*

OR

Discuss the mechanism of flow control.[R.T.U. 2013]

OR

Explain with the help of a diagram operation of the flow control Mechanism in TCP. *[R.T.U. 2011]*

(b) Explain the need of multiplexing at transport layer. Describe the multiplexing and De-multiplexing with help of suitable diagram. *[R.T.U. 2016]*

Ans.(a) Flow Control - Flow control is a technique for speed-matching of transmitter and receiver. Flow control ensures that a transmitting station does not overflow a receiving station with data. Flow control is a set of procedures used to restrict the amount of data that sender can send while waiting for acknowledgment. Generally, a need for flow control arises whenever there is a constraint on the communication rate between two points due to limited capacity of the communication lines or the processing hardware. Thus, a flow control scheme may be required between two users at the transport layer, between a user and an entry point of the subnet (network layer), between two nodes of the subnet (network layer), or between two gateways of an interconnected network (internet layer). Often, flow control is applied independently to individual sessions, but there is a strong interaction between its effects on different sessions because the sessions share the network's resources.

There are many approaches to flow control, including the following:

1. **Call Blocking -** Here a session is simply blocked from entering the network (its access request is denied). Such control is needed, for example, when the session requires a minimum guaranteed data rate that the network cannot provide due to limited uncommitted transmission capacity. A typical situation is that of the voice telephone network and, more generally, circuit switched networks, all of which use flow control of this type.
2. **Packet Discarding -** When a node with no available buffer space receives a packet, it has no alternative but to discard the packet. More generally, however, packets may be discarded while buffer space is still available if they belong to sessions that are using more than their fair share of some resource, are likely to cause congestion for higher-priority sessions, are likely to be discarded eventually along their path, and so on.
3. **Packet Blocking -** When a packet is discarded at some node, the network resources that were used to get the packet to that node are wasted. It is thus preferable to restrict a session's packets from entering the network if after entering they are to be discarded.
4. **Packet Scheduling -** In addition to discarding packets, a sub network node can exercise flow control by selectively expediting or delaying the transmission of the packets of various sessions.

The two main objectives of flow control are - first, strike a good compromise between throttling sessions (subject to minimum data rate requirements) and keeping average delay and buffer overflow at a reasonable level. Second, maintain fairness between sessions in providing the requisite quality of service.

Buffering - Buffering is done to cope with a speed mismatch between the producer and consumer of a data stream or to adapt between devices that have different data transfer sizes. Buffering implies the need to screen data from its final intended place so that it can be edited or otherwise processed before being moved to a regular file or database. Kernel I/O Subsystem maintains a memory area known as buffer that stores data while they are transferred between two devices with an application operation. A buffer is a data area shared by hardware devices or program processes that operate at different speeds or with different sets of priorities.

The buffer allows each device or process to operate without being held up by the other. In order for a buffer to be effective, the size of the buffer and the algorithms for moving data into and out of the buffer need to be considered by the buffer designer. Like a cache, a buffer is a "midpoint holding place" but exist not so much to accelerate the speed of an activity as to support the coordination of separate activities.

Variation in performance of loosely-coupled microprocessors and network switches and advent of a variety user-level communication protocols often create a temporary mismatch between the rates at which network messages are generated, transferred, and consumed. Buffering smooth out these rates and helps create a balanced system.

Flows can be buffered on the receiving side before being delivered. Buffering them does not affect the reliability or bandwidth and increases the delay but it smooths out the jitter.

Fig. shows the smoothing the output stream by buffering packets. A stream of packets being delivered with substantial jitter.

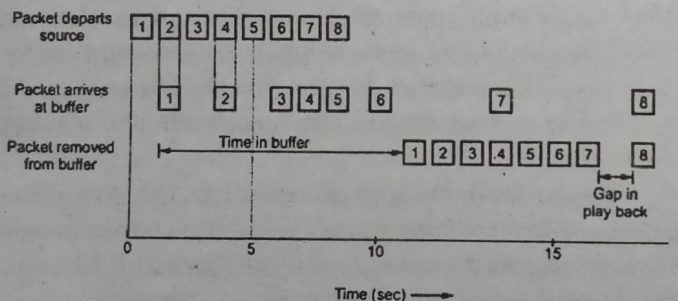


Fig. : Smoothing the output stream

- At t = 0 sec, the packet L is sent from the server and arrives at the client at t = 1 sec.
- Packet 2 undergoes more delay and takes 2 sec to arrive. As the packets arrive, they are buffered on the client machine.
- At t = 10 sec, playback begins. At this time, packets 1 through 6 have been buffered so that they can be removed from the buffer at uniform intervals for smooth play.
- But packet 8 has been delayed so much that it is not available when its play slot comes up, so playback must stop until it arrives, creating an annoying gap in the music or movie.

Flow Control and Buffering : The basic similarity between data link and transport layer flow control is that in both layers a sliding window or other scheme is needed on each connection to keep a fast transmitter from overrunning a slow receiver. The main difference is that a router has relatively few lines whereas a host may have numerous open connections.

If the network service is unreliable, the sender must buffer all TPDU's sent, just as in the data link layer. With reliable network services, other tradeoffs become possible. If the sender knows that the receiver always has buffer space, it needs not to retain TPDU's it sends. Otherwise it will have to buffer anyway, because the network layer acknowledgment only means that the TPDU has arrived, not that it was accepted.

The optimum tradeoff between source and destination buffering depends on the type of traffic. For low bandwidth, bursty traffic (like produced by an interactive terminal) it is better to buffer at the sender, using dynamically acquired buffers. For high bandwidth, smooth traffic (like a file transfer) it is better to buffer at the receiver, to allow the data to flow at maximum speed.

As connections are opened and closed and as the traffic pattern changes, the sender and receiver need to dynamically adjust their buffer allocations. Depending on the variation in TPDU size, one can opt for a chained fixed or variable size buffer or a large circular buffer per connection.

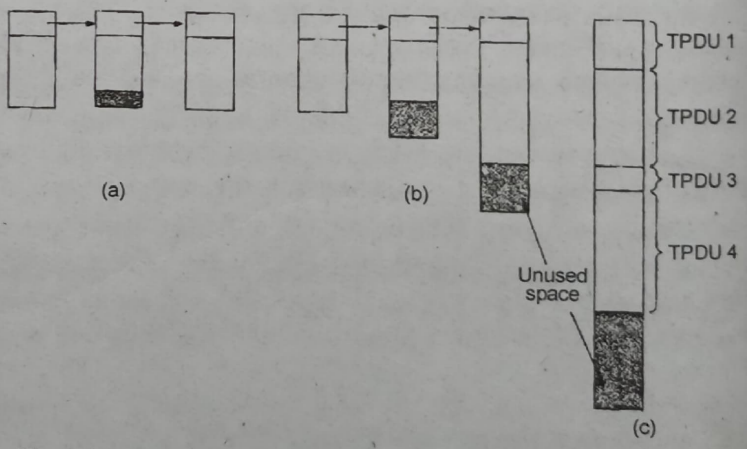


Fig. : (a) Chained fixed-size buffers, (b) Chained variable-sized buffers, (c) One large circular buffer per connection

A general way to manage dynamic buffer allocation is to decouple the buffering from the acknowledgments, in contrast to the sliding window protocol of the data link layer. Initially the sender requests a certain amount of buffer space on the receiver side, based on its perceived needs. The

receiver grants as many as it can afford. Every time the sender transmits a TPDU, it must decrement its allocation, stopping all together if it reaches 0. The receiver then separately piggybacks both ACKs and the amount of available buffer space onto the reversed traffic. Dynamic buffer management means, in effect, a variable sized window.

Ans.(b) Need of multiplexing at transport layer - The Transport layer is the link between the Application layer and the lower layer that are responsible for network transmission. This layer accepts data from different conversations and passes it down to the lower layers as manageable pieces that can be eventually multiplexed over the media. Multiplexing enables many different communications, from many different users, to be interleaved (multiplexed) on the same network, at the same time. This provides the means to both send and receive data when running multiple applications. The header is added to each segment to identify message.

Multiplexing : The addressing mechanism allows multiplexing and demultiplexing by the transport layer, as shown in Figure.

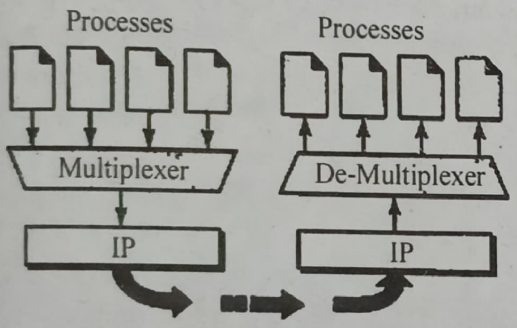


Fig. : Multiplexing and demultiplexing

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

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

The devices which perform multiplexing and ***de-multiplexing*** is called multiplexer and ***de-multiplexer*** respectively.

To make efficient use of high-speed telecommunications lines, some form of multiplexing is used. Multiplexing, allows several transmission sources to share a larger transmission capacity. The two common forms of multiplexing are frequency division multiplexing (FDM) and time division multiplexing (TDM).

Frequency division multiplexing can be used with analog signals A number of signals are carried simultaneously on the same medium by allocating to each signal a different frequency band. Modulation equipment is needed to move each signal to the required frequency band, and multiplexing equipment is needed to combine the modulated signals.

Synchronous time division multiplexing can be used with digital signals or analog signals carrying digital data- In this form of multiplexing, data from various sources are carried in repetitive frames. Each frame consists of a set of time slots and each source b assigned one or more time slots per frame. The effect is to interleave bits of data from the various sources.

Statistical time division multiplexing provides a generally more efficient service then synchronous TDM for the support of terminals. With statistical TDM. time slots are not preassigned to particular data sources. Rather, user data are buffered and transmitted as rapidly as possible using available time slots.

Multiplexing : To improve transmission efficiency, the transport layer has the option of the multiplexing.

Multiplexing at this layer occurs two ways: **upward**, meaning that multiple transport layer connections use the same network connection, or **downward**, meaning that one transport layer connection uses multiple network connections.

Types of Multiplexing

1. Upward Multiplexing : The transport layer uses virtual circuits based on the services of the lower three layers. Normally, the underlying networks charge for each virtual circuit connection. To make more cost-effective use of an established circuit, the transport layer can send several transmissions bound for the same destination along the same path by upward multiplexing. This means, if the underlying network protocol has a high throughput, for example in the range of 1 Gbps, and the user can create data only in the range of Mbps, then several users can share one network connection.

2. Downward Multiplexing : Downward multiplexing allows the transport layer to split a single connection among several different paths to improve throughput (speed of delivery). This option is useful when the underlying networks have a low or slow capacity. For example, some network layer protocols have restrictions on the sequence numbers that can be handled. X.25 uses a three-bit numbering code, so sequence numbers are restricted to the range of 0 to 7 (only eight packets may be sent before acknowledgement is required). In this case, throughput can be unacceptably low. To counteract this problem, the transport layer can opt to use more than one virtual circuit at the network layer to improve throughput. By sending several data segments at once, delivery is faster.

Q.13 ***Discuss the TCP connection establishment and release.*** ***[R.T.U. 2016]***

OR

Describe the TCP connection management. ***[R.T.U. 2013, Raj. Univ. 2008, 2007, 2006, 2005]***

OR

Does TCP use the 3-way handshaking for connection establishment and connection release? Discuss the processes used for the activities. Draw suitable diagrams. ***[R.T.U. 2012]***

OR

Explain the three way handshake protocol and justify that it successfully handles all possible issues during connection establishment in TCP. ***[R.T.U. 2011]***

Ans. TCP Connection Management : Suppose a process running in one host (client) wants to initiate a connection with another process in another host (server). The client application process first informs the client TCP that it wants to establish a connection to a process in the server. The TCP in client then proceeds to establish a TCP connection with the TCP in the server in the following :

Step 1 : The client–side TCP first sends a special TCP segment to the server–side TCP. This special segment contains no application–layer data. But one of the flag bits in the segment's header, the SYN bit, is set to 1. For this reason, this special segment is referred to as a SYN segment. In addition, the client chooses an initial sequence number (client_isn) and puts this number in the sequence number field of the initial TCP SYN segment. This segment is encapsulated within an IP datagram and sent to the server.

Step 2 : Once the IP datagram containing the TCP SYN segment arrives at the server host, the server extracts the TCP SYN segment from the datagram, allocates the TCP buffer and variable to the connection, and sends a connection–granted segment to the client TCP. This connection–granted segment contains three important pieces of information in the segment header. First, the SYN bit is set to 1. Second, the acknowledgement field of the TCP segment header is set to client_isn + 1. Finally, the server chooses its own initial sequence number (server_isn) and puts this value in the sequence number field of the TCP segment header.

Step 3 : Upon receiving the connection–granted segment, the client also allocates buffers and variables to the connection. The client host then sends the server yet another segment, this last segment acknowledges the servers connection–granted segment. The SYN bit is set to 0, since the connection is established.

Once the preceding three steps have been completed, the client and server hosts can send segments containing data to each other. In each of these future segments, the SYN bit will be set to zero. Note that in order to establish the connection, three packets are sent between the two hosts, as illustrated in figure. For this reason, this connection–establishment procedure is often referred to as a three–way handshake. It is intresting to note that a rock climber and a belayer (who is stationed below the rock climber and whose job it is to handle the climber's safety rope) use a three–way–handshake communication protocol that is identical to TCP's to ensure that both sides are ready before the climber begins ascent.

All good things must come to an end, and the same is true with a TCP connection. Further of the two processes participating in a TCP connection can end the connection. When a connection ends, the "resources" (that is, the buffers and variables) in the hosts are deal located. As an example, suppose the client decides to close the connection, as shown in figure (b). The client application process issues a close command. This causes the client TCP to send a special TCP segment to the server process.

This segment has a flag bit in the segment's header, the FIN bit (fig.(b)) set to 1. When the server receives this segment, it sends the client an acknowledgement segment in return. The server then sends its own shutdown segment, which has the FIN bit set to 1. Finally, the client acknowledges the server's shutdown segment. At this point, all the resources in the two hosts are now deallocated.

During the life of a TCP connection, the TCP protocol running in each host makes transitions through various TCP states. Fig.(c) illustrates a typical sequence of TCP states that are visited by the client TCP. The client TCP begins in the CLOSED state. The application on the client side initiates a new TCP connection. This causes TCP in the client to send a SYN segment to TCP in the server. After having sent the SYN segment, the client TCP enters the SYN_SENT state.

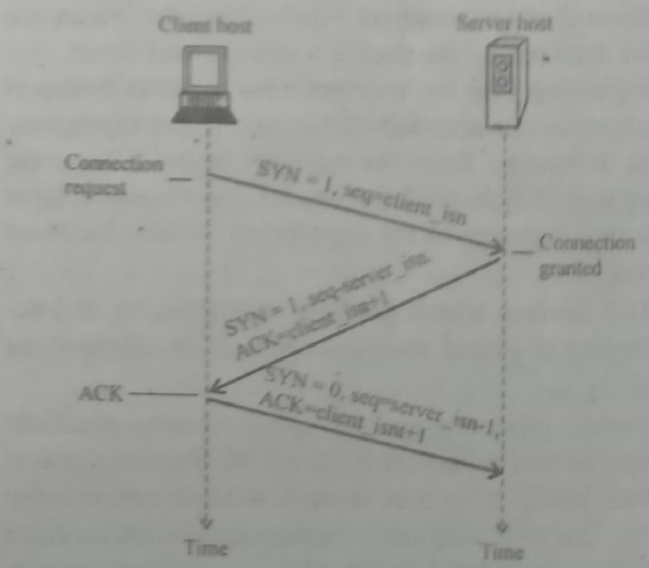


Fig. (a) : TCP three–way handshake : segment exchange

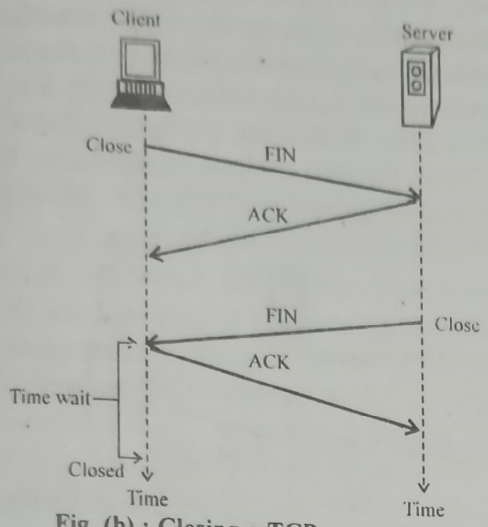


Fig. (b) : Closing a TCP connection

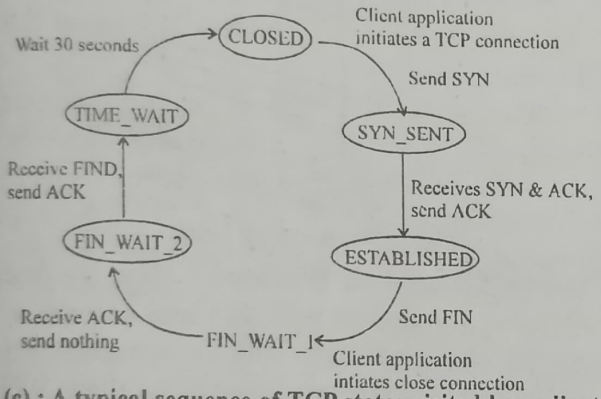


Fig. (c) : A typical sequence of TCP states visited by a client TCP

While in the SYN_SENT state, the client TCP waits for a segment from the server TCP that includes an acknowledgement for the client's previous segment and has the SYN bit set to 1. Having received such a segment, the client TCP enters the ESTABLISHED state. While in the ESTABLSHED state, the TCP client can send and receive TCP segments containing payload (that is, application–generated) data.

Suppose that the client application decides, it wants to close the connection. (Note that the server could also choose to close the connection). This causes the client TCP to send a TCP segment with the the FIN bit set to 1 and to enter the FIN_WAIT_1 state. While in the FIN_WAIT_1 state, the client TCP waits for a TCP segment from the server with an acknowledgement.

When it receives this segment, client waits for another segment from the server with the FIN bit set to 1, after receiving this segment, the client TCP acknowledges the server's segment and enters the TIME_WAIT state. The TIME_WAIT state lets the TCP client resend the final acknowledgement in case the ACK is lost. The time spent in the TIME_WAIT state is implementation–dependent, but typical values are 30 seconds, 1 minute and 2 minutes. After the wait, the connection formally closes and all resources on the client side (including port numbers) are released.

Fig. (d) illustrates the series of states typically visited by the server–side TCP, assuming the client begins connection tear–down. These two state–transition diagrams, show how a TCP connection is normally established and shut down.

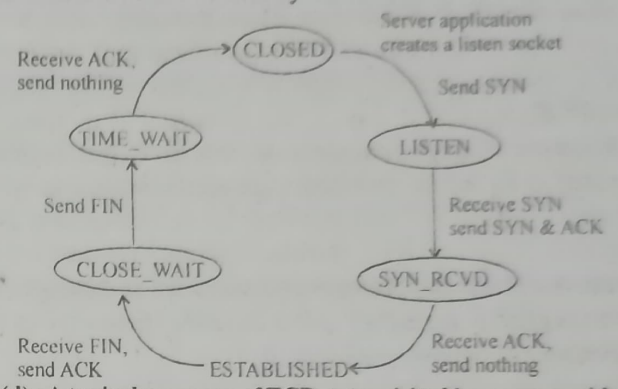


Fig. (d) : A typical sequence of TCP states visited by a server–side TCP

Q.14 ***Write a technical note on TCP congestion control.*** ***[R.T.U. 2016]***

OR

Describe the static congestion window and the dynamic congestion window. How TCP implements congestion control. ***[Raj. Univ. 2007]***

Ans. Static Congestion Windows : Although TCP uses a dynamic congestion window, it is instructive to analyze first the case of a static congestion windows. Let W, a positive integer, denote a fixed-size static congestion windows. For the static congestion windows, the server is not permitted to have more than W unacknowledged outstanding segments. When the server receives the request from the client, the server immediately sends W segments back to back to the client

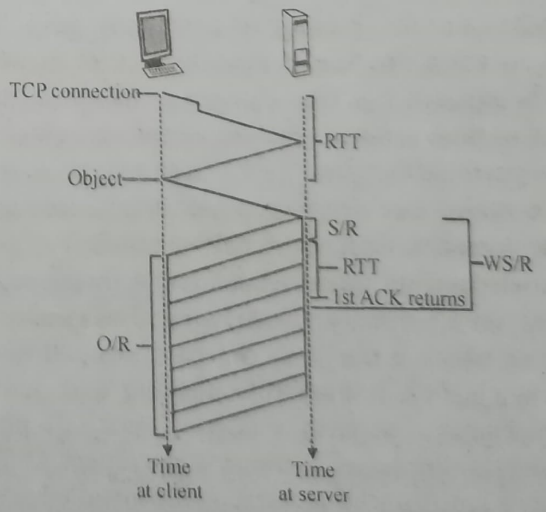


Fig. 1 : WS/R > RTT + S/R.

The server then sends one segment into the network for each acknowledgement it receives from the client. The server continue to sent one segment for each acknowledgement until all of the scantness of the object have been sent. There are two cases to consider :

1. WS/R > RTT + S/R. In this case, the server receives an acknowledgement for the first segment in the first windows before the server completes the transmission of the first windows .

2. WS/R < RTT + S/R. In this case, the server transmits the first windows's worth of segment before the server receives an acknowledgement for the first segment in the windows.

Case - 1 : Let us first consider the first case, which is illustrated in figure. In this figure the windows size is W = 4 segments. One RTT is required to intimate the TCP connection. After one RTT, the client sends a request for the object (which is piggybacked onto the third segment in the three way TCP handshake). After a total of two RTTs the client begins to receive data from server.

Segments arrive periodically from the server every S/R seconds, and the client acknowledges every segment it receives from the server. Because the server receives the first acknowledgement before it completes sending a window's a worth of scantness, the server continues to transmit segment after having transmitted the first window's worth of segments. And because the acknowledgements arrive periodically at the server every S/R seconds from the time when the first acknowledgement arrives, the server transmits scantness continuously until it has transmitted the entire object . Thus, once them server starts to transmit the object at rate R, it continues to trans mit entire object is transmitted. The latency therefore is 2 RTT + O/R.

Case - 2 : Now let us consider the second case, which is illustrated in figure 2. In this figure windows size is W = 2 segments. Once again, after a total of two RTTs the client begins to receive scantness from the server . These scantness arrive periodically every S/R seconds, and the client acknowledges every scantness it receives from the server. The server must stall and wait for an acknowledgement before resumming transmission. When an acknowledgement finally arrives, the server sends a new segment to the client. With the first acknowledgement, a window's worth of acknowledgements arrives, and each successive acknowledgement is spaced by S/R seconds. For each of these ackowledgments, the server sends exactly one segment. Thus, the server alternates between two states : a transmitting state, during which it transmits W scantness and installed state, during which it transmits nothing and waits for acknowledgement. The latency is equal to 2RTT plus the time required for the server to transmit the object, O/R, plus the amount of time that the server is in the installed state. To determine the amount of time the server is in the installed state, let K be the number of windows of data that cover the object, that is K = O/WS (if O/WS is not an integer, then round up K to the nearest integer).

Latency = 2 RTT + O/R + (K – 1) [S/R + RTT – WS/R]

Combining the two cases, we obtain

Latency = 2RTT + O/R + (K – 1) [S/R + RTT – WS/R]$^+$

Where $[x]^+ = \max(x, 0)$. Notice that the delay has three components : 2RTT to set up the TCP connection and to request and begin to receive the object, O/R, the item for the server to transmit the object, and a final term (K – 1) [S/R + RTT – WS / R]$^+$ for the amount of time the server installs.

This completes our analysis of static windows. The following analysis for dynamic windows is more complicated but parallels that for static windows.

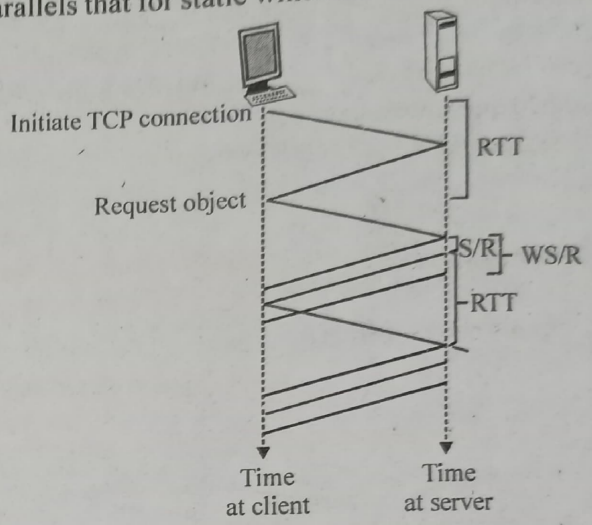


Fig. 2: WS/R < RTT + S/R

Dynamic Congestion Windows : Here, we describe TCP's dynamic congestion windows into account in the latency model. The server starts with a congestion windows of one segment and sends one segment to the client. When it receives an acknowledgement for the segment, it increases its congestion windows to two segments and sends two segments to the client (spaced apart by S/R seconds). As it receives the acknowledgements and sends four scantness to the client (again spaced apart by S/R seconds.) The process continues, with the congestion windows doubling every RTT. A timing diagram for TCP is illustrated in fig.

Note that O/S is the number of segments in the object in figure 3O/S = 15. Consider the number of segments that are in each of the windows. The first window contain one segment, the second windows contains two segments, and the third window contains four segments. More generally, the k^{th} windows contains 2k–1 scantness. Let K be the number of windows that cover the object, in the preceding diagram, K = 4. In general, we can express K in terms of O/S as follows :

$$K = \min\left\{k; 2^0 + 2^{10} + ... + 2^{k-1} \geq \frac{O}{S}\right\}$$

$$= \min\left\{k; 2^k - 1 \geq \frac{O}{S}\right\}$$

$$= \min\left\{k; k \geq \log_2\left(\frac{O}{S} + 1\right)\right\} = \left[\log_2\left(\frac{O}{S} + 1\right)\right]$$

After transmitting a window's worth of data, the server may install (that, is, stop transmitting) while it waits for an acknowledgement. In figure the server installs after transmitting the first and second windows but not after transmitting the third . Let us now calculate the amount of install time after transmitting the k^{th} window. From the time the server begins to transmit the k^{th} window until the time when the server receives an acknowledgement for the first segment in the windows is S/R + RTT. The transmission time of the k^{th} window is $(S/R)2^{k-1}$. The install time is the difference of these two quantities, that is :

$$[S/R + RTT - 2^{k-1}(S/R)]^+$$

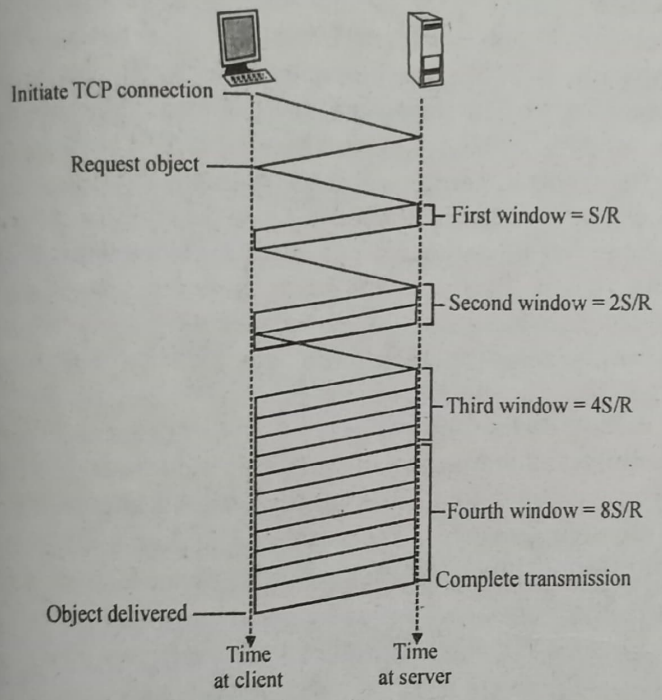


Fig. 3: TCP timing during slow start

The server can potentially stall after the transmission of each of the first k – 1 windows. (The server is done after the transmission of the k^{th} window). We can 2RTT for setting up the TCP connection and requesting the file, O/R, the transmission time of the object, and the sum of all the installed times. Thus

$$\text{Latency} = 2RTT + \frac{O}{R} + \sum_{K=1}^{k-1}\left[\frac{S}{R} + RTT - 2^{k-1}\frac{S}{R}\right]^+$$

Compare this equation with the latency equation for static congestion windows, all the terms are exactly the same except that the term WS/R for static windows has been replaced by 2^{k-1} (S/R) for dynamic windows. To obtain a more compact expression for the latency, let Q be the number of times the server would stall if the object contained an infinite number of segments. Paralleling a derivation similar to that for K, we obtain

$$Q = \left[\log_2\left(1 + \frac{RTT}{S/R}\right)\right] + 1$$

The actual number of times that the server stalls is P = min {Q, K – 1}. In figure P = Q = 2. Combining the equations gives the following closed form expression for the latency.

$$\text{Latency} = 2RTT + \frac{O}{R} + P = [RTT + \frac{S}{R}] - (2^P - 1)\frac{S}{R}$$

Q.15 ***Discuss the TCP header and segment structure in detail.***

OR

Draw and explain TCP Header and segment structure.

[R.T.U. 2016]

Ans. TCP Segment Header : Figure shows the layout of a TCP segment. Every segment begins with a fixed-format, 20 byte header. The fixed header may be followed by header options. After the options, if any, up to 65,535 – 20 – 20 = 65,495 data bytes may follow, where the first 20 refer to the IP header and the second to the TCP header. Segments without any data are legal and are commonly used for acknowledgements and control messages.

Here, TCP header is described field by field. The *Source port* and *Destination port* field identify the local end points of the connection. A port plus its host's IP address forms a 48-bit unique end point. The source and destination end points together identify the connection.

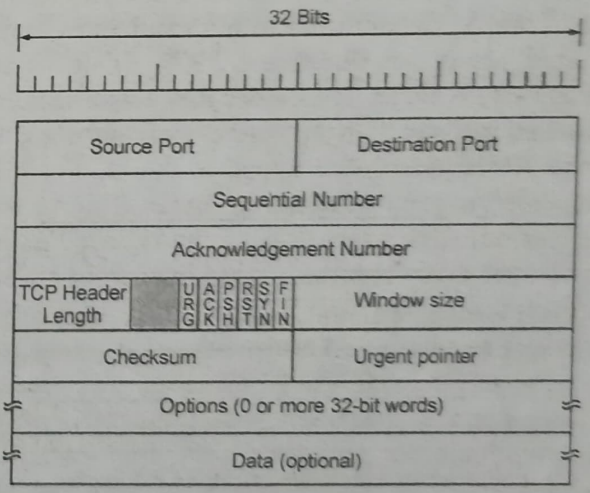


Fig. : TCP Header

The *Sequence number* and *Acknowledgement number* fields perform their usual functions. Note that the latter specifies the next byte expected, not the last byte correctly received. Both are 32 bits long because every byte of data is numbered in a TCP stream.

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, so the header is, too. Technically, this field really indicates the start of the data within the segment, measurement in 32-bit words, but that number is just the header length in words, so the effect is the same.

Next comes a 6-bit field that is not used. The fact that

this field has survived intact for over a quarter of a century is testimony to how well though out TCP is Lesser protocols would have needed it to fix bugs in the original design.

Now come six 1-bit flags. *URG* is set to 1 if the *Urgent* pointer is in use. The Urgent pointer is used to indicate a byte offset from the current sequence number at which urgent data are to be found. This facility is in lieu of interrupt messages. This facility is a bare-bones way of allowing the sender to signal the receiver without getting TCP itself involved in the reason for the interrupt.

The ACK bit is set to 1 to indicate that the *Acknowledgement number* is valid. If ACK is 0, the segment does not contain an acknowledgement so the Acknowledgement number field is ignored.

The *PSH* bit indicates *PUSHed* data. The receiver is hereby kindly requested to deliver the data to the application upon arrival and not buffer it untill a full buffer has been received (which it might otherwise do for efficiency).

The *RST* bit used to reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection.

The *SYN* bit is used to establish connections. The connection request has SYN = 1 and ACK = 0 to indicate that the piggyback acknowledgement field is not in use. The connection reply does bear an acknowledgement, so it has SYN = 1 and ACK = 1. In essence the SYN bit is used to denote CONNECTION REQUEST and CONNECTION ACCEPTED, with the ACK bit used to distinguish between those two possibilities.

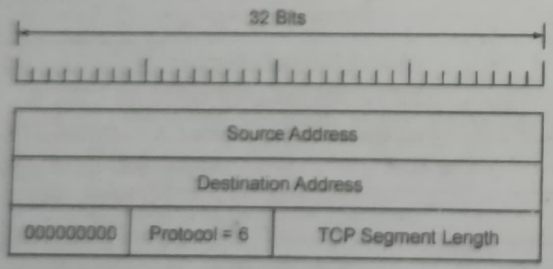


Fig. Pseudoheader Included in the TCP Checksum

The *FIN* bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, the closing process may continue to receive data indefinitely. Both SYN and FIN segments have sequence number and are thus guaranteed to be processed in the correct order. Flow control in TCP is handled using a variable-sized sliding window. The Window size field tells how many bytes may be sent starting at the byte acknowledged. A Window size field of 0 is legal and says that the bytes upto and including Acknowledgement number – 1 have been received, but that the receiver is currently badly in need of a rest and would like no more data for the moment, thank you. The receiver can later grant permission to send by transmitting a segment with the same Acknowledgement number and a nonzero Window size field.

A *Checksum* is also provided for extra reliability. It checksums the header, the data and the conceptual pseudoheader shown in Fig. When performing this computation, the TCP Checksum field is set to zero and the data field is padded out with an additional zero byte if its length is an odd number. The checksum algorithm is simply to add up all the 16-bit words in one's complement and then to take the one's complement of the sum. As a consequence, when the receiver performs the calculation on the entire segment, including the Checksum field, the result should be 0.

The pseudoheader contains the 32-bit IP addresses of the source and destination machines, the protocol number for TCP (6) and the byte count for the TCP segment (including the header). Including the pseudoheader in the TCP checksum compuatation helps detect misdelivered packets, but including it also violates the protocol hierarchy since the IP addresses in it belong to the IP layer, not to the TCP layer. UDP uses the same pseudoheader for its checksum.

The *Options field* provides a way to add extra facilities not covered by the regular header. The most important option is the one that allows each host to specify the maximum TCP payload it is willing to accept. Using large segments is more efficient than using small ones because the 20-byte header can then be amortized over more data, but small hosts may not be able to handle big segments.

During connection setup, each side can announce its maximum and see its partner's. If a host does not use this option, it defaults to a 536-byte payload. All Internet hosts are required to accept TCP segments of 536 + 20 = 556 bytes. The maximum segment size in the two directions need not be the same.

For lines with high bandwidth, highly delay, or both, the 64-KB window is often a problem. On a T3 line (44.736 Mbps), it takes only 12 msec to output a full 64-KB window. If the round-trip propagation delay is 50 msec (which is typical for a transcontinental fiber), the sender will be idle 3/4 of the time waiting for acknowledgements. On a satellite connection, the situation is even worse. A large window size would allow the sender to keep pumping data out, but using the 16-bit Window size field, there is no way to express such a size. In RFC 1323, a Window scale option was proposed, allowing the sender and receiver to negotiate a window scale factor. This number allows both sides to shift the Window size field up to 14 bits to the left, thus allowing windows of up 2^{30} bytes. Most TCP implementations now support this option.

Another option proposed by RFC 1106 and how widely implemented is the use of the selective repeat instead of go back n protocol. If the receiver gets one bad segment and then a large number of good ones, the normal TCP protocol will eventually time out and retransmit all the unacknowledged segments, including all those that were received correctly (i.e., the go back n protocol). RFC 1106 introduced NACKs to allow the receiver to ask for a specific segment (or segments). After it gets these, it can acknowledge all the buffered data, thus reducing the amount of data retransmitted.

Q.16 ***Describe the differences between a confirmed service and an unconfirmed service. Do the following functions fall into the category of confirmed service, unconfirmed service, both types or neither:***

(i) Connection establishment

(ii) Data transfer in a connection oriented service

(iii) Data transfer in a connectionless service

(iv) Connection release

Justify your answer. *[R.T.U. 2015, Raj. Univ. 2008]*

Ans. Service can be either confirmed or unconfirmed. In a confirmed service, there is a request, an indication, a response and a confirm. In an unconfirmed service, there is just a request and an indication. CONNECT is always a confirmed service because the remote peer must agree to establish a connection. Data transfer, on the other hand, can be either confirmed or unconfirmed, depending on whether or not the sender needs an acknowledgement. Both kinds of services are used in networks. To make the concept of a service, let us consider as an example a simple connection oriented service with eight service primitives as follows :

1. **CONNECT. request :** Request a connection to be established
2. **CONNECT. indication :** Signal the called parity
3. **CONNECT. response :** Used by the caller to accept/reject calls
4. **CONNECT. confirm :** Tell the caller whether the call was accepted
5. **DATA. request :** Request that data be sent
6. **DATA. indication :** Signal the arrival of data
7. **DISCONNECT. request :** Request that a connection be released
8. **DISCONNECT. indication :** Signal the peer about the request

In this example, CONNECT is a confirmed service (an explicit response is required), whereas DISCONNECT is unconfirmed (no response).

From the given point :

(i) Connect establishment : Confirmed service

(ii) Data transfer in a connection oriented service : Both types

(iii) Data transfer in a connection less service : Neither

(iv) Connection release : Unconfirmed service

Q.17 ***Explain the TCP service model.*** *[R.T.U. 2015]*

Ans. TCP Service Model : TCP service is obtained by both the sender and receiver creating end points, called sockets. Each socket has a socket number (address) consisting of the IP address of the host and a 16-bit number local to that host, called a port. A port is the TCP name for a TSAP. For TCP service to be obtained, a connection must be explicitly established between a socket on the sending machine and a socket on the receiving machine.

A socket may be used for multiple connections at the time. In other words, two or more connections may terminate at the same socket. Connections are identified by the socket identifiers at both ends, that is, (socket1, socket2). No virtual circuit numbers or other identifiers are used.

Port numbers below 1024 are called well-known ports and are reserved for standard services. For example, any process wishing to establish a connection to a host to transfer a file using FTP can connect to the destination host's port 21 to contact its FTP daemon.

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

Port	Protocol	Use
21	FTP	File Transfer
23	Telnet	Remote Login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup Information about a User
80	HTTP	World Wide Web
110	POP-3	Remote E-mail Access
119	NNTP	USENET News

Fig. : Some assigned ports

All TCP connections are full duplex and point-to-point. Full duplex means that traffic can go in both directions at the same time. Point-to-point means that each connections has exactly two end points. TCP does not support multicasting or broadcasting.

A TCP connection is a byte stream, not a message stream. Message boundaries are not preserved end to end. For example, if the sending process does four 512-byte writes to a TCP stream, these data may be delivered to the receiving process as four 512-byte chunks, two 1024-byte chunks, one 2048-byte chunk or some other way. There is no way for the receiver to detect the unit(s) in which the data were written.

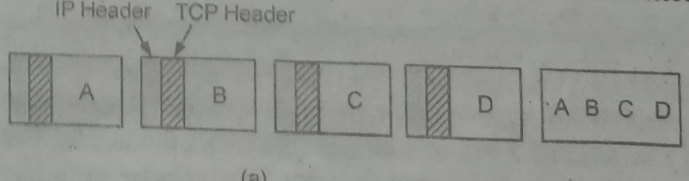


Fig. : (a) Four 512-byte segments sent as separate IP datagrams (b) The 2048 bytes of data delivered to the application in a single READ call

Files in UNIX have this property too. The reader of a file cannot tell whether the files was written a block at a time, a byte at a time, or all in one blow. As with a UNIX file, the TCP software has no idea of what the bytes mean and no interest in finding out. A byte is just a byte.

When an application passes data to TCP, TCP may send it immediately or buffer it (in order to collect a larger amount to send at once), at its discretion. However, sometimes, the application really wants the data to be sent immediately. For example, suppose a user is logged in to a remote machine. After a command line has been-finished and the carriage return types, it is essential that the line be shipped off to the remote machine immediately and not buffered until the next line comes in. To force data out, applications can use the PUSH flag, which tells TCP not to delay the transmission.

Some early applications used the PUSH flag as a kind of marker to delineate message boundaries. While this trick sometimes works, it sometimes fails since not all implementations of TCP pass the PUSH flag to the application on the receiving side. Furthermore, if additional PUSHes come in before the first one has been transmitted (e.g. because the output line is busy), TCP is free to collect all the PUSHed data into a single IP datagram, with no separation between the various pieces.

One last feature of the TCP service is *urgent data*. When an interactive user hits the DEL or CTRL-C key to break off a remote computation that has already begun, the sending application puts some control information in the data stream and gives it to TCP along the URGENT flag. This event causes TCP to stop accumulating data and transmit everything it has for that connection immediately.

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

Q.18 ***Explain the significance of following control bits in TCP :***

(i) SYN

(ii) ACK

(iii) RST

(iv) FIN ***[R.T.U. 2014]***

Ans.(a) (i)SYN : The SYN bit is used ot establish connection. The connection request has SYN = 1 and ACK = 0 to indicate that the piggyback acknowledgement field is not in use. The connection reply does bear an acknowledgement, so it has SYN = 1 and ACK = 1. In essence the SYN bit is used to denote CONNECTION REQUEST and CONNECTION ACCEPTED, with the ACK bit used to distinguish between those two possibilities.

(ii) ACK : These are a little trickier than sequence numbers. TCP is full-duplex, so that Host A may be receiving data from Host B while it sends data to Host B (as part of the same TCP connection). Each of the segments that arrive from Host B has a sequence number for the data flowing from B to A. The acknowledgement number that Host A puts in its segment is the sequence number of the next byte Host A is expecting from Host B. Suppose the Host A has received all bytes numbered 0 through 535 from B and suppose that it is about to send a segment to Host B. Host A is waiting for byte 536 and all the subsequent bytes in Host B's data stream. So Host A puts 536 in the acknowledgement number field of the segment i sends to B.

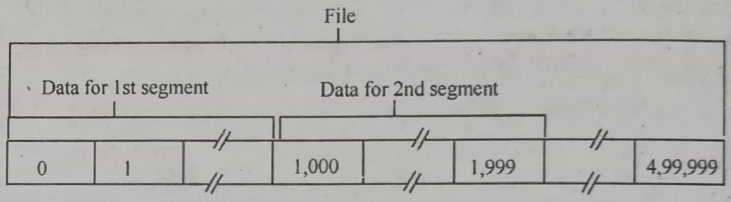


Fig. Dividing file data into TCP segments

As another example, suppose that Host A has received one segment from Host B containing bytes 0 through 535 and another segment containing bytes 900 through 1,000. For some reason Host A has not yet received bytes 536 through 899. In this example, Host A is still waiting for byte 536 (and beyond) in order to recreate B's data stream. Thus, A's next segment to B will contain 536 in the acknowledgement number field. Because TCP only acknowledges bytes up to the first missing byte in the stream, TCP is said to provide **cumulative acknowledgements.**

(iii) RST : Reset the connection. The RST bit is used to RESET the TCP connection due to unrecoverable errors. When an RST is received in a TCP segment, the receiver must respond by immediately terminating the connection. A RESET causes both sides immediately to release the connection and all its resources. As a result, transfer of data ceases in both directions, which can result in loss of data that is in transit. A TCP RST indicates an abnormal termination of the connection.

(iv) FIN: No more data from the sender. Receiving a TCP segment with the FIN flag does not mean that transferring data in the opposite direction is not possible. Because TCP is a fully duplex connection, the FIN flag will cause the closing of connection only in one direction. To close a TCP connection gracefully, applications use the FIN flag.

Q.19 ***(a) In Leaky bucket algorithm, should one packet independent of size of packet be allowed or constant amount of data be allowed to flow? Discuss.*** ***[R.T.U. 2012]***

OR

Write notes on the Leaky Bucket Algorithm. *[Raj. Univ. 2004]*

(b) Imagine a flow specification that has a maximum packet size of 10 million bytes/sec, a token bucket size of 1 million bytes and a maximum transmission rate of 50 million bytes/sec. How long can a burst at maximum speed last ?

(c) An ATM network uses a token bucket scheme for traffic shaping. A new token is put into bucket every 5 μs. Each token is good for one cell of 53 bytes. What is the maximum sustainable data rate ?

(d) For a hierarchical routing with 4500 routers, compute the size of cluster and region to minimize the routing table entries. Assume a maximum number of 3 levels.

[R.T.U. 2012]

Ans. (a) Leaky Bucket Algorithm

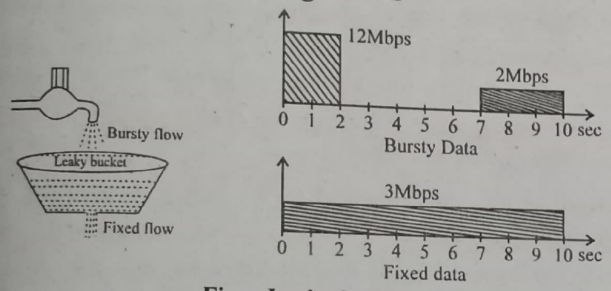


Fig. : Leaky bucket

If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket–unless the bucket is empty. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called leaky bucket can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent out at an average rate. The figure shows a leaky bucket and its effects.

In the figure, we assume that the network has committed a bandwidth of 3Mbps for a host. The use of leaky bucket shapes the input traffic to make it conform to this commitment. In this figure the host sends a burst of data at a rate of 12 Mbps for 2 sec., for a total of 24 megabits of data. The host is silent for 5 second and then sends data at a rate of 2 Mbps for 3 sec, for a total of 6 megabits of data. In all, the host has sent 30 megabits of data in 10 seconds.

The leaky bucket smooths the traffic by sending out data at a rate of 3Mbps during the same 10 second. Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth then is sent aside for this host. The leaky bucket may prevent congestion. As an analogy, consider the free way during rush hour (bursty traffic). If, instead, commuters could stagger their working hours, congestion on our free ways could be avoided.

Leaky Bucket Implementation

A simple leaky bucket implementation is shown in fig. FIFO queue holds the packets. If the traffic consists of fixed size packets (e.g. cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consists of variable length packets, the fixed output rate must be based on the number of bytes or bits.

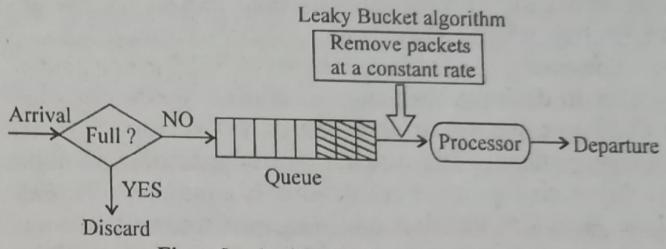


Fig. : Leaky bucket implementation

The following is an algorithm for variable–length packets :

(1) Initialize a counter to n at the tick of the clock.

(2) In n is greater than size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until n is smaller than the packet size.

(3) Reset the counter and go to step 1.

Ans. (b) Given token bucket capacity C = 1MB/sec.

Maximum output rate M = 50MB/sec.

Packet size = 10 MB/sec.

Then maximum speed S is given by

$$S = C/(M-P)$$

$$= \frac{1}{50-10} = \frac{1}{40}$$

$$= 0.025 \text{ msec.}$$

Ans. (c) The maximum sustainable data rate is given by

$$C + PS = MS$$

Where, C = token bucket capacity

P = token arrival rate

S = burst length = 5μs (given)

M = maximum output rate = 53 bytes

Hence maximum sustainable data rate is

$$5 \times 53 = 265 \text{ bytes/sec.}$$

Ans. (d) Given 4500 routers. If the subnet is partitioned into 50 regions of 90 router each. Each router needs 90 local entries plus 49 remote entries for a total of entries. If a three-level hierarchy is chosen, with eight clusters, each containing 9 regions of 60 routers, each router needs 10 entries for local routers, 8 entries for routing to other regions within its own clusters, for a total of 51 entries.

□□□

APPLICATION LAYER 5

PREVIOUS YEARS QUESTIONS

PART-A

Q.1 ***What are the three FTP transmission modes?*** *[R.T.U. 2019]*

Ans. For transforming files across the internet connection FTP uses three transmission modes :

1. Stream Mode
2. Block Mode
3. Compressed Mode

Q.2 ***What are the two main categories of DNS messages?*** *[R.T.U. 2019]*

Ans. DNS Message : DNS has two types of messages : *query* and *response*. Both types have the same format. The query message consists of a *header* and *question records*; the response message consists of a *header*, *question records*, *answer records*, *authoritative records*, and *additional records* (see fig.).

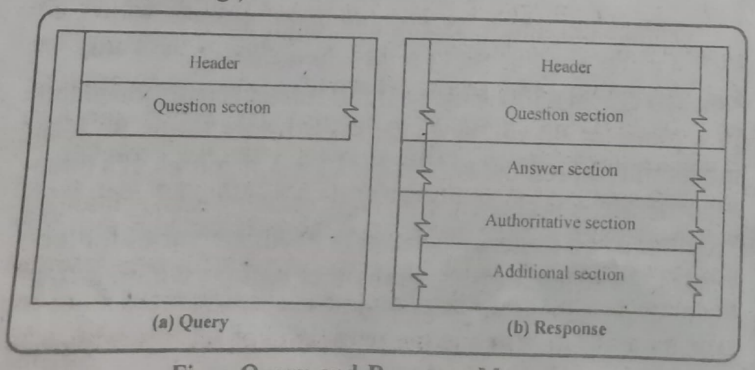


Fig. : Query and Response Messages

Q.3 ***Explain authoritative DNS.*** *[R.T.U. 2019]*

Ans. Authoritative DNS : An authoritative name server is a name server that gives answers that have been configured by an original source, for example, the domain administrator or by dynamic DNS methods, in contrast to answers that were obtained via a regular DNS query to another name server. An authoritativeonly name server only returns answers to queries about domain names that have been specifically configured by the administrator .An authoritative name server can either be a master server or a slave server. A master server is a server that stores the original (master) copies of all zone records. A slave server uses an automatic updating mechanism of the DNS protocol in communication with its master to maintain an identical copy of the master records.

Q.4 ***Explain Non-Authoritative DNS.*** *[R.T.U. 2019]*

Ans. Non-Authoritative DNS

Non-authoritative name servers do not contain copies of any domains. Instead they have a cache file that is constructed from all the DNS lookups it has performed in the past for which it has gotten an authoritative response. When a non-authoritative server queries an authoritative server and receives an authoritative answer, it passes that answer along to the querier as an authoritative answer. Thus, non-authoritative servers can answer authoritatively for a given resolution request. However, non-authoritative servers are not authoritative for any domain they do not contain specific zone files for. Most often, a non-authoritative server answers with a previous lookup from its lookup cache. Any answer retrieved from the cache of any server is deemed non-authoritative because it did not come from an authoritative server.

Q.5 ***Explain different services of application layer.***

Ans. Application layer provides the following services :

(i) Less time consumption *i.e.* fast delivery
(ii) Reliable data transfer
(iii) Less amount of congestion throughout the network
(iv) Safety of data in context of threads due to intruders etc.

Q.6 ***Explain use of cookies in WWW and HTTP.***

Ans. Use of Cookies in WWW : When a client sends a request to a server, the browser looks in the cookie directory to see if it can find a cookie sent by that server. If found, the cookie is included in the request. When the server receives the request, it knows that this is an old client, not a new one. Note that the contents of the cookie are never read by the browser or disclosed to the user. It is a cookie *made* by the server and *read* by the server.

Use of Cookies in HTTP : An HTTP server is stateless. This simplifies server design and has permitted engineers to develop high-performance web servers that can handle thousands of simultaneous TCP connections. It is often desirable for a website to identify users, either because the server wishes to restrict user access or because it wants to serve content as function of user identity. For these purposes. HTTP uses cookies. Cookies allows sites to keep track of users.

Q.7 ***Write the components of cookie technology.***

Ans. Cookie Technology has four components :

(i) A cookie header line in HTTP response message
(ii) A cookie header line in HTTP request message
(iii) A cookie file kept on user's and system and managed by the user's browser.
(iv) A back-end data base at the website.

PART-B

Q.8 ***(a) What is Proxy server and how it is related to HTTP.***
(b) What is URL and what are its components? Explain.
(c) In electronic mail, what is MIME? *[R.T.U. 2017]*

Ans. (a) Proxy Server: A proxy server is a server (a computer system or an application) that acts as an intermediary for requests from client seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity. Proxies were invented to add structure and encapsulation to distributed systems. Today, most proxies are web proxies, facilitating access to content on the World Wide Web, providing anonymity and may be used to bypass IP address blocking.

Related to HTTP : HTTPS is Hypertext Transfer Protocol over Security Socket Layer (HTTP over SSL). It's a more safety protocol than HTTP. In view of security requirements, more and more sites transfer from HTTP to HTTPs, especially bank sites and online paying sites. As a default setting, HTTPS works upon port 443.

Proxy server handles HTTPS requests from clients is always called HTTPS proxy server. It's similar with HTTP proxy server, the only difference is the protocols they focus on. No matter HTTP or HTTPS proxy server, they both can carry out caching of information downloaded from the Internet. This is a very useful function which can speed up surfing and reduce network traffic.

Most proxy servers act both as an HTTP proxy server and as an HTTPS proxy server. Proxy settings on clients for both HTTP and HTTPS are similar, the only thing you need to care is HTTPS is mostly identified by "Secure" or "Seccurity".

Ans. (b) Uniform Resource Locator (URL) : A client that wants to access a Web page needs the address. To facilitate the access of documents distributed throughout the world, HTTP uses locators. The uniform resource locator (URL) is a standard for specifying any kind of information on the Internet. The URL defines four things: protocol, host computer, port and path as shown as shown in fig.

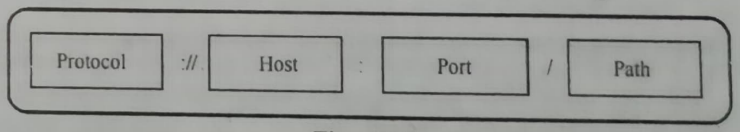


Fig. : URL

The **protocol** is the client/server program used to retrieve the document. Many different protocols can retrieve a document; among them is FTP or HTTP. The most common today is HTTP. The **host** is the computer on which the information is located, although the name of the computer can be an alias. Web pages are usually stored in computers, and computers are given alias names that usually begin with the characters "www". This is not mandatory, however, as the host can be any name given to the computer that hosts the Web page. The URL can optionally contain the **port** number of the server. If the port is included, it is inserted between the host and the path, and it is separated from the host by a colon. **Path** is the pathname of the file where the information is located. The path contains slashes.

Ans.(c) MIME (Multipurpose Internet Mail Extension)

MIME describes a "globally recognized set of data type" that is recognized by all servers and all browsers anywhere on the web. MIME was originally established to make it possible for systems to "interchange multimedia mail", and the standard was simply incorporated by the designers of the web.

The MIME specification includes the following elements :

1. File new message header fields. These fields provide information about the body of the message.

2. A number of content formats.

3. Transfer Encoding.

The five header fields defined in MIME are :

- **MIME Version :** Must have parameter value 1.0.
- **Content–Types :** Describes the data contained in the body with sufficient details.
- **Content Transfer Encoding :** Indicates the type of transformation that has been used to represent the body of the message.
- **Content–ID :** Used to identify MIME entities uniquely in multiple context.
- **Content–Description:** A text description of the object with in the body.

MIME Content–Type : The MIME Content–type consists of two parts :

(i) General Category of the object to be managed (text, video, audio etc).

(ii) Specific Type (lotus, Postscript etc).

MIME Transfer Encoding : The objective is to provide reliable delivery across the largest range of environment. The MIME Standard defines two methods of encoding data.

(a) Quoted–Printable, (b) base 64 transfer encoding

Q.9 ***In DNS, can a single host have (i) multiple hostnames and (ii) multiple addresses? How the records are organized in such cases?*** *[R.T.U. 2014]*

Ans. (i) Host names are the short names or nicknames that corresponds to an address in a network. DNS (Domain Name Server) is a naming system that provides mapping of IP address of a single host to hostnames.

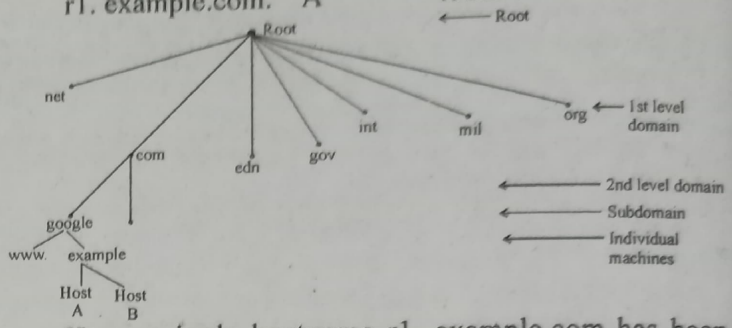
A single host computer can have many hostnames in network.

(ii) A single host can have multiple addresses by providing a hostname DNS server directs is to a set of multiple addresses assigned to that hostname so that user can choose the one to which he wants to connect.

DNS is organized in a hierarchically that moves from top level domains to specific hostnames. The top leves represents the root of DNS and it is extended by 1st level domain and 2nd level domain. The domains can be further classified to subdomains and individual machines as shown in the figure.

Similarly using this hierarchical structure, more than one IP address can be assigned to one host name as

r1. example.com.	A	192.2.2.10
r1. example.com.	A	192.2.2.12
r1. example.com.	A	192.2.2.15



Here a single hostname r1. example.com has been assigned three IP addresses namelly 192.2.2.10, 192.2.2.12 and 192.2.2.15.

In order to assign multiple hostnames for same IP address as

r1. example.com.	C NAME	r2.example.com.
r2. example.com.	A	192.0.2.15

From here the CName is used to assign the IP address (192.0.2.15) related to r1.example.com as other hostname in form of r2. example.com.

Multiple hostnames and IP addresses complicates the process of resolving IP addresses. So, they should be used with care.

Q.10 ***(a) What is the role of cookies in World Wide Web?*** *[R.T.U. 2014]*

(b) Write short notes on the Performance enhancement in WWW. *[R.T.U. 2014,12]*

Ans. (a) The World Wide Web was originally designed as a stateless entity. A client sends a request; a server responds. Their relationship is over. The original design of WWW, retrieving publicly available documents, exactly fits this purpose. Today the Web has other functions; some are listed here.

1. Some websites need to allow access to registered clients only.
2. Websites are being used as electronic stores that allow users to browse through the store, select wanted items, put them in an electronic cart, and pay at the end with a credit card.
3. Some websites are used as portals: the user selects the Web pages he wants to see.
4. Some websites are just advertising.

For these purposes, the cookie mechanism was devised.

Cookies : A **cookie**, also known as a **web cookie, browser cookie,** and **HTTP cookie,** is a piece of text stored on a user's computer by their web browser. A cookie can be used for authentication, storing site preferences, shopping cart contents, the identifier for a server-based session, or anything else that can be accomplished through storing text data.

A cookie consists of one or more name-value pairs containing bits of information, which may be encrypted for information privacy and data security purposes. The cookie is sent as a field in the header of the HTTP response by a web server to a web browser and then sent back unchanged by the browser each time it accesses that server.

Cookies may be set by the server with or without an expiration date. Cookies without an expiration date exist until the browser terminates, while cookies with an expiration date may be stored by the browser until the expiration date passes. Users may also manually delete cookies in order to save space or to avoid privacy issues.

Creation and Storage of Cookies

1. When a server receives a request from a client; it stores information about the client in a file or a string. The information may include the domain name of the client, the contents of the cookie (information the server has gathered about the client such as name, registration number, and so on), a timestamp, and other information depending on the implementation.
2. The server includes the cookie in the response that it sends to the client.
3. When the client receives the response, the browser stores the cookie in the cookie directory, which is sorted by the domain server name.

Using Cookies : *Refer to Q.6.*

Ans. (b) Performance enhancement in WWW : WWW stands for "World Wide Web." and it is not a synonym for the Internet. The World Wide Web, or just "the Web," as ordinary people call it, is a subset of the Internet. The Web consists of pages that can be accessed using a Web browser. The Internet is the actual network of networks where all the information resides. Things like Telnet, FTP, Internet gaming, Internet Relay Chat (IRC), and e-mail are all part of the Internet, but are not part of the World Wide Web. The Hyper-Text Transfer Protocol (HTTP) is the method used to transfer Web pages to your computer. With hypertext, a word or phrase can contain a link to another Web site. All Web pages are written in the hyper-text markup language (HTML), which works in conjunction with HTTP.

Q.11 ***What E-mail privacy? Why do we need POP3 or IMAP4 for electronic mail.*** ***[R.T.U. 2013]***

Ans. The protection of email from unauthorized access and inspection is known as electronic privacy. In countries with a constitutional guarantee of the secrecy of correspondence, email is equated with letters and thus legally protected from all forms of eavesdropping.

The requirement refers generally to both the user's expectation that something is private and also to society's expectation that the thing should be private.

There are three situations in which privacy of e-mail could be a concern.

1. Interception during transmission, for example, by a wiretap on the telephone line at the sender's building.

2. Reading during storage on the destination computer. For example, if one sends e-mail to lstudent@fplc.edu, this e-mail is stored on a hard drive of a computer at fplc until the recipient deletes it. If the recipient does not read the message within a reasonable time, typically a few months of sending the message, the system operator may delete the message to recover space on the hard drive for other users. Good operating practice of a computer system involves making routine backup copies (typically on magnetic tape) of all files on the hard drive, since hard drives can fail. An e-mail may be retrieved from a backup tape even after that e-mail was deleted from the hard drive by the recipient.

3. Disclosure of contents by the recipient.

The actual mail transfer is done through Message Transfer Agents (MTA). To send mail, a system must have the client MTA, and to reveive mail, a system must have a server MTA. The actual formal protocol that defines the MTA client and server in the Internet is called the Simple Mail Protocol (SMTP).

SMTP is used two times, between the sender and the sender's mail server and between the two mail servers. SMTP simply defines how commands and responses must be sent back and forth (forward).

SMTP uses commands and responses to transfer messages between an MTA client and an MTA server as shown in figure 1.

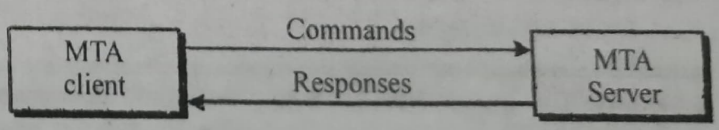


Fig. 1 : Commands and responses

SMTP expects the destination host, the mail server receiving the mail, to be on-line all the time, otherwise, a TCP connection cannot be established. For this reason, it is not practical to establish an SMTP session with a desktop computer because desktop computers are usually powered down at the end of the day.

In many organizations, mail is received by an SMTP server that is always on-line. This SMTP server provides a mail-drop service. The server receives the mail on behalf of every host in the organization. Workstations interact with the SMTP host to retrieve messages by using a client-server protocol such as **Post Office Protocol (POP)**, version 3 (POP3).

Although POP3 is used to download messages from the server, the SMTP client is still needed on the desktop to forward messages from the workstation user to its SMTP mail server.

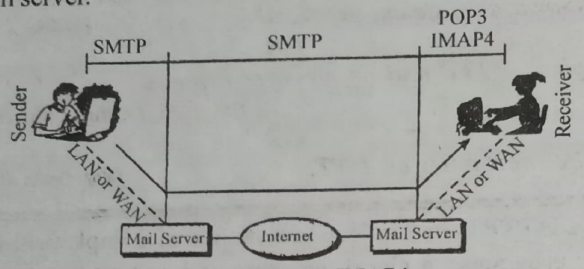


Fig. 2 : POP3 and IMAP4

Another mail access protocol is Internet Mail Access Protocol version 4 (IMAP4). IMAP4 is similar to POP3, but it has more features, IMAP4 is more powerful and more complex.

POP3 is deficient in several ways. It does not allow the user to organize her mail on the server, the user cannot have different folders on the server. In addition, POP3 does not allow the user to partially check the contents of the mail before downloading.

IMAP4 provides the following extra functions :

- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- A user can create, delete or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

Q.12 ***Explain the differences in persistent and non-persistent HTTP.*** ***[R.T.U. 2012]***

Ans. Persistent and Non-persistent HTTP are differentiated below, with the help of their advantages and disadvantages :

Advantages of Persistent HTTP Connections

Faster Content Delivery, less round-trip time, everything is served via the same TCP stream which obviously saves lots of time. When adding HTTP pipelining support to this, things are even faster.

This is also extremely beneficial when it comes to secure delivery using the SSL protocol, which require extra round-trips.

1. Less CPU Usage: We are involving a less amount of low level OS routine calls.

2. Reduced Network Congestion: Less packets on the line, and more control for TCP to handle the congestion in a single (from RFC 2616 notes on persistent connections).

Disadvantages

1. Possible Scalability Issues: In case of a traffic burst, all the "slots" on the web server (the connections pool) are kept busy by few users, while everybody else waits for a server response. This also happens with non-persistent connections as well, however the time to serve a different HTTP request is lower than with persistent connection, because there is no time-out period.

2. No Simplicity Friendliness: A server serving simple one time files (such as AJAX request, basic HTML files with no embedded objects, XML files, updates statuses) has no reason to server HTTP content via persistent connection, because the one-time required content can be served using one HTTP request and the client will be gone.

3. Non-CDN Friendly: If serving content from a CDN which delivers content from multiple web servers for the purpose of increasing the delivery speed, the CDN solution is nilled by not being able to deliver multiple HTTP response via the same TCP connection.

Advantages of Non-Persistent HTTP connections

1. Possibly More Scalable : Depending on the design of the application and the usage patterns (the disadvantages for persistent HTTP connections), more clients can be served if they require content from the server sporadically.

2. Simple Server Architecture : The server may be a bit faster if it does not require the implementation of the persistent HTTP connections and the pipelining support.

Disadvantages

1. Possibly Less Scalable: Depending on the type of traffic the server gets, serving individual HTTP requests on their own TCP stream may quickly starve the server's resources.

2. More CPU Usage: There are low level operating system routines involved in opening a new TCP stream for each request. This puts the web server under more work.

Q.13 ***What is DNS poisoning? Explain the bad effect of DNS poisoning.*** ***[R.T.U. 2012]***

Ans. DNS Poisoning :

DNS Poisoning or DNS Cache Poisoning

DNS poisoning is also called DNS cache poisoning, and refers to the corruption of DNS tables and caches so

that a domain name points to a malicious IP address. Once the user is redirected to the malicious IP address his/her computer can be infected with worms, viruses, spy ware etc.

Cache poisoning is mostly done through spam emails, or through web-links and banners that attracts users to click on them. A simple click causes the user to be re-directed to a DNS poisoned server.

Cache poisoning is a security of data integrity compromise in the Domain Name System (DNS). The compromise occurs when data is introduced into a DNS name server's cache database that did not originate from authoritative DNS sources. It may be a deliberate attempt of a maliciously crafted attack on a name server. It may also be an unintended result of a misconfiguration of a DNS cache or from improper software data and caches it for performance optimization, it is considered poisoned, supplying the non-authentic data to the clients of the server.

A domain name server translates a domain name (such as example.com) into an IP address that Internet hosts use to contact Internet resources. If a DNS server is poisoned, it may return an incorrect IP address, diverting traffic to another computer.

Effect of DNS Poisoning

Poisoning attacks on a single DNS server can affect the users serviced directly by the compromised server or indirectly by its downstream server(s) if applicable.

To perform a cache poisoning attack, the attacker exploits a flaw in the DNS software. If the server does not correctly validate DNS responses to ensure that they are from an authoritative source (for example by using DNSSEC) the server will end up caching the incorrect entries locally and serve them to other users that make the same request. This technique can be used to direct users of a website to another site of the attacker's choosing.

- For example, an attacker spoofs the IP address DNS entries for a target website on a given DNS server, replacing them with the IP address of a server he controls. Then he creates files on the server he controls with names matching those on the target server.
- These files could contain malicious content, such as a computer worm or a computer virus. A user whose computer has referenced the poisoned DNS server would be tricked into accepting content coming from a non-authentic server and unknowingly download malicious content.

PART-C

Q.14 ***Explain the HTTP protocol with the help of suitable diagrams.*** *[R.T.U. 2019]*

OR

Describe the steps involved when a web browser requests for and obtains a web page from a web server. Why HTTP is known as stateless protocol? *[R.T.U. 2010]*

OR

Explain HTTP and its message formats. *[R.T.U. 2015, 13, Raj. Univ. 2008]*

OR

Write short notes on HTTP. *[Raj. Univ. 2006]*

Ans. HTTP is an Application layer protocol, implemented in two programs : a **client program** and a **server program**. The client program and server program, executing on different end systems, talk to each other by exchanging HTTP messages.

A **Web page** consists of **objects.** An **object** is simply a file–HTML file, a JPEG image, a GIF image, a Java applet, an audio clip etc., that is **addressable by a single URL.**

A **browser** is a **user agent** for the **web,** it displays the requested **Web page** and provides numerous navigational and configuration features.

HTTP defines how web clients request web pages from the web and how servers transfer web pages to client. Both HTTP/1.0 and HTTP/1.1 use **TCP** as their underlying transport protocol.

The HTTP client first initiates a TCP connection with the server. Once the connection is established, the browser and the server processes access TCP through their **Socket** I/Fs. On the client side the socket I/F is the "door" between, the **client process and the TCP connection**, on the server side it is the "door" between the **server process** and the TCP connection.

Once the client sends a message into its socket I/F, the message is "out of the client's hands" and is "in the hands of TCP". TCP is used because it provides the **reliable service.**

HTTP can use both **non-persistent connections** and **persistent connections**.

HTTP Message Format :

1. HTTP Request Message :

GET/Somedir/page.html HTTP/1.1 → (Request line)
Host : www.nkjaipur.com
Connection : close → (header lines)
User–agent : Mozilla/4.0 → (Netscape browser)

Accept–language

Request line has three fields : (1) The method field, (2) The URL field, (3) The HTTP version field.

The header line **Host** specifies the host on which the object resides.

In header line **Connection,** the browser is telling the server that it doesn't want to use persistent connection, it wants the server to close the connection after sending the request object.

The header line **User–agent,** specifies the type of browser used to make request to the server.

The header line **Accept–language,** indicates that the user prefers to receive a **French Version** of the object.

2. HTTP Response Message :

HTTP/1.1 200 OK 3 → Status line

Connection : Close

Date : TUE, 08 Feb. 2005 11:30:15 Ist → When the HTTP response was created and sent by the server.

Server : Apache1.3.0 (Unix)

Host–modified : SUN, 06 Feb 2005 16:15:20 Ist → When the object was created or last modified.

Content–length : 7557

Content–type : Image /JPEG

Data ..... data → Entity body

The status line has three fields : The protocol version field, a status code, and corresponding status message.

Ex. (1) 200 OK : Request succeeded and the information is returned in the response.

(2) 301 Named permanently : Requested object has been permanently named.

(3) 400 Bad Request : The request could not be understood by the server.

(4) 404 Not found : The requested document does not exist on 12 server.

(5) 505 HTTP Version Not Supported : The request HTTP protocol version is not supported by the server.

Most Web pages consist of a base HTML file and several referenced objects. For example, if a Web page contains HTML text and five JPEG images, the the Web page has six objects: the base HTML file plus the five images. The base HTML file references the other objects in the page with the objects' URLs. Each URL has two components: the host name of the server that houses the object and object's path name.

For example, the URL http://www.someSchool.edu/someDepartment/picture.gif has www.someSchool.edu for a host name and /someDepartment/picture.gif for a path name. A browser is a user agent for the Web, it displays the requested Web page to the user and provides numerous navigational and configuration features.

HTTP defines how Web clients (for example, browsers) request Web pages from Web servers and how servers transfer Web pages to clients. but the general idea is illustrated in Figure. When a user requests a Web page (for example, click on a hyperlink), the browser sends HTTP request messages for the objects in the page to the server. The server receives the requests and responds with HTTP response message that contain the objects.

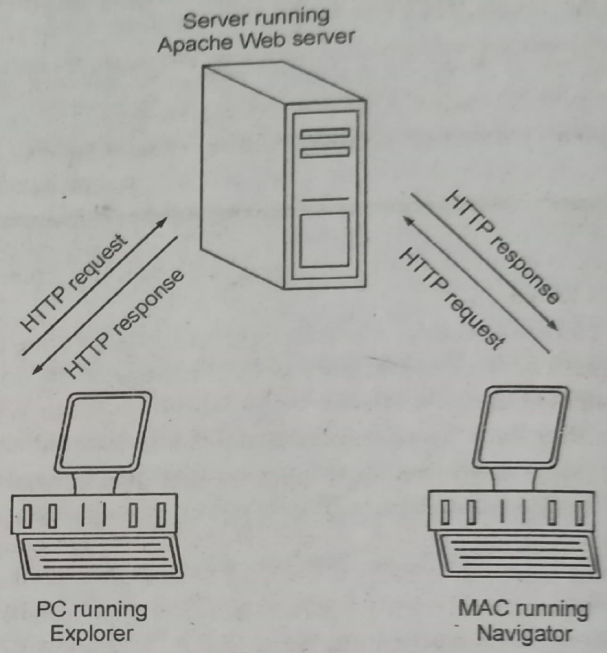


Fig. : HTTP request-response behaviour

A Web server running 1.1 can "talk" with a browser running 1.0, and a browser running 1.1 can "talk" with a server running 1.0. Because HTTP/1.1 is now dominant, henceforth when we refer to HTTP we are referring to HTTP/1.1.

HTTP uses TCP as its uderlying transport protocol (rather than running on top of UDP). The HTTP client first initiates a TCP connection with the server. Once the connection is established, the browser and the server processes access TCP through their socket interfaces. On the client side the socket interface is the door between the client process and the TCP connection, on the server side it is door between the server process and the TCP connection. The client sends HTTP request messages into its socket interface and receives HTTP response messages from its socket interface. Similarly, the HTTP server receives request messages from its socket interface and sends response message into its socket interface. Once the client sends a message into its socket interface, the message is out of the client's hands and is "in the hands" of TCP. TCP provides a reliable data transfer service to HTTP. This implies that each HTTP request message emitted by a client process eventually arrives intact at the server, similarly, each HTTP response message emitted by the server process eventually architecture – HTTP need not worry about lost data or the details of how TCP recovers from loss or reordering of data within the network. That is the job of TCP and the protocols in the lower layers of the protocols stack.

It is important to note that the server sends requests files to clients without storing any state information about the client. If a particular client asks for the same object twice in a period of a few seconds, the server does not responds by saying that it just served the object to the client, instead, the server resends the object, as it has completely forgotten what it did earlier. Because an HTTP server maintains no information about the clients, HTTP is said to be *stateless protocol.*

Q.15 *What is Network Security? Explain the principles of Network Security. Also discuss the various challenges in implementation of security in computer network.*

[R.T.U. 2016]

Ans. Network security is the process through which a network is secured against internal and external threats of various forms.

Network security measures to protect data during their transmission. It does not only concern the security in the computers at each end of the communication chain. When transmitting data the communication channel should not be vulnerable to attack. Securing the network is just as important as securing the computers and encrypting the message. With the advent of the internet, security became a major concern and the history of security allows a better understanding of the emergence of security technology. System and network technology is a key technology for a wide variety of applications. Security is crucial to networks and applications. Although, network security is a critical requirement in emerging networks, there is a significant lack of security methods that can be easily implemented.

An effective network security plan is developed with the understanding of security issues, potential attackers, needed level of security, and factors that make a network vulnerable to attack. Security protocols sometimes usually appear as part of a single layer of the OSI network reference model. Current work is being performed in using a layered approach to secure network design. The layers of the security model correspond to the OSI model layers. This security approach leads to an effective and efficient design which circumvents some of the common security problems. A secure network will also prevent someone from inserting unauthorized messages into the network.

Principles of Network Security -

1. Access – Authorized users are provided the means to communicate to and from a particular network. The availability of data is a measure of the data's accessibility. For example, if a server were down only five minutes per year, it would have an availability of 99.999 percent (that is, "five nines" of availability).

Here are a couple of examples of how an attacker could attempt to compromise the availability of a network:

- He could send improperly formatted data to a networked device, resulting in an unhandled exception error.
- He could flood a network system with an excessive amount of traffic or requests. This would consume the system's processing resources and prevent the system from responding to many legitimate requests. This type of attack is called a denial-of-service (DoS) attack.

2. Confidentiality – Information in the network remains private. Data confidentiality implies keeping data private. This privacy could entail physically or logically restricting access to sensitive data or encrypting traffic traversing a network. A network that provides confidentiality would do the following, as a few examples:

- Use network security mechanisms (for example, firewalls and access control lists [ACL]) to prevent unauthorized access to network resources.
- Require appropriate credentials (for example, usernames and passwords) to access specific network resources.
- Encrypt traffic such that an attacker could not decipher any traffic he captured from the network.

3. Authentication – Ensure the users of the network are who they say they are. Communication entities identify each other through the use of digital certificates. Web-server authentication is mandatory whereas client authentication is kept optional.

4. Message Integrity – Data integrity ensures that data has not been modified in transit. Also, a data integrity solution might perform origin authentication to verify that traffic is originating from the source that should be sending it.

Examples of integrity violations include –

- Modifying the appearance of a corporate website
- Intercepting and altering an e-commerce transaction
- Modifying financial records that are stored electronically

Various challenges in implementation of security in computer network : Various network security threats are eavesdropping, injecting false data via intercept/alter, service spoofing, and resource exhaustion. Some smart grid administrators do not even concern themselves with the spread of malware (like from viruses or Trojan horses) or the risk of a remote attacker assuming control of the system, believing that the firewall and other network protection on their computer system will be sufficient. However, many of these systems use HTTP and TCP/IP protocols, two systems that have documented vulnerabilities. Eavesdropping is the situation where an outsider intruder listens or gathers data intended for the smart grid system. In this attack, the attacker, or eavesdropper, taps into the transmission signal between the data source (a home sensor, for instance) and the smart grid control centre. Eavesdropper can intercede when the

data is encoded and when it is decoded. Such successful decoding could then lead to the next type of attack: injecting false data. In this attack, the malicious intruder intercepts valid data and transmits false data to the control centre. Most control systems are decided to question or ignore data whose mean square difference from the normal or expected is too high. Cyber-security in the smart grid is required at the perimeter as well as internal to the network. The standard perimeter defense would include firewalls, intrusion detection systems, and secure architecture, while the internal defense would include integrity checks, network monitoring, and log analysis. In addition, it is necessary to institute a key exchange mechanism along with protocols for end-to-end encryption of data. It is also necessary to institute robustness to false-data injection (FDI) and denial-of-service attacks by creating redundant channels and fall-back positions for state estimation and load forecasting. The human factor takes into account the different habits, personalities, computer proficiencies and education which tend to cause the greatest challenges to network security administrators. Any security system, no matter how well-designed and implemented, will have to depend on people to manage the network, and use it as intended.

Q.16 ***Draw and explain Domain Name System (DNS) record structure.*** ***[R.T.U. 2016]***

OR

Explain DNS with its messaging scheme and record format. Discuss the resolution process of DNS. ***[R.T.U. 2013]***

Ans. DNS Message : *Refer to Q.2.*

- **Header :** Both query and response messages have the same header format with some fields set to zero for the query messages. The header is 12 bytes, and its format is shown in figure.

Identification	Flags
Number of question records	Number of answer records (all 0s in query message)
Number of authoritative records (all 0s in query message)	Number of additional records (all 0s in query message)

Fig. : Header Format

The **identification** subfield is used by the client to match the response with the query. The client uses a different identification number each time it sends a query. The server duplicates this number in the corresponding response. The **flags** subfield is a collection of subfields that define the type of the message, the type of answer requested, the type of desired resolution (recursive or iterative), and so on. The **number of question records** subfield contains the number of queries in the question section of the message. The **number of answer records** subfield contains the number of answer records in the answer section of the response message. Its value is zero in the query message. The **number of authoritative records** subfield contains the number of authoritative records in the authoritative section of a response message. Its value is zero in the query message. Finally, the **number of additional records** subfield contains the number additional records in the additional section of a response message. Its value is zero in the query message.

- **Question Section :** This is a section consisting of one or more question records. It is present on both query and response messages.
- **Answer Section :** This is a section consisting of one or more resource records. It is present only on response messages. This section includes the answer from the server to the client (resolver).
- **Authoritative Section :** This is a section consisting of one or more resource records. It is present only on response messages. This section gives information (domain name) about one or more authoritative servers for the query.
- **Additional Information Section :** This is a section consisting of one or more resource records. It is present only on response messages. This section provides additional information that may help the resolver. For example, a server may give the domain name of an authoritative server to the resolver in the authoritative section, and include the IP address of the same authoritative server in the additional information section.

DNS Records

There are two types of records used in DNS. The question records are used in the question section of the query and response messages. The resource records are used in the answer, authoritative and additional information sections of the response message.

- **Question Record :** A question record is used by the client to get information from a server. This contains the domain name.
- **Resource Record :** Each domain name (each node on the tree) is associated with a record called the resource record. The server database consists of resource records. Resource records are also what are returned by the server to the client.

Resource Record

A Resource Record (RR) is the basic data element in the domain name system. Each record has a type (*A, MX, etc.*), an expiration time limit, a class, and some type-specific data. Resource records of the same type define a resource record set. The order of resource records in a set, returned by a resolver to an application, is undefined, but often servers implement round-robin ordering to achieve load balancing.

Field	Description	Length (octets)
NAME	Name of the node to which this record pertains	(variable)
TYPE	Type of RR in numeric form (e.g. 15 or MX RRs)	2
CLASS	Class code	2
TTL	Count of seconds that the RR stays valid (The maximum is $2^{31}-1$), which is about 68 years.)	4
REDLENGTH	Length of RDATA field	2
RDATA	Additional RR-specific data	(variable)

Fig. : Resource Record (RR) Fields

- **NAME** is the fully qualified domain name of the node in the tree. On the wire, the name may be shortened using label compression where ends of domain names mentioned earlier in the packet can be substituted for the end of the current domain name.
- **TYPE** is the record type. It indicates the format of the data and it gives a hint of its intended use. For example, the *A* record is used to translate from a domain name to an IPv4 address, the *NS* record lists which name servers can answer lookups on a DNS zone, and the *MX* record specifies the mail server used to handle mail for a domain specified in an e-mail address .
- The **CLASS** of a record is set to IN (for *Internet*) for common DNS records involving Internet hostnames, servers, or IP addresses. In addition, the classes Chaos (CN) and Hesiod (HS) exist. Each class is an independent name space with potentially different delegations of DNS zones.
- **TTL** is the time to live of the resource record. It determines when a resource should be removed from a cache.
- **RDATA** is data of type-specific relevance, such as the IP address for address records, or the priority and hostname for MX records. Well known record types may use label compression in the RDATA field, but "unknown" record types must not.

In addition to resource records defined in a zone file, the domain name system also defines several request types that are used only in communication with other DNS nodes (*on the wire*), such as when performing zone transfers.

Resolution Process in DNS

Mapping a name to an address or an address to a name is called *name-address resolution.*

1. Resolver

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 called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide information.

After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.

2. Mapping Names to Addresses

Generally, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping.

If the domain name is from the generic domains section, the resolver receives a domain name such as *"chal.atc.jhda.edu."*. The query is sent by the resolver to the local DNS server for resolution. If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.

If the domain name is from the country domains section, the resolver receives a domain name such as *"ch.jhda.cu.ca.us."*. The procedure is the same.

3. Mapping Addresses to Names

A client can send an IP address to a server to be mapped to a domain name. This is called a PTR query. To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address is reversed and the two labels *in-addr* and *arpa* are appended to create a domain acceptable by the inverse domain section. For example, if the resolver receives the IP address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending. The domain name sent is *"121.45.34.132.in-addr.arpa."* which is received by the local DNS and resolved.

4. Recursive Resolution : The client (resolver) can ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer.

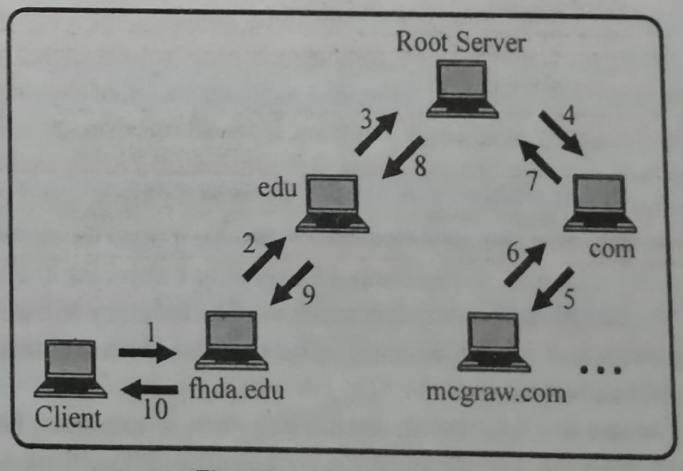


Fig. : Recursive Resolution

If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response. If the parent is the

authority, it responds; otherwise, it sends the query to yet another server. When the query is finally resolved, the response travels back until it finally reaches the requesting client. This is called recursive resolution and is shown in figure.

5. Iterative Resolution : If the client does not ask for a recursive answer, the mapping can be done iteratively. If the server is an authority for the name, it sends the answer. If no, it returns (to the client) 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. If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwise, it returns the IP address of a new server to the client. Now the client must repeat the query to the third server. This process is called iterative resolution because the client repeats the same query to multiple servers. In figure the client queries four servers before it gets an answer from the mcgraw.com server.

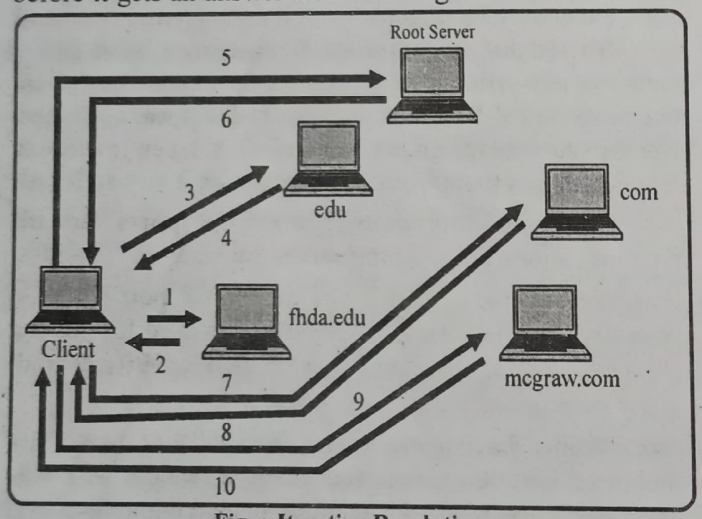


Fig. : Iterative Resolution

6. Caching

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. Reduction of this search time would increase efficiency. DNS 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 solve the problem. However, 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 speeds up resolution, but it can also be problematic. If a server caches a mapping for a long time, it may send an outdated mapping to the client. To counter this, two techniques are used. First, the authoritative server always adds information to the mapping called *time-to-live* (TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server. Second, DNS requires that each server keep 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.

Q.17 ***Write short note on :***
(a) World Wide Web (WWW)
(b) File Transfer Protocol (FTP) ***[R.T.U. 2016]***

Ans. (a) World Wide Web (WWW) : The WWW today is a distributed client/server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *sites,* as shown in fig.1.

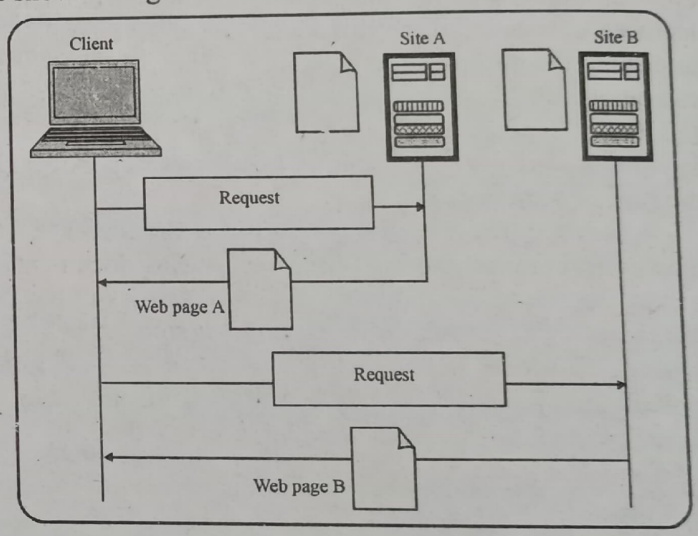


Fig.1 : WWW Architecture

Each site holds one or more documents, referred to as **Web pages**. Each Web page can contain a link to other pages in the same site or at other sites. The pages can be retrieved and viewed by using browsers. Let us go through the scenario shown in fig.1. The client needs to see some information that it knows belongs to site A. It sends a request through its browser, a program that is designed to fetch Web documents. The request, among other information, includes the address of the site and the Web page, called the **URL** (Uniform Resource Locator). The server at site A finds the document and sends it to the client. When the user views the document, she finds some references to other documents, including a Web page at site B. The reference has the URL for the new site. The user is also interested in seeing this document. The client sends another request to the new site, and the new page is retrieved.

Client (Browser) : A variety of companies offers many commercial browsers that interpret and display a Web document, and all use nearly the same architecture.

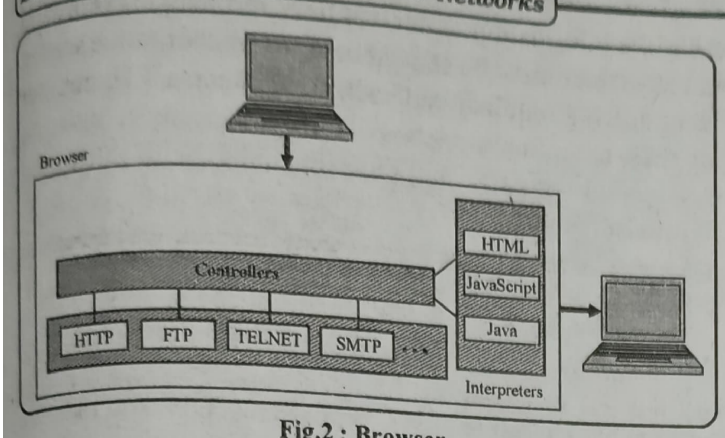


Fig.2 : Browser

Each browser usually consists of three parts: a controller, client protocol, and interpreters. The controller receives input from the keyboard or the 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. The client protocol can be one of the protocols such as File Transfer Protocol (FTP) or Hyper Text Transfer Protocol (HTTP). The interpreter can be HTML, Java, or JavaScript, depending on the type of document.

Server : The Web page is stored at the server. Each time a client request arrives, the corresponding document is sent to the client. To improve efficiency, servers normally store requested files in a cache in memory; memory is faster to access than disk. A server can also become more efficient through multithreading or multiprocessing. In this case, a server can answer more than one request at a time.

Uniform Resource Locator (URL) : *Refer to Q.8(b).*

Cookies : *Refer to Q.9(a).*

Using Cookies : *Refer to Q.6.*

Ans.(b) File Transfer Protocol (FTP) : The File Transfer Protocol (FTP) is used to copy files between two computer systems over the TCP connection. The FTP overcomes the problem of different file systems used in the network.

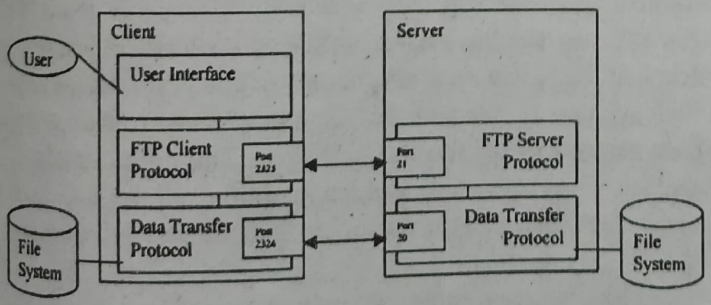


Fig. : FTP protocol model

In the FTP, the user communicates with a user interface in the local FTP client process. The local FTP client process makes a control connection to the remote server's FTP server protocol. FTP server protocol is located in the TCP port 21. The local FTP client acts as a protocol interpreter who interprets the user commands to the acronyms used between the client and the server protocol. The control connection is basically a simple TELNET's NVT session. The control connection is used in a very simple way-

The client sends commands across the control connection to the server. The server replies to the messages according to the server protocol. If the user request a data transfer, a special data connection is opened between the server and the client and the files are sent through this connection. Separate data transfer process created for the server and the client. The data connection exists until the command that it was created for is executed. File transfer protocol contains set of command words and their parameters and numeric codes as responses. The command words can be classified to the access control, file management, data format setting, file transfer, site, error recovery and restart commands.

Earlier FTP protocols used for the Internet standard were drafted by the Internet Engineering Task Force committee as a series of RFC (Request for Comments) formal documents. In 1971 the FTP protocol RFC 114 was published. Over the years the document was revised with newer versions making changes to improve the FTP protocol.

FTP connects using two TCP ports for all communications between the server and user-

COMMAND Port: This is the main TCP port which is created when a session is connected. It is used for passing commands and replies. Port 21 (unsecured) or 990 (secured) are the default command ports used.

DATA Port: Each time when files or directories are transferred between server and client, a random TCP data connection is established and data transfer commences over the connection. Once data transfer is complete, the connection is closed. Subsequent data connections are established and terminated as required. Data connections are never left open.

Q.18 ***Explain e-mail architecture along with its components.*** ***[R.T.U. 2010]***

Ans. E-mail Architecture : To explain the architecture of e-mail, we have four scenarios. We begin with the simplest situation and add complexity as we proceed. The fourth scenario is the most common in the exchange of email. These scenarios provide a general overview of e-mail system.

First Scenario : In the first scenario, the sender and the receiver of the e-mail are users (or application programs) on the same system, they are directly connected to a shared system. The administrator has created one mailbox for each user where the received messages are stored. A mailbox is part of a local hard drive, a special file with permission

restrictions. Only the owner of the mailbox has access to it. When **A**, a user, needs to send a message to **B**, another user, **A** runs a user agent (UA) program to prepare the message and store it in **B's** mailbox. The message has the sender and recipient mailbox addresses (names of files). **B** can retrieve and read the contents of his mailbox at his convenience, using a user agent. Fig. 1 shows the concept.

"When the sender and the receiver of an e-mail are on the same system, we need only two user agents".

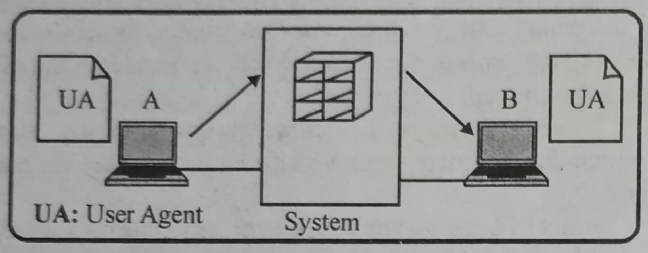


Fig. 1 : First Scenario in E-mail

Second Scenario : In the second scenario, the sender and the receiver of the e-mail are users (or application programs) on two different systems. The message needs to be sent over the Internet. Here we need user agents (UAs) and message transfer agents (MTAs), as shown in fig. 2.

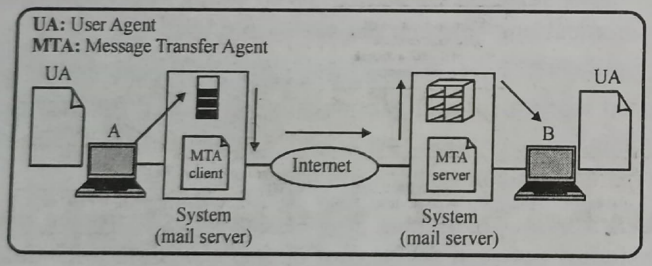


Fig. 2 : Second Scenario in E-mail

A needs to use a user agent program to send message to the system at own site. The system (sometimes called the mail server) at its site uses a queue to store messages waiting to be sent. **B** also needs a user agent program to retrieve messages stored in the mailbox of the system at its site. The message, however, needs to be sent through the Internet from **A's** site to **B's** site. Here two message transfer agents are needed : one client and one server. Like most client/server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection. The client, on the other hand, can be alerted by the system when there is a message in the queue to be sent.

"When the sender and the receiver of an e-mail are on different systems, we need 2 UAs and a pair of MTAs (client and server)".

Third Scenario : In the third scenario, **B**, as in the second scenario, is directly connected to his system. **A**, however, is separated from her system. Either **A** is connected to the system *via.* a point-to-point WAN, such as a dial-up modem, a DSL, or a cable modem, or is connected to a LAN in an organization that uses one mail server for handling e-mails and all users send their messages to this mail server. Fig. 3 shows the situation.

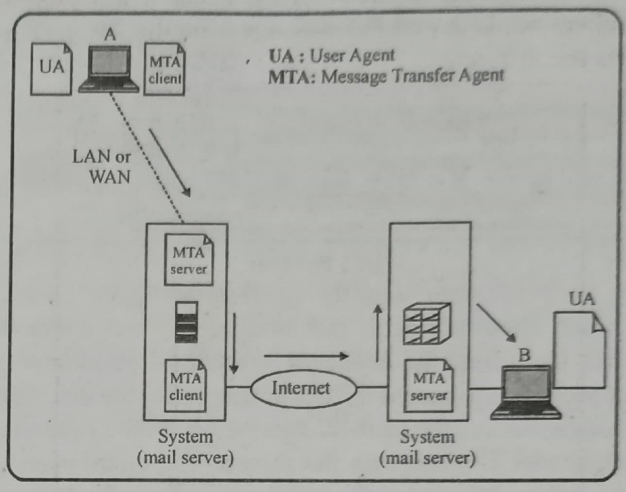


Fig. 3 : Third scenario in E-mail

A still needs a user agent to prepare its message. **A**, then needs to send the message through the LAN or WAN. This can be done through a pair of message transfer agents (client and server). Whenever **A** has a message to send, calls the user agent which, in turn calls the MTA client. The MTA client establishes a connection with the MTA server on the system, which is running all the time. The system at **A's** site queues all messages received. It then uses an MTA client to send the messages to the system at **B's** site, the system receives the message and stores it in **B's** mailbox. At his convenience, **B** uses its user agent to retrieve the message and reads it. Note that we need two pairs of MTA client/server programs.

"When the sender is connected to the mail server via a LAN or a WAN, we need two UAs and two pairs of MTAs (client and server)".

Fourth Scenario : In the fourth and most common scenario, **B** is also connected to mail server by a WAN or a LAN. After the message has arrived at **B's** mail server, **B** needs to retrieve it. Here, we need another set of client/server agents, which we call message access agents (MAAs). **B** uses an MAA client to retrieve its messages. The client sends a request to the MAA server, which is running all the time, and requests the transfer of the messages. The situation is shown in Fig. 4.

There are two important points here. First, **B** cannot bypass the mail server and use the MTA server directly. To use MTA server directly, **B** would need to run the MTA server all the time because it does not know when a message will arrive. This implies that **B** must keep its computer on all the time if he is connected to his system through a LAN. If he is connected through a WAN, he must keep the connection up all the time. Neither of these situations is feasible today.

Second, note that **B** needs another pair of client/server programs : message access programs. This is so because an MTA client/server program is a *push* program : the client pushes the message to the server. **B** needs a *pull* program. The client needs to pull the message from the server. Fig. 5 shows the difference.

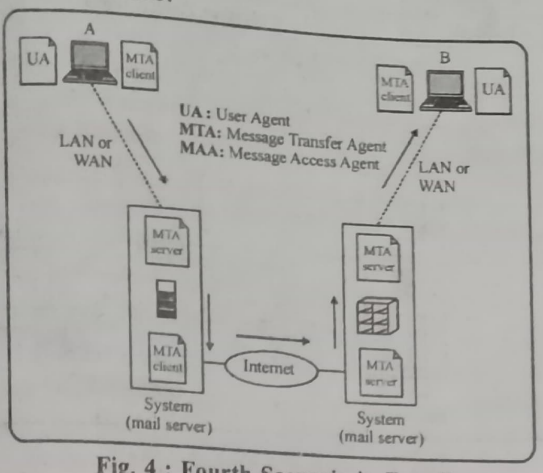


Fig. 4 : Fourth Scenario in E-mail

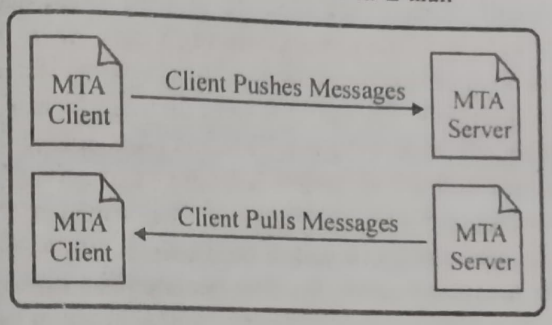


Fig. 5 : Push versus Pull in E-mail

"When both sender and receiver are connected to the mail server via a LAN or a WAN, we need two UAs, two pairs of MTAs (client and server), and a pair of MAAs (client and server). This is the most common situation today".

Q.19 ***(a) Explain the WWW services of the Internet. Also distinguish between static, dynamic and active web documents.***

(b) What do you mean by DNS?

[R.T.U. 2009, Raj.Univ. 2008]

OR

Explain the services provided by the DNS and discuss the problems related with centralized design for DNS.

[Raj. Univ. 2006]

Ans.(a) World Wide Web (WWW) : The **World Wide Web (WWW)**, or the **web**, is a repository of information spread all over the world and linked together. sThe WWW has a unique combination of flexibility, portability, and user-friendly features that distinguish it from other services provided by the Internet.

The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called web sites.

Hypertext and Hypermedia

The WWW uses the concept of hypertext and hypermedia. In a hypertext environment, information is stored in a set of documents that are linked together using the concept of pointers. An item can be associated with another document using a pointer. The reader who is browsing through the document can move to other documents by choosing (clicking) the items that are linked to other documents. Fig.(1) shows the concept of hypertext.

Whereas hypertext documents contain only text, hypermedia documents can contain pictures, graphics and sound.

A unit of hypertext or hypermedia available on the Web is called a **page.** The main page for an organization or an individual is known as a **homepage.**

Browser Architecture

A variety of vendors offer commercial browsers that interpret and display a web document, and all of them use nearly the same architecture.

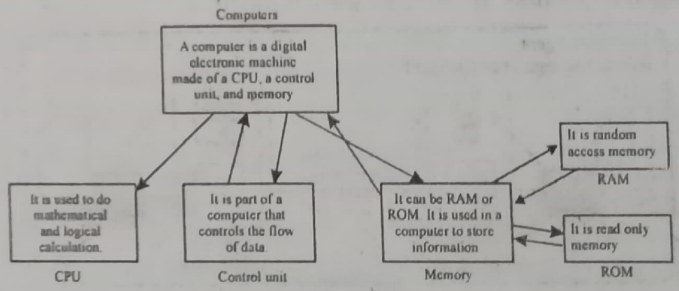


Fig.1: Concept of Hypertext

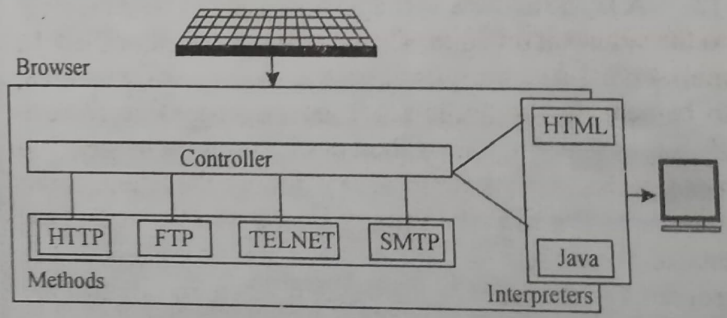


Fig.2: Browser Architecture

Each browser usually consists of three parts: a controller, client programs, and interperters. The **controller** receives input from the keyboard or the 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. The **client programs** can be one of the methods (protocols) such as HTTP, FTP, or TELNET. The **interpreter** can be HTML or Java, depending on the type of document.

Web Documents : The documents in the WWW can be grouped into three broad categories: static, dynamic, and active (Figure 3). The category is based on the time when the contents of the document are determined.

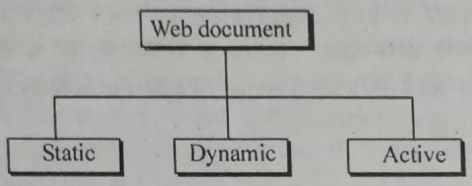
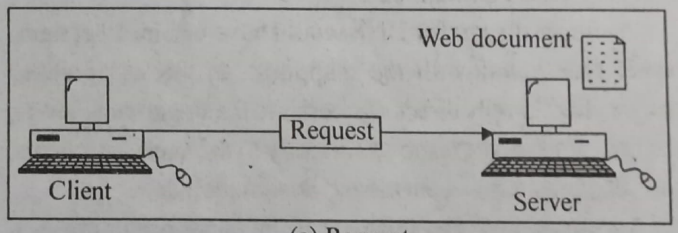


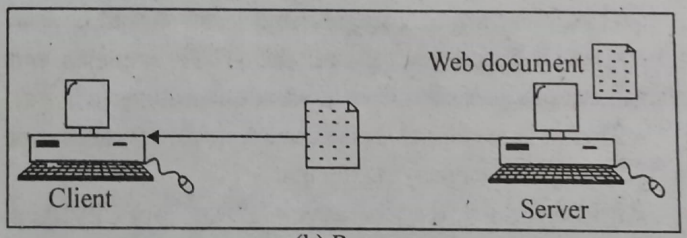
Fig.3: Categories of Web Documents

The documents in the WWW can be grouped into three broad categories: static, dynamic, and active (Fig. 3). The category is based on the time when the contents of the document are determined.

Static Documents : Static documents are fixed-content documents that are created and stored in a server. The client can get only a copy of the document. In other words, the contents of the file are determined when the file is created, not when it is used. Of course, the contents in the server can be changed, but the user cannot change it. When a client accesses the document, a copy of the document is sent. The user can then use a browsing program to display the document (Fig.4).



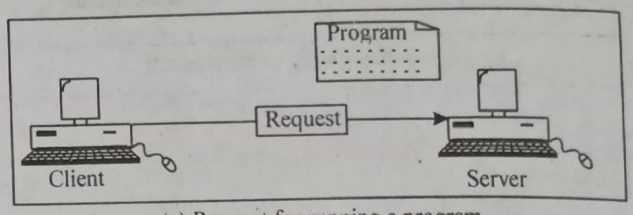
(a) Request



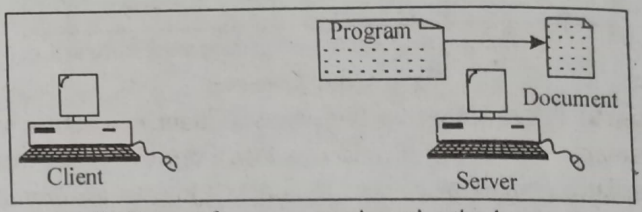
(b) Response

Fig.4 : Static Document

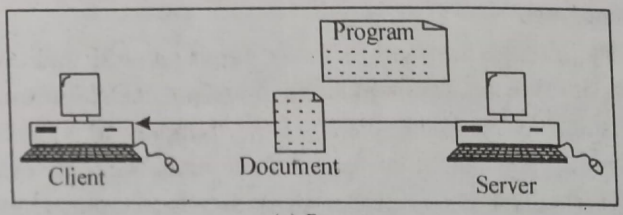
Dynamic Documents : Dynamic documents do not exist in a predefined format. Instead, a dynamic document is created by a web server whenever a browser requests the document. When a request arrives, the web server runs an application program that creates the dynamic document. The server returns the output of the program as a response to the browser that requested the document. Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another. A very simple example of a dynamic document is getting the time and date from the server. Time and date are kinds of information that are dynamic in that they change from moment to moment. The client can request that the server runs a program such as the date program in UNIX and sends the result of the program to the client. Fig.5 illustrates the steps in sending and responding to a dynamic document.



(a) Request for running a program



(b) Request for program and creating the document



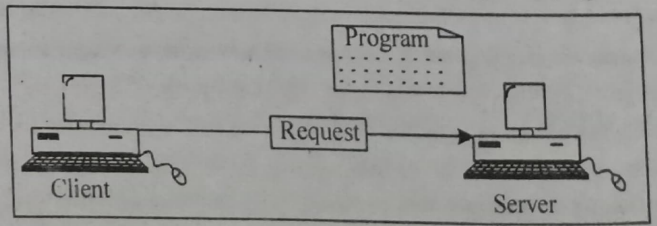
(c) Response

Fig.5: Dynamic Document

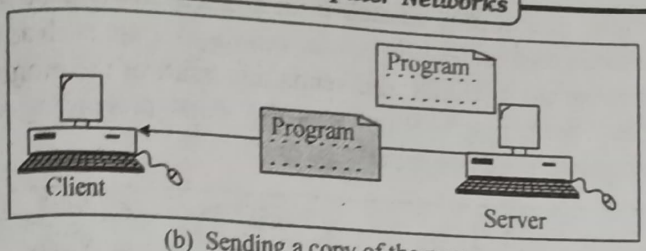
A server that handles dynamic documents follows these steps :

1. The server examines the URL to find if it defines a dynamic document.
2. If the URL defines a dynamic document, the server executes the program.
3. The server sends the output of the program to the client (browser).

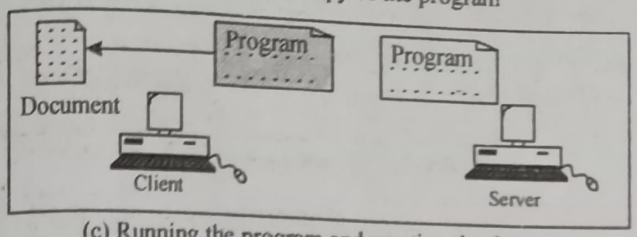
Active Documents : For many applications, we need a program to be run at the client site. These are called **active documents**. For example, imagine we want to run a program that creates animated graphics on the screen or interacts with the user. The program definitely needs to be run at the client site where the animation or interaction takes place. When a browser requests an active document, the server sends a copy of the document in the form of bytecode. The document is then run at the client (browser) site (Figure 6).



(a) Request for a copy of a program



(b) Sending a copy of the program



(c) Running the program and creating the document

Fig.6: Active Document

Ans.(b) DNS : There are two ways to identify a host–by a hostname and by an IP address. An Internet service that translates domain name into IP address, known as ***domain name service***. Domain names are alphabetic, they are easier to remember.

The Internet however, is really based on an IP address. Every time we get a domain name, therefore, a DNS service must translate the name into the corresponding IP address. This is the main task of the Internet's Domain Name Service (DNS). The DNS is: (1) a distribution database implemented in a hierarchy of name servers (2) an application–layer protocol that allows host and name servers to communicate in order to provide the translation service. Name servers are often UNIX machines running the Berkeley Internet name domain (BIND) software. The DNS protocol runs over UDP and uses Port No.53.

DNS provides a few other important services in addition to translating hostnames to IP addresses :

• **Host aliasing :** A host with a complicated host name can have one or more alias names. For example, a hostname such as relay 1. West–coast. Enterprises.com could have, two aliases such as enterprise.com and www.enterprise.com. In this case, the hostname relay 1. West–coast–enterprise–com is said to be canonical hostname DNS can be invoked by an application to obtain the canonical hostname for a supplied hostname as well as the IP address of the host.

• **Main server aliasing :** For obvious reasons, it is highly desirable that e–mail be mnemonic. For example, if dash has an account with hotmail, dash's e–mail address might be as simple as dash @ hotmail.com. However, the hostname of the Hotmail server is more complicated and much less mnemonic than simply hotmail.com. In fact, the MX record permits a Company's mail server and Web server to have identical (aliased) hostnames. For example, a Company's Web server and mail server can both be called enterprise.com.

• **Load distribution :** DNS is also being used to perform load distribution among replicated servers, such as replicated Web servers. Busy sites, such as cnn.com, are replicated over multiple servers, with each server running on a different end system, and each having a different IP address. For replicated Web servers, a set of IP address is thus associated with one canonical hostname.

The DNS database contains this set of IP addresses. When clients make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP address ,but rotates the ordering of the addresses within each reply. Because a client typically sends its HTTP request message to the IP address that is listed first in the set, DNS rotation distributes the traffic among all the replicated servers.

Problems related with centeralized design for DNS

DNS is a black box providing a simple, straight forward translation service. But infact, the black box that implements the service is complex, consisting of a large number of name servers distributed around the globe, as well as an application layer protocol that specifies how the name servers and querying hosts communicate.

A simple design for DNS would have one internet name server that contains all the mappings. In this centeralized design, client simply direct all queries to the single name server, and the name server responds directly to the querying clients. The problems with a centralized design include.

• **A single point of failure :** If the name server crashes, so does the entire internet.

• **Traffic volume :** A single name server would have to handle all DNS queries (for all the HTTP requests and e-mail messages generated from hundreds of millions of hosts).

• **Distant centralized database :** A single server cannot be "close to" all the querying clients.

If we put the single name server in New York City, then all queries from Australia must to other side of the globe, perhaps over slow and congested links. This can lead to significant delays.

• **Maintenance :** The single name server would have to keep records for all Internet hosts. Not only would this centralized database be huge, but it would have to be updated frequently to account for every new host. There are also authentication and authorization problems associated with allowing any user to register a host with the centralized database.

In summary, a centralized database in a single name server simply doesn't scale.

Q.20 *Explain the working of SMTP.*

[R.T.U. 2009, Raj. Univ. 2006]

Ans. Simple Mail Transfer Protocol (SMTP) : One of the most popular network services is **electronic mail (e-mail)**. The TCP/IP protocol that supports electronic mail on the Internet is called **Simple Mail Transfer Protocol (SMTP)**. It is a system for sending messages to other computer users based on e-mail addresses. SMTP provides for mail exchange between users on the same or different computers and supports :

- Sending a single message to one or more recipients.
- Sending messages that include text, voice, video, or graphics.
- Sending messages to users on networks outside the Internet.

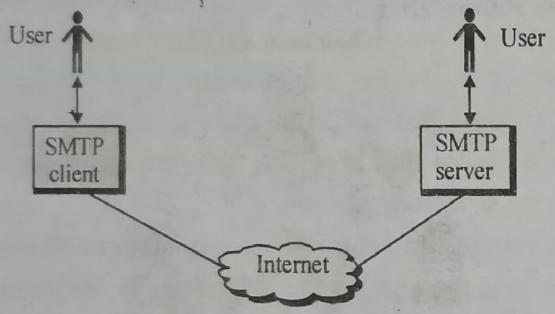


Fig.1: SMTP Concept

A simple SMTP system is shown in Fig.1. The SMTP client and server can be break down into two components: **user agent (UA)** and **mail transfer agent (MTA)**.

The UA prepares the message, creates the envelope, and puts the message in the envelope. The MTA transfers the mail across the Internet. Figure 2 shows the previous figure with the addition of these two components.

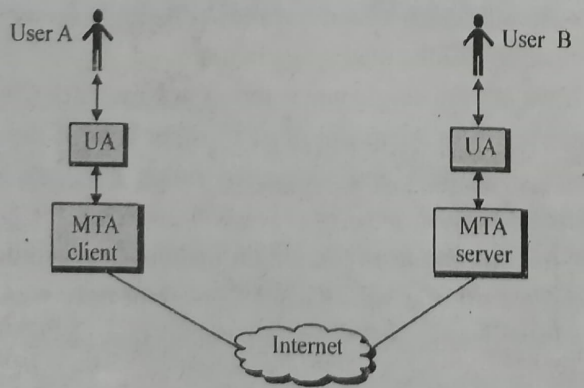


Fig.2: UAs and MTAs

SMTP protocol allows a more complex system than the one shown. Relaying could be involved. Instead of just one MTA at the sender site and one at the receiving site, other MTAs, acting either as client or server, can relay the mail.

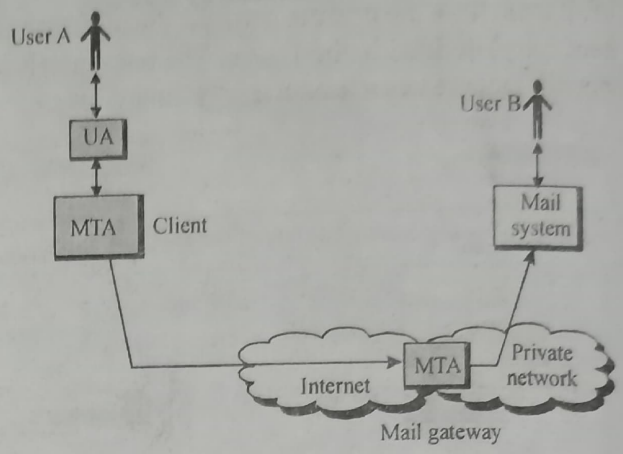


Fig.3: Mail Gateway

The relaying system allows sites that do not use the TCP/IP protocol suite to send e-mail to users on other sites that may or may not use the TCP/IP protocol suite. This is accomplished through the use of a **mail gateway**, which is a relay MTA that receives mail prepared by a protocol other than SMTP and transform it to SMTP format before sending it. It can also receive mail in SMTP format and change it to another format before sending it (Figure 3).

User Agent (UA) : A user agent is defined in SMTP. The UA is normally a program used to send and receive mail. Popular user agent programs are MH, Berkeley Mail, Elm, Zmail, and Mush.

Some user agents have an extra user interface that allows window-type interactions with the systems.

Addresses : To deliver mail, a mail handling system must use a unique addressing system. The addressing system used by SMTP consists of two parts : a **local part** and a **domain name,** separated by an @ sign.

Local Part : The local part defines the name of a special file, called the user mailbox, where all of the mail received for a user is stored for retrieval by the user agent.

Domain Name : The second part of the address is the domain name. An organization usually selects one or more hosts to receive and send e-mail, they are sometimes called mail exchangers. The domain name assigned to each mail exchanger either comes from the DNS database or is a logical name (for example, the name of the organization).

Mail Transfer Agent (MTA) : The actual mail transfer is done through mail transfer agents (MTAs). To send mail, a system must have a client MTA, and to receive mail, a system must have a server MTA. Although SMTP does not define a specific MTA, sendmail is a commonly used UNIX system MTA.

SMTP simply defines how commands and responses must be sent back and forth. Each network is free to choose a software package for implementation. Figure 4 illustrates the process of sending and receiving e-mail described above. For a computer to be able to send and receive mail using

SMTP, it must have most of the entities (the user interface is not necessary) defined in the figure. The user interface is a component that creates a user-friendly environment.

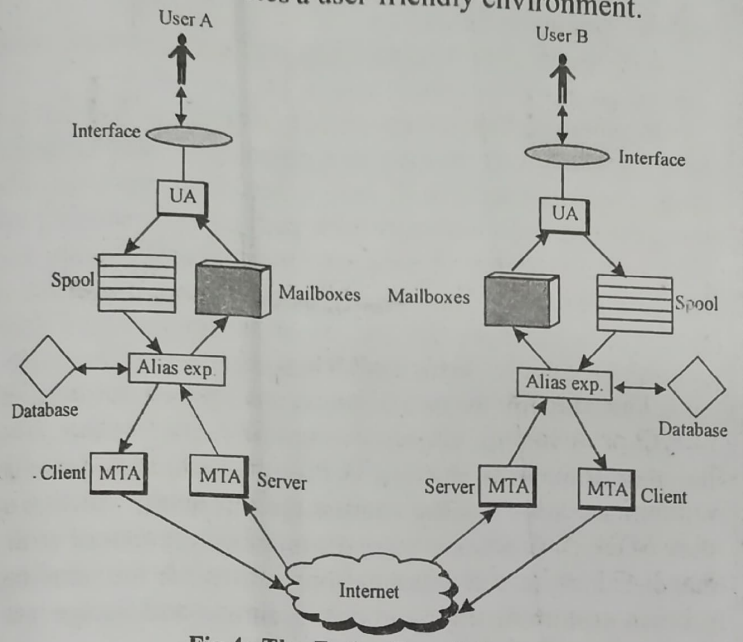


Fig.4: The Entire E-mail System

Q.21 ***Each internet host has at least one local name server and one authoritative name server. Describe role of each of these services in DNS.*** ***[Raj.Univ. 2007]***

Ans. In order to deal with the issue of scale, the DNS uses a large number of name servers, organized in a hierarchical fashion and distributed around the world. No single name server has all of the mappings for all of the hosts in the Internet. Instead, the mappings are distributed across the name servers. To a first approximation, there are three types of name servers: local name servers, root name servers, and authoritative name servers. These name servers, again to a first approximation, interact with each other and with the querying host as follows.

Local name servers : Each ISP-such as a university, an academic department, an employee's company, or a residential ISP—has a local name server (also called a default name server). When a host issues a DNS query message, the message is first sent to the host's local name server. The IP address of the local name server is typically configured by hand in a host. (On a Windows machine, we can find the IP address of the local name server used by our PC by opening the Control Panel, and then selecting "Network," then selecting an installed TCP/IP component, and then selecting the DNS configuration folder tab.) The local name server is typically "close to" the client. In the case of an institutional ISP, it may be on the same LAN as the client host, for a residential ISP, the name server is typically separated from the client host by no more than a few routers. If a host requests a translation for another host that is part of the same local ISP, then the local name server will immediately be able to provide the requested IP address. For example, when the host surf. eurecom.fr requests the IP address for baie.eurecom.fr, the local name server at eurecom will be able to provide the requested IP address without contacting any other name servers.

Root name servers : In the Internet there are a dozen or so root name servers, most of which are currently located in North America. When a local name server cannot immediately satisfy a query from a host (because it does not have a record for the hostname being requested), the local name server behaves as a DNS client and queries one of the root name servers has a record for the hostname, it sends a DNS reply message to the local name server, and the local name server then sends a DNS reply to the querying host. But the root name server may not have a record for the hostname. Instead, the root name server knows the IP address of an "authoritative name server" that has the mapping for that particular hostname.

Authoritative name servers : Every host is registered with an authoritative name server.

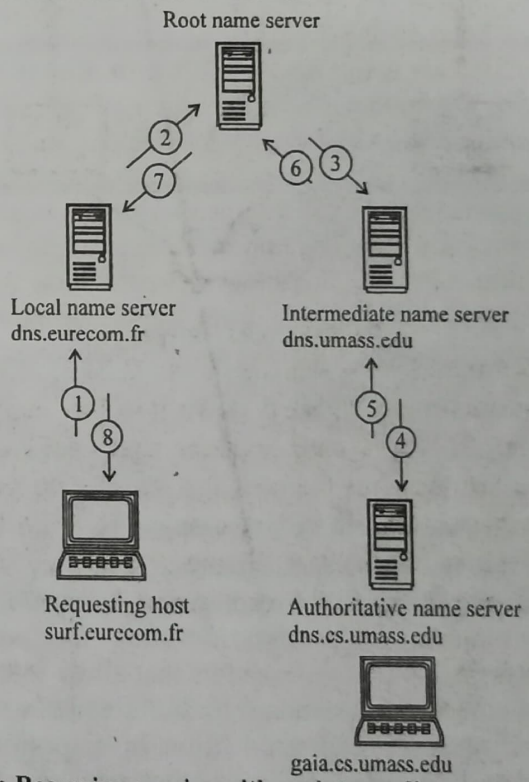


Fig. : Recursive queries with an intermediate name server between the root and authoritative name servers

Typically, the authoritative name server for a host is a name server in the host's local ISP. (Actually, each host is required to have at least two authoritative name servers, in case of failures.) By definition, a name server is authoritative for a host if it always has a DNS record that translates the host's hostname to that host's IP address. When an authoritative name server is queried by a root server, the authoritative name server responds with a DNS reply that contains the requested mapping. The root server then forwards the mapping to the local name server, which in turn forwards the mapping to the requesting host. Many name servers act as both local and authoritative name servers.

To illustrate this, consider the example above with the host surf.eurecom.fr querying for the IP address of gaia.cs.umass.edu. Suppose now that the University of Massachusetts has a name server for the University, called **dns.uinass.edu.** Also suppose that each of the departments at the University of Massachusetts has its own name server, and that each departmental name server is authoritative for all the hosts in the department. As shown in Figure, when the root name server receives a query for a host with hostname ending with uinass.edu, it forwards the query to the name server dns.umass.edu. This name server forwards all queries with hostnames ending with.cs.umass.edu to the name server dns.cs.umass.edu, which is authoritative for all hostnames ending with dns.cs.umass.edu. The authoritative name server sends the desired mapping to the intermediate name server, dns.umass.edu, which forwards the mapping to the root name server, which forwards the mapping to the local name server, dns.eurecom. fr, which forwards the mapping to the requesting host. In this example, eight DNS messages are sent.

Q.22 ***Explain what an Application Layer Protocol defines. Describe the role played by Socket in communication between two processes.*** *[Raj. Univ. 2006]*

Ans. Application–Layer Protocols : An application–layer protocol is only one piece(albeit, a big piece) of a network application. The Web is a network application that allows users to obtain "documents" from Web servers on demand. The Web application consists of many components, including a standard for document formats (that is, HTML), Web browers, Web servers and an application–layer protocol.

The Web's application–layer protocol, HTTP (the Hyper Text Transfer Protocol), defines the format and sequence of the message that are passed between browser and Web server. Thus, the Web's application–layer protocol, HTTP, is only one piece of the Web application. Internet electronic mail also has many components, including : mail servers that house user mailboxes, mail readers that allow users to read and create messages, a standard for defining the structure of an e–mail message, and application–layer protocols that define how messages are passed between servers, how messages are passed between servers and mail readers, and how the contents of certain parts of the mail message (for example, a mail message header) are to be interpreted.

The principal application–layer protocol for electronic mail is SMTP (Simple Mail Transfer Protocol). Thus, e–mail's principal application–layer protocol, SMTP, is only one piece (albeit, a big piece) of the e–mail application.

An application–layer protocol defines how an application's processes, running on different end systems, pass messages to each other. In particular, an application–layer protocol defines :

- The types of messages exchanged, for example, request messages and response messages.
- The syntax of the various message types, such as the fields in the message and how the fields are delineated.
- The semantics of the fields, that is, the meaning of the information in the fields.
- Rules for determining when and how a process sends messages and responds to messages.

Processes Communicating Across a Network : Many applications involve two processes in two different hosts communicating with each other over a network. The two processes communicate with each other by sending and receiving messages. A process sends messages into, and receives messages from, the network through its **socket**. A process's socket can be thought of as the process's door. When a process wants to send a message to another process on another host, it throws the message out its door (socket).

This sending process assumes that there is a transportation infrastructure on the other side of its door that will transport the message across the Internet to the door of the destination process. Once the message arrives at the destination host, the message passes through the receiving process's door (socket), and the receiving process then acts on the message.

Fig. illustrates socket communication between two processes that communicate over the Internet. (Fig. assumes that the underlying transport protocol is TCP, although the UDP protocol could be used as well in the Internet).

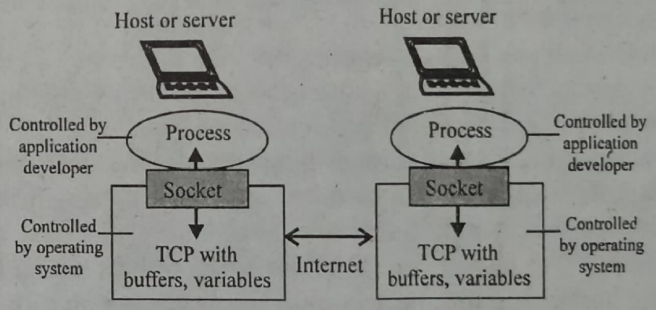


Fig. : Application processes, sockets and underlying transport protocol

A **socket** is the interface between the application layer and the transport layer within a host. It is also referred to as the **API (application programmer's interface)** between the application and the network, since the socket is the programming interface with which network applications are built in the Internet. The application developer has control of everything on the application–layer side of the socket but has little control of the transport–layer side of the socket.

The only control that the application developer has on the transport–layer side is (1) the choice of transport protocol and (2) perhaps the ability to fix a few transport–layer parameters such as maximum buffer and maximum segment sizes. Once the application developer chooses a transport protocol (if a choice is available), the application is built using the transport–layer services provided by that protocol.

□□□