CBCS SCHEME

USN | | | | | | | | | | BCS502

Fifth Semester B.E./B.Tech. Degree Examination, Dec.2025/Jan.2026
Computer Networks

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Explain fundamental characteristics and data representation in data communication.	6	L1	CO1
	b.	Discuss types of connection and the basic topologies in networks.	6	L1	CO1
	c.	Explain packet switching and circuit switching with neat diagrams.	8	L1	CO1
		OR			
Q.2	a.	Explain the layers of TCP/IP protocol suite.	8	L2	CO2
	b.	Explain types of packet switched networks and evaluate the total delay time of both.	12	L2	CO2
		Module – 2			
Q.3	a.	Explain types of errors and Hamming distance. Find the Hamming distance between the following : i) d(000, 011) ii) d(10101, 11110).	8	L3	CO2
	b.	Describe the working of CRC encoder and decoder. Perform division with respect to the following : Data word : 1001 Divisor : 1011.	12	L3	CO2
		OR			
Q.4	a.	Explain stop-and-wait protocol with FSM.	6	L2	CO2
	b.	Explain the three types of frames in HDLC.	8	L2	CO2
	c.	Discuss controlled-access protocol using reservation method.	6	L2	CO2
		Module – 3			
Q.5	a.	Explain the services offered by network layer.	6	L2	CO2
	b.	Define address space. Differentiate between classful addressing and classless addressing.	8	L2	CO2
	c.	Explain Network Address Resolution (NAT) with a neat diagram.	6	L2	CO3

		OR			
Q.6	**a.**	Explain IPv6 packet format in detail.	**6**	**L2**	**CO2**
	b.	Discuss D-V routing highlighting the importance of distance vector.	**7**	**L2**	**CO2**
	c.	Describe BGP protocol in detail.	**7**	**L2**	**CO4**
		Module – 4			
Q.7	**a.**	Explain the concept of port numbers mentioning ICANN ranges.	**5**	**L2**	**CO3**
	b.	Explain Go-Back-N protocol.	**9**	**L2**	**CO4**
	c.	Explain TCP segment format with a neat diagram.	**6**	**L3**	**CO3**
		OR			
Q.8	**a.**	Discuss the connection establishment in TCP.	**8**	**L2**	**CO3**
	b.	Explain error control in TCP using acknowledgements.	**4**	**L2**	**CO3**
	c.	Discuss three algorithms for handling congestion in TCP.	**8**	**L2**	**CO3**
		Module – 5			
Q.9	**a.**	Discuss application layer paradigms with neat diagram.	**5**	**L2**	**CO3**
	b.	Explain the use of sockets in process-to-process communication.	**7**	**L2**	**CO3**
	c.	Discuss the connection types in HTTP along with formats of messages.	**8**	**L2**	**CO3**
		OR			
Q.10	**a.**	Explain POP and IMAP protocols.	**8**	**L2**	**CO4**
	b.	Discuss the applications of SSH protocol.	**4**	**L2**	**CO4**
	c.	Explain resolution in DNS.	**8**	**L2**	**CO3**

* * * * *

VTU EXAMINATION Dec-2025/Jan-2026 SOLUTION

Sub:	**Computer Networks**					Sub Code:	**BCS502**	Branch:	**AINDS / CS (DS)**
Date:		Duration:	**3 Hrs.**	Max Marks:	**100**	Sem	**V**		**OBE**

			MARKS	**RBT**	**CO**
Q	a	Explain fundamental characteristics and data representation in data communication	6	L1	CO1
	b	Discuss types of connection and basic topologies in the network	6	L1	CO1
	c	Explain packet switching and circuit switching with neat diagram	8	L1	CO1
Q 2	a	Explain layers of TCP/IP protocol suit	8	L2	CO1
	b	Explain types of packet switched networks and evaluate the delay time of both	12	L2	CO1
Q 3	a	Explain types of errors and Hamming distance. Find hamming distance between the following i) (000,011) ii) (10101,11110)	8	L3	CO2
	b	Describe the working of CRCencoder and decoder. Perform division wrt the following Dataword: 1001 Divisor: 1011	12	L3	CO2
Q 4	a	Explain Stop and wait protocol with FSM	6	L2	CO2
	b	Explain three types of frames in HDLC	8	L2	CO2
	c	Discuss controlled access protocol using reservation method	6	L2	CO2
Q 5	a	Explain the services offered by network layer	6	L2	CO2
	b	Define address space. Differentiate between classful addressing and classless addressing.	8	L2	CO2
	c	Explain Network Address Resolution(NAT)with a neat diagram	6	L2	CO3
Q 6	a	Explain Ipv6 packet format in detail.	6	L2	C02

	b	Discuss D-V routing highlighting the importance of distance vector	7	L2	C02
	c	Describe BGP protocol in detail	6	L2	C04
Q 7	a	Explain the concept of port numbers mentioning ICANN ranges	5	L3	C03
	b	Explain Go-Back-N protocol	9	L2	C03
	c	Explain TCP segment format with a neat diagram	6	L3	C03
Q 8	a	Describe the connection establishment in TCP	8	L2	C03
	b	Explain error control in TCP using acknowledgements	4	L2	C03
	c	Discuss three algorithms for handling congestion in TCP	8	L2	C03
Q 9	a	Discuss Application layer paradigms with neat diagram	5	L2	C03
	b	Explain the use of sockets in process-to-process communication	7	L2	C03
	c	Discuss the connection types in HTTP along with the formats of ,messages	8	L2	C03
Q 10	a	Explain POP and IMAP protocols	8	L2	C04
	b	Discuss the applications of SSH protocol	4	L2	C04
	c	Explain resolution in DNS	8	L2	C03

Q 1 a) Explain fundamental characteristics and data representation in data communication

Ans- Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable. For data communications to occur, the communicating devices must be part of a communication system made up of a com bination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics: deliv ery, accuracy, timeliness, and jitter. 1. Delivery. The system must deliver data to the correct destination. Data must be received by the intended device or user and only by that device or user. 2. Accuracy.

The system must deliver the data accurately. Data that have been altered in transmission and left uncorrected are unusable. 3. Timeliness. The system must deliver data in a timely manner. Data delivered late are useless. In the case of video and audio, timely delivery means delivering data as they are produced, in the same order that they are produced, and without signifi cant delay. This kind of delivery is called real-time transmission. 4. Jitter. Jitter refers to the variation in the packet arrival time. It is the uneven delay in the delivery of audio or video packets. For example, let us assume that

video packets are sent every 30 ms. If some of the packets arrive with 30-ms delay and others with 40-ms delay, an uneven quality in the video is the result.

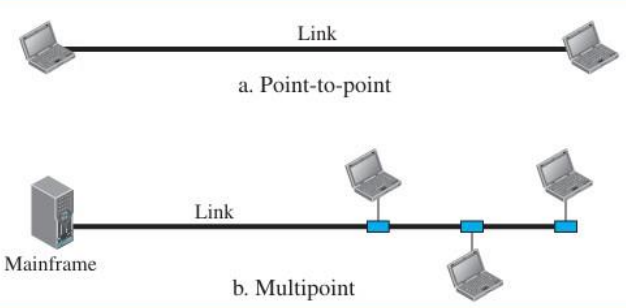
Data Representation

Information today comes in different forms such as text, numbers, images, audio, and video. Text In data communications, text is represented as a bit pattern, a sequence of bits (0s or 1s). Different sets of bit patterns have been designed to represent text symbols. Each set is called a code, and the process of representing symbols is called coding. Today, the prevalent coding system is called Unicode, which uses 32 bits to represent a symbol or character used in any language in the world. The American Standard Code for Infor mation Interchange (ASCII), developed some decades ago in the United States, now constitutes the first 127 characters in Unicode and is also referred to as Basic Latin. Appendix A includes part of the Unicode. Numbers Numbers are also represented by bit patterns. However, a code such as ASCII is not used to represent numbers; the number is directly converted to a binary number to simplify mathematical operations. Appendix B discusses several different numbering systems. Images Images are also represented by bit patterns. In its simplest form, an image is composed of a matrix of pixels (picture elements), where each pixel is a small dot. The size of the pixel depends on the resolution. For example, an image can be divided into 1000 pixels or 10,000 pixels. In the second case, there is a better representation of the image (better resolution), but more memory is needed to store the image.

Audio Audio refers to the recording or broadcasting of sound or music. Audio is by nature different from text, numbers, or images. It is continuous, not discrete. Even when we use a microphone to change voice or music to an electric signal, we create a continuous signal. We will learn more about audio in Chapter 26. Video Video refers to the recording or broadcasting of a picture or movie. Video can either be produced as a continuous entity (e.g., by a TV camera), or it can be a combination of images, each a discrete entity, arranged to convey the idea of motion.

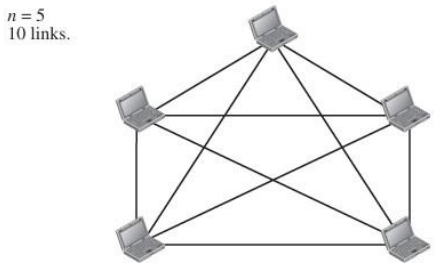
Q 1 b) Discuss types of connection and basic topologies in the network

Ans- Type of Connection A network is two or more devices connected through links. A link is a communications pathway that transfers data from one device to another. For visualization purposes, it is simplest to imagine any link as a line drawn between two points. For communication to occur, two devices must be connected in some way to the same link at the same time. There are two possible types of connections: point-to-point and multipoint. Point-to-Point A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices. Most point-to-point connections use an actual length of wire or cable to connect the two ends, but other options, such as microwave or satellite links, are also possible as shown in Figure . When we change television channels by infrared remote control, we are establishing a point-to-point connection between the remote control and the television's control system. Multipoint A multipoint (also called multidrop) connection is one in which more than two spe cific devices share a single link



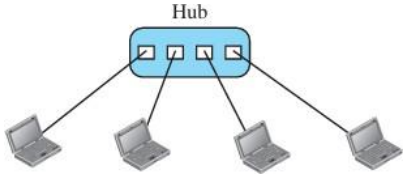
In a multipoint environment, the capacity of the channel is shared, either spatially or temporally. If several devices can use the link simultaneously, it is a spatially shared connection. If users must take turns, it is a timeshared connection. Physical Topology The term physical topology refers to the way

in which a network is laid out physically. Two or more devices connect to a link; two or more links form a topology. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another. There are four basic topologies possible: mesh, star, bus, and ring. Mesh Topology In a mesh topology, every device has a dedicated point-to-point link to every other device. The term dedicated means that the link carries traffic only between the two devices it connects. To find the number of physical links in a fully connected mesh net work with n nodes, we first consider that each node must be connected to every other node. Node 1 must be connected to n – 1 nodes, node 2 must be connected to n – 1 nodes, and finally node n must be connected to n – 1 nodes. We need n (n – 1) physical links. However, if each physical link allows communication in both directions (duplex mode), we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need n (n – 1) / 2 duplex-mode links. To accommodate that many links, every device on the network must have n – 1 input/output (I/O) ports (see Figure 1.4) to be connected to the other n – 1 stations.



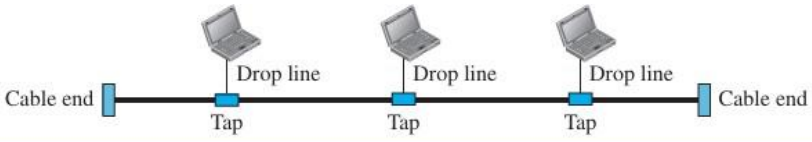
Star Topology:

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



Bus Topology :

The preceding examples all describe point-to-point connections. A bus topology, on the other hand, is multipoint. One long cable acts as a backbone to link all the devices in a network

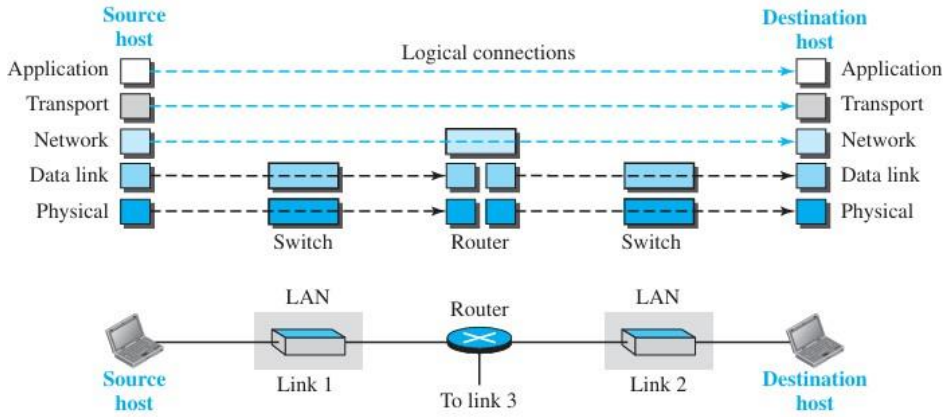


Q 1 c) Explain packet switching and circuit switching with neat diagram

Ans-

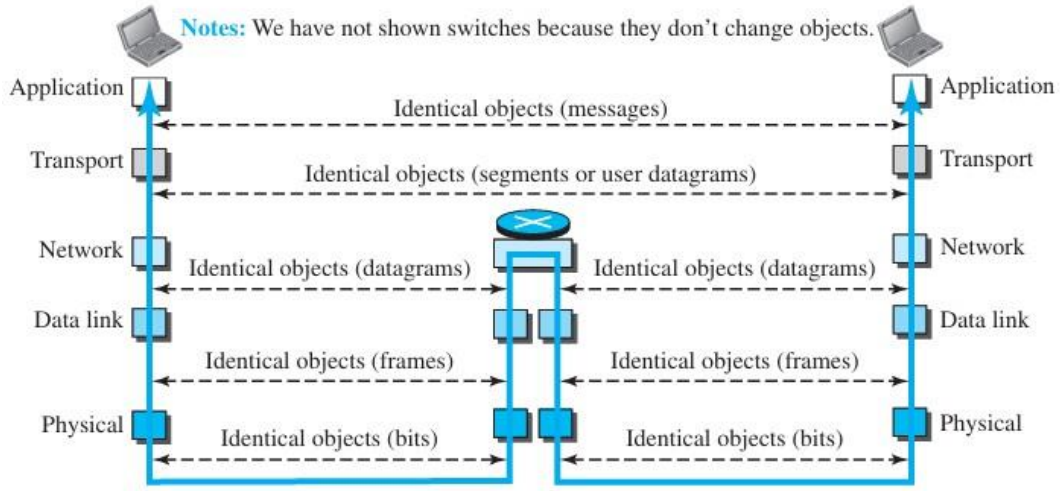
Q 2 a) Explain layers of TCP/IP protocol suit

Ans- Layers in the TCP/IP Protocol Suite After the above introduction, we briefly discuss the functions and duties of layers in the TCP/IP protocol suite. Each layer is discussed in detail in the next five parts of the book. To better understand the duties of each layer, we need to think about the logical connections between layers. Figure 2.6 shows logical connections in our simple internet.



Using logical connections makes it easier for us to think about the duty of each layer. As the figure shows, the duty of the application, transport, and network layers is end-to-end. However, the duty of the data-link and physical layers is hop-to-hop, in which a hop is a host or router. In other words, the domain of duty of the top three layers is the internet, and the domain of duty of the two lower layers is the link. Another way of thinking of the logical connections is to think about the data unit created from each layer. In the top three layers, the data unit (packets) should not be changed by any router or link-layer switch. In the bottom two layers, the packet created by the host is changed only by the routers, not by the link-layer switches. Figure 2.7 shows the second principle discussed previously for protocol layering. We show the identical objects below each layer related to each device.

Figure 2.7 *Identical objects in the TCP/IP protocol suite*



Note that, although the logical connection at the network layer is between the two hosts, we can only say that identical objects exist between two hops in this case because a router may fragment the packet at the network layer and send more packets than received. Note that the link between two hops does not change the object.

Q 2 b) Explain types of packet switched networks and evaluate the delay time of both

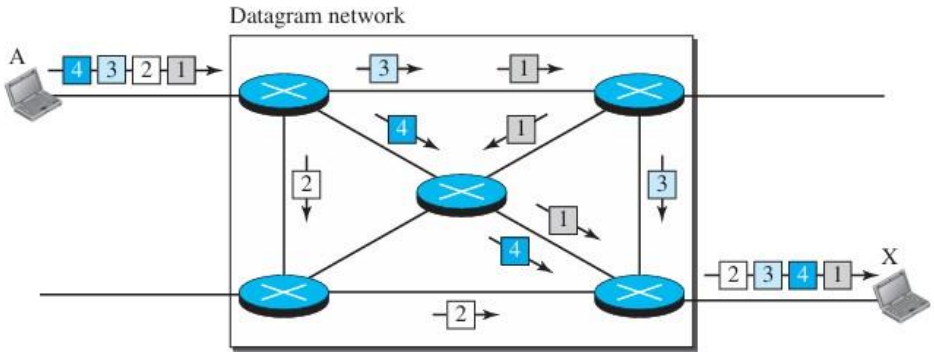
Ans- Packet-switched networks transmit data by **breaking messages into packets**, which are routed independently through the network. There are **two main types** of packet-switched networks.

Datagram Packet-Switched Network (Connectionless)

In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multipacket transmission, the network treats it as though it existed alone. Packets in this approach

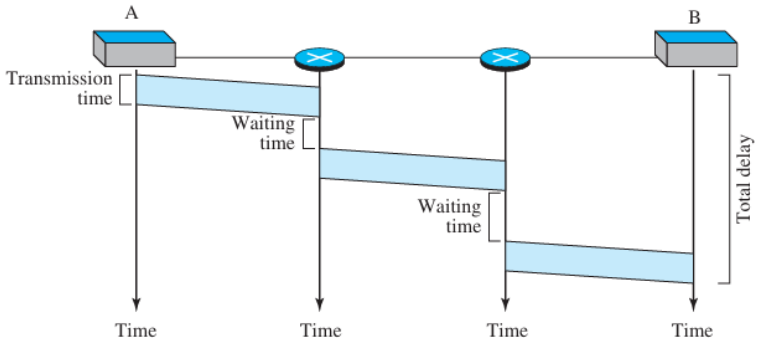
are referred to as datagrams. Datagram switching is normally done at the network layer. We briefly discuss datagram networks here as a comparison with circuit-switched and virtual-circuit switched networks.

Figure 8.7 *A datagram network with four switches (routers)*



In this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application. The datagram networks are sometimes referred to as connectionless networks. The term connectionless here means that the switch (packet switch) does not keep information about the connection state. There are no setup or teardown phases. Each packet is treated the same by a switch regardless of its source or destination. Delay There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message. Figure 8.9 gives an example of delay in a datagram network for one packet.

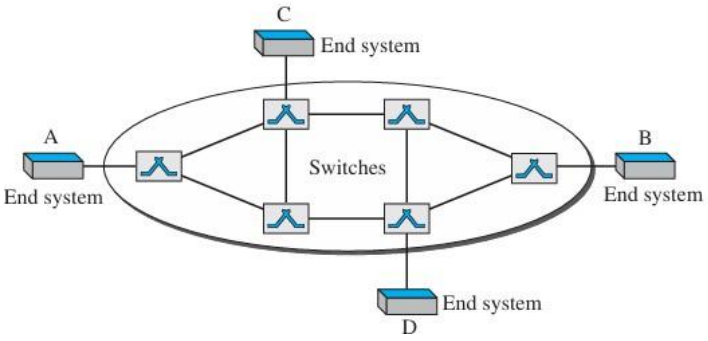
Figure 8.9 *Delay in a datagram network*



The packet travels through two switches. There are three transmission times (3T), three propagation delays (slopes 3τ of the lines), and two waiting times (w1 + w2). We ignore the processing time in each switch. The total delay is Total delay 5 3T 1 3τ 1 w1 1 w2 8.3.2 Virtual-Circuit Networks A virtual-circuit network is a cross between a circuit-switched network and a datagram network. It has some characteristics of both. 1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase. 2. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network. 3. As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header

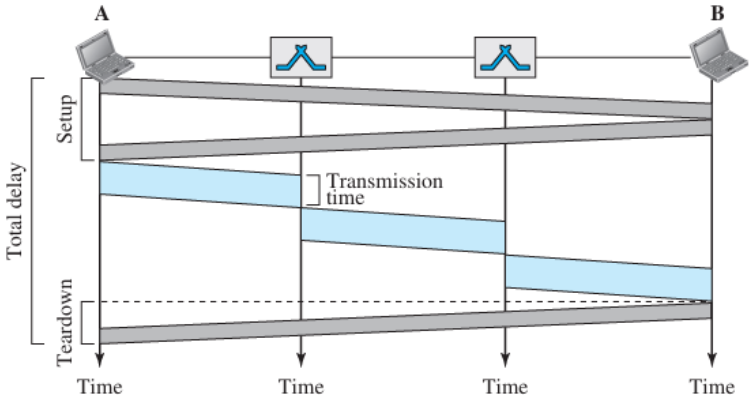
has local jurisdiction (it defines what the next switch should be and the channel on which the packet is being carried), not end-to-end jurisdiction. The reader may ask how the intermediate switches know where to send the packet if there is no final destination address carried by a packet. The answer will be clear when we discuss virtual-circuit identifiers in the next section. 4. As in a circuit-switched network, all packets follow the same path established during the connection
5. A virtual-circuit network is normally implemented in the data-link layer, while a circuit-switched network is implemented in the physical layer and a datagram net work in the network layer. But this may change in the future. Figure 8.10 is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.

Figure 8.10 *Virtual-circuit network*



Delay in Virtual-Circuit Networks In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets. Figure 8.16 shows the delay for a packet traveling through two switches in a virtual-circuit network.

Figure 8.16 *Delay in a virtual-circuit network*



The packet is traveling through two switches (routers). There are three transmis sion times (3T), three propagation times (3τ), data transfer depicted by the sloping lines, a setup delay (which includes transmission and propagation in two directions), and a teardown delay (which includes transmission and propagation in one direction). We ignore the processing time in each switch. The total delay time is
Total delay + 3T + 3τ + setup delay + teardown delay. Circuit-Switched Technology in WANs As we will see in Chapter 14, virtual-circuit networks are used in switched WANs such as ATM networks. The data-link layer of these technologies is well suited to the virtual circuit technology. Switching at the data-link layer in a switched WAN is normally implemented by using virtual-circuit

techniques.

Q 3 a) Explain types of errors and Hamming distance. Find hamming distance between the following
i) (000,011)
ii) (10101,11110

Ans- # 1. Types of Errors in Data Communication

During data transmission, errors may occur due to noise, interference, or signal distortion. The main types are:

a) Single-bit Error

- Only **one bit** in the data unit is changed.
- Example:
  Sent: `101011`
  Received: `101001`

b) Multiple-bit Error

- **More than one bit** is changed, but not necessarily adjacent.
- Example:
  Sent: `110101`
  Received: `100001`

c) Burst Error

- **Two or more consecutive bits** are changed.
- Most common type of error in data transmission.
- Example:
  Sent: `111000111`
  Received: `111111011`

000
011

Differences at positions: 2, 3 so Hamming distance=2

10101
11110

Differences at positions: 2, 4, 5 so Hamming distance=3

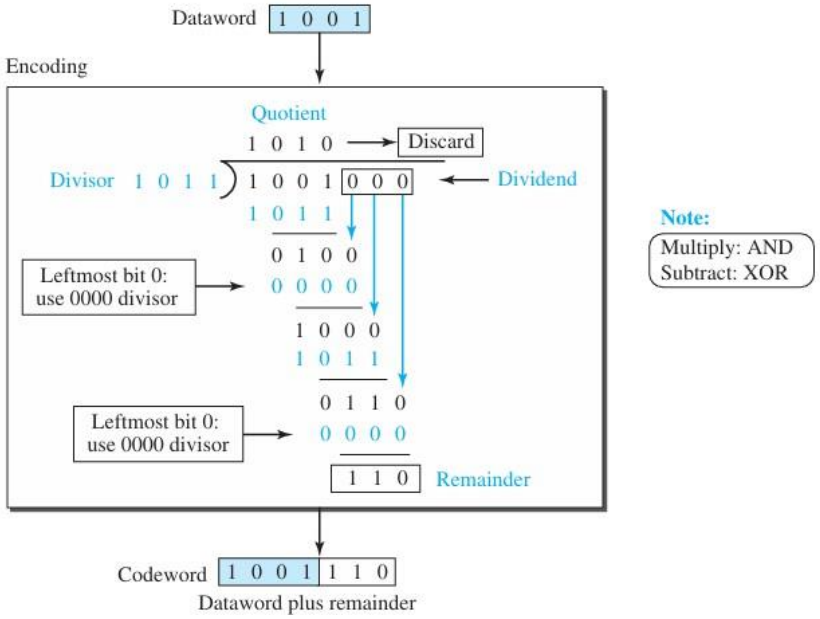
Q 3 b) Describe the working of CRCencoder and decoder. Perform division wrt the following

Dataword: 1001

Divisor: 1011

Ans-In the encoder, the dataword has k bits (4 here); the codeword has n bits (7 here). The size of the dataword is augmented by adding n − k (3 here) 0s to the right-hand side of the word. The n-bit result is fed into the generator. The generator uses a divisor of size n − k + 1 (4 here), predefined and agreed upon. The generator divides the augmented dataword by the divisor (modulo-2 division). The quotient of the division is discarded; the remainder (r2r1r0) is appended to the dataword to create the codeword. The decoder receives the codeword (possibly corrupted in transition). A copy of all n bits is fed to the checker, which is a replica of the generator. The remainder produced by the checker is a syndrome of n − k (3 here) bits, which is fed to the decision logic analyzer. The analyzer has a simple function. If the syndrome bits are all 0s, the 4 left most bits of the codeword are accepted as the dataword (interpreted as no error); other wise, the 4 bits are discarded (error). Encoder Let us take a closer look at the encoder. The encoder takes a dataword and augments it with n − k number of 0s. It then divides the augmented dataword by the divisor, as shown in Figure 10.6.

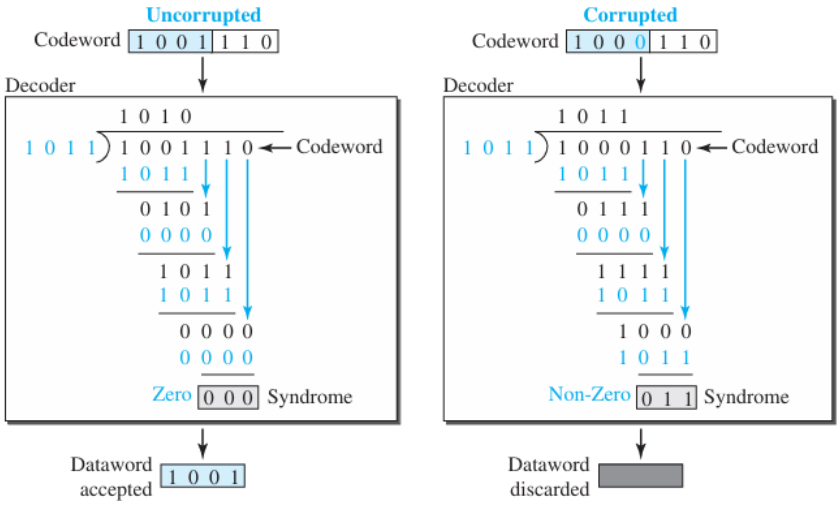
Figure 10.6 *Division in CRC encoder*



Decoder

The codeword can change during transmission. The decoder does the same division process as the encoder. The remainder of the division is the syndrome. If the syndrome is all 0s, there is no error with a high probability; the dataword is sepa rated from the received codeword and accepted. Otherwise, everything is discarded. Figure 10.7 shows two cases: The left-hand figure shows the value of the syndrome when no error has occurred; the syndrome is 000. The right-hand part of the figure shows the case in which there is a single error. The syndrome is not all 0s (it is 011).

Figure 10.7 *Division in the CRC decoder for two cases*

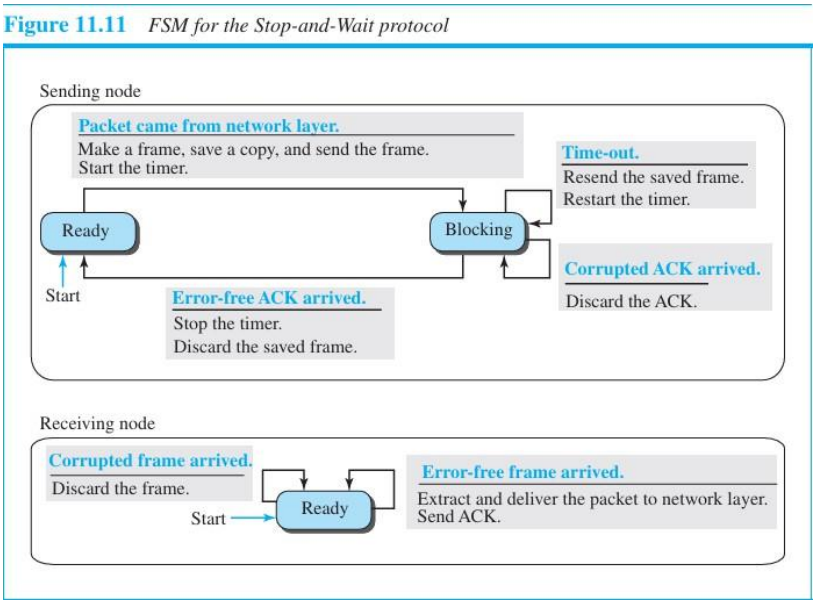


Q 4 a) Explain Stop and wait protocol with FSM

Ans- Sender States The sender is initially in the ready state, but it can move between the ready and block ing state.

Ready State. When the sender is in this state, it is only waiting for a packet from the network layer. If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame. The sender then moves to the blocking state. ❑ Blocking State. When the sender is in this state, three events can occur: a. If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer. b. If a corrupted ACK arrives, it is discarded. c. If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state. Receiver The receiver is always in the ready state. Two events may occur: a. If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent. b. If a corrupted frame arrives, the frame is discarded.

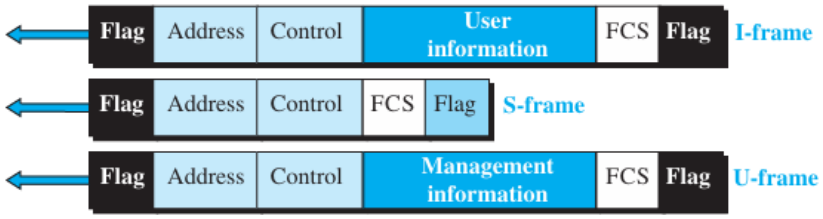
Figure 11.11 *FSM for the Stop-and-Wait protocol*



Q 4 b) Explain three types of frames in HDLC

Ans- Framing To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames: information frames (I-frames), supervisory frames (S-frames), and unnumbered frames (U-frames). Each type of frame serves as an envelope for the transmission of a different type of message. I frames are used to data-link user data and control information relating to user data (piggy backing). S-frames are used only to transport control information. U-frames are reserved for system management. Information carried by U-frames is intended for managing the link itself. Each frame in HDLC may contain up to six fields, as shown in Figure 11.16: a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field. In multiple-frame transmissions, the end ing flag of one frame can serve as the beginning flag of the next frame.

Figure 11.16 *HDLC frames*



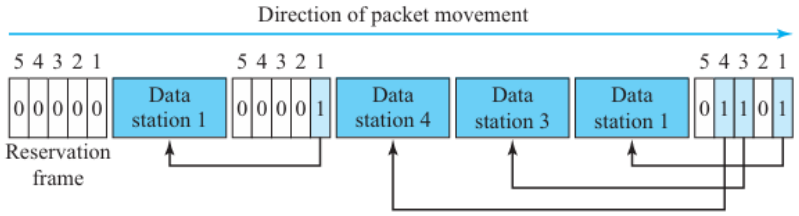
Flag field. This field contains synchronization pattern 01111110, which identifies both the beginning and the end of a frame. ❑ Address field. This field contains the address of the secondary station. If a primary station created the frame, it contains a to address. If a secondary station creates the frame, it contains a from address. The address field can be one byte or several bytes long, depending on the needs of the network.

Control field. The control field is one or two bytes used for flow and error control. The interpretation of bits are discussed later. ❑ Information field. The information field contains the user's data from the network layer or management information. Its length can vary from one network to another. ❑ FCS field. The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte CRC.

Q 4 c) Discuss controlled access protocol using reservation method

Ans- In controlled access, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We dis cuss three controlled-access methods. 12.2.1 Reservation In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval. If there are N stations in the system, there are exactly N reservation minislots in the reservation frame. Each minislot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minislot. The stations that have made res ervations can send their data frames after the reservation frame. Figure 12.18 shows a situation with five stations and a five-minislot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the sec ond interval, only station 1 has made a reservation.

Figure 12.18 *Reservation access method*

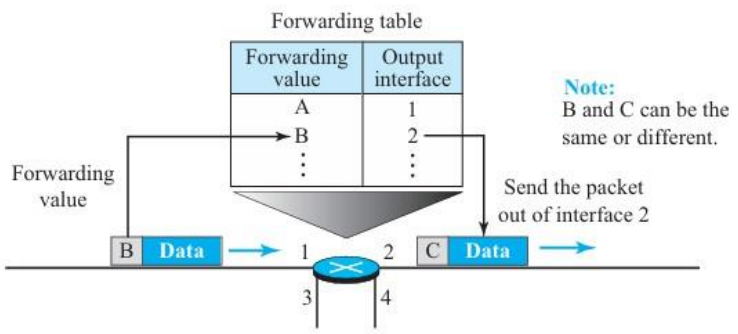


Q 5 a) Explain the services offered by network layer

Ans- Packetizing The first duty of the network layer is definitely packetizing: encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulat ing the payload from the network-layer packet at the destination. In other words, one duty of the network layer is to carry a payload from the source to the destination without changing it or using it. The network layer is doing the service of a carrier such as the postal office, which is responsible for delivery of packages from a sender to a receiver without changing or using the contents.

Routing and Forwarding Other duties of the network layer, which are as important as the first, are routing and forwarding, which are directly related to each other. Routing The network layer is responsible for routing the packet from its source to the destination. A physical network is a combination of networks (LANs and WANs) and routers that connect them. This means that there is more than one route from the source to the destination. The network layer is responsible for finding the best one among these possible routes. The network layer needs to have some specific strategies for defining the best route. In the Internet today, this is done by running some routing protocols to help the routers coordinate their knowledge about the neighborhood and to come up with consistent tables to be used when a packet arrives. The routing protocols, which we discuss in Chapters 20 and 21, should be run before any communication occurs. Forwarding If routing is applying strategies and running some routing protocols to create the decision-making tables for each router, forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces. The decision-making table a router normally uses for applying this action is sometimes called the forwarding table and sometimes the routing table. When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (in unicast routing) or to some attached networks (in multicast routing). To make this decision, the router uses a piece of information in the packet header, which can be the destination address or a label, to find the corresponding output interface number in the forwarding table. Figure 18.2 shows the idea of the forwarding process in a router.

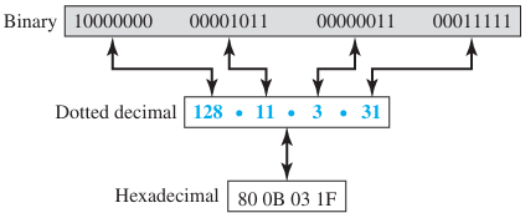
Figure 18.2 *Forwarding process*



Q 5 b) Define address space. Differentiate between classful addressing and classless addressing.

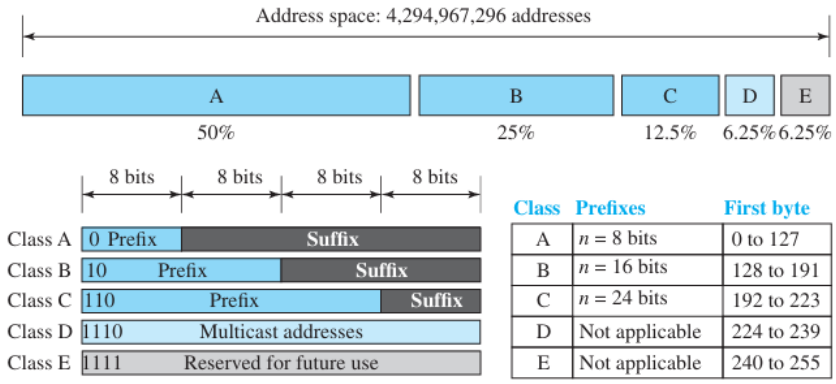
Ans- Address Space A protocol like IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. If a protocol uses b bits to define an address, the address space is 2b because each bit can have two different values (0 or 1). IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than four billion). If there were no restrictions, more than 4 billion devices could be connected to the Internet. Notation There are three common notations to show an IPv4 address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16). In binary notation, an IPv4 address is displayed as 32 bits. To make the address more readable, one or more spaces are usually inserted between each octet (8 bits). Each octet is often referred to as a byte. To make the IPv4 address more compact and easier to read, it is usually written in decimal form with a decimal point (dot) separating the bytes. This format is referred to as dotted-decimal notation. Note that because each byte (octet) is only 8 bits, each number in the dotted-decimal notation is between 0 and 255.

Figure 18.16 *Three different notations in IPv4 addressing*



Classful Addressing When the Internet started, an IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24). The whole address space was divided into five classes (class A, B, C, D, and E), as shown in Figure 18.18. This scheme is referred to as classful addressing. Although classful addressing belongs to the past, it helps us to understand classless addressing, discussed later. In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can have only seven bits as the network identifier. This means there are only 27 = 128 networks in the world that can have a class A address. In class B, the network length is 16 bits, but since the first two bits, which are (10)2, define the class, we can have only 14 bits as the network identifier. This means there are only 214 = 16,384 networks in the world that can have a class B address. All addresses that start with (110)2 belong to class C. In class C, the network length is 24 bits, but since three bits define the class, we can have only 21 bits as the network identifier. This means there are 221 = 2,097,152 networks in the world that can have a class C address. Class D is not divided into prefix and suffix. It is used for multicast addresses. All addresses that start with 1111 in binary belong to class E. As in Class D, Class E is not divided into prefix and suffix and is used as reserve.

Figure 18.18 *Occupation of the address space in classful addressing*



Class	Prefixes	First byte
A	n = 8 bits	0 to 127
B	n = 16 bits	128 to 191
C	n = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

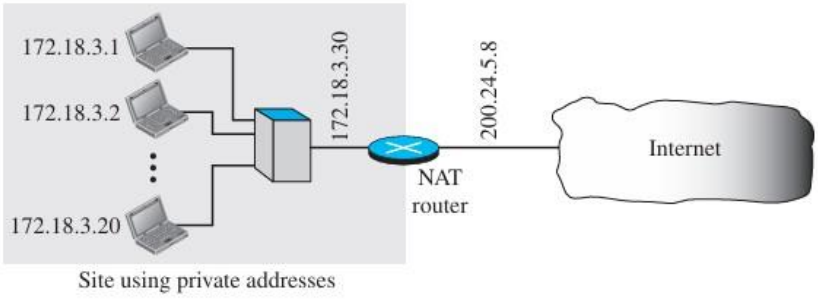
Classless Addressing Subnetting and supernetting in classful addressing did not really solve the address depletion problem. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution. The larger address space, however, requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed. Although the long-range solution has already been devised and is called IPv6 (discussed later), a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization. The short-term solution still uses IPv4 addresses, but it is called classless addressing. In other words, the class privilege was removed from the distribution to compensate for the address depletion. There was another motivation for classless addressing. During the 1990s, Internet Service Providers (ISPs) came into prominence. An ISP is an organization that pro vides Internet access for individuals, small businesses, and midsize organizations that do not want to create an Internet site and become involved in providing Internet ser vices (such as electronic mail) for their employees. An ISP can provide these services. An ISP is granted a large range of addresses and then subdivides the addresses (in groups of 1, 2, 4, 8, 16, and so on), giving a range of addresses to a household or a small business. The customers are connected via a dial-up modem, DSL, or cable modem to the ISP. However, each customer needs some IPv4 addresses. In 1996, the Internet authorities announced a new architecture called classless addressing. In classless addressing, variable-length blocks are used that belong to no classes. We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on. In classless addressing, the whole address space is divided into variable length blocks. The prefix in an address defines the block (network); the suffix defines the node (device). Theoretically, we can have a block of 20, 21, 22, ...,232 addresses. One of the restrictions, as we discuss later, is that the number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses. Figure 18.19 shows the division of the whole address space into nonoverlapping blocks.Unlike classful addressing, the prefix length in classless addressing is variable. We can have a prefix length that ranges from 0 to 32. The size of the network is inversely proportional to the length of the prefix. A small prefix means a larger network; a large prefix means a smaller network.

Figure 18.19 *Variable-length blocks in classless addressing*

Q 5 c) Explain Network Address Resolution(NAT)with a neat diagram

Ans- Network Address Resolution (NAT) The distribution of addresses through ISPs has created a new problem. Assume that an ISP has granted a small range of addresses to a small business or a household. If the business grows or the household needs a larger range, the ISP may not be able to grant the demand because the addresses before and after the range may have already been allocated to other networks. In most situations, however, only a portion of computers in a small network need access to the Internet simultaneously. This means that the number of allocated addresses does not have to match the number of computers in the network. For example, assume that in a small business with 20 computers the maximum number of computers that access the Internet simultaneously is only 4. Most of the computers are either doing some task that does not need Internet access or communicating with each other. This small business can use the TCP/IP protocol for both internal and uni versal communication. The business can use 20 (or 25) addresses from the private block addresses (discussed before) for internal communication; five addresses for uni versal communication can be assigned by the ISP. A technology that can provide the mapping between the private and universal addresses, and at the same time support virtual private networks, which we discuss in Chapter 32, is Network Address Translation (NAT). The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world. The site must have only one connection to the global Internet through a NAT-capable router that runs NAT software. Figure 18.29 shows a simple implementation of NAT.

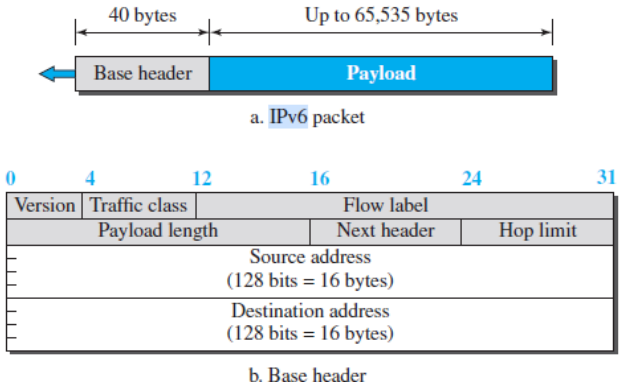
Figure 18.29 *NAT*



As the figure shows, the private network uses private addresses. The router that connects the network to the global address uses one private address and one global address. The private network is invisible to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

Q 6 a) Explain IPV6 packet format in detail

Figure 22.6 *IPv6 datagram*



Version. The 4-bit version field defines the version number of the IP. For IPv6, the

value is 6.

❑ Traffic class. The 8-bit traffic class field is used to distinguish different payloads ith different delivery requirements. It replaces the type-of-service field in IPv4.

❑ Flow label. The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data. We will discuss this field later.

❑ Payload length. The 2-byte payload length field defines the length of the IP datagram excluding the header. Note that IPv4 defines two fields related to the length: header length and total length. In IPv6, the length of the base header is fixed (40 bytes); only the length of the payload needs to be defined.

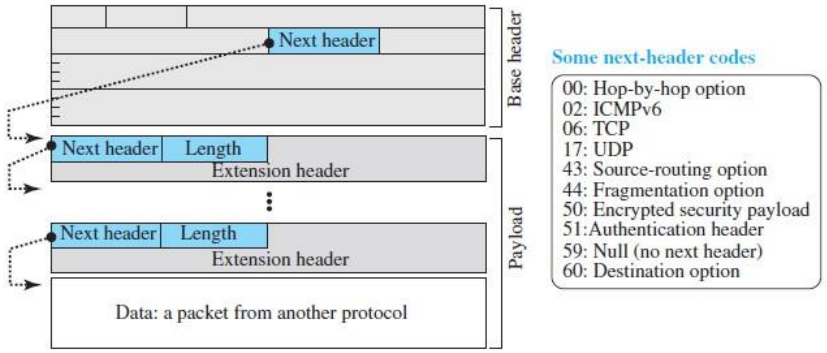
❑ Next header. The next header is an 8-bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram. This field is similar to the protocol field in IPv4, but we talk more about it when we discuss the payload.

❑ Hop limit. The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.

❑ Source and destination addresses. The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.

❑ Payload. Compared to IPv4, the payload field in IPv6 has a different format and meaning, as shown in Figure 22.7. The payload in IPv6 means a combination of zero or more extension headers(options) followed by the data from other protocols (UDP, TCP, and so on). In IPv6, options, which are part of the header in IPv4, are designed as extension headers. The payload can have as many extension headers as required by the situation. Each extension header has two mandatory fields, next header and the length, followed by information related to the particular option. Note that each next header field value (code) defines the type of the next header (hop-by-hop option, sourcerouting option, . . .); the last next header field defines the protocol (UDP, TCP, . . .)

Figure 22.7 *Payload in an IPv6 datagram*



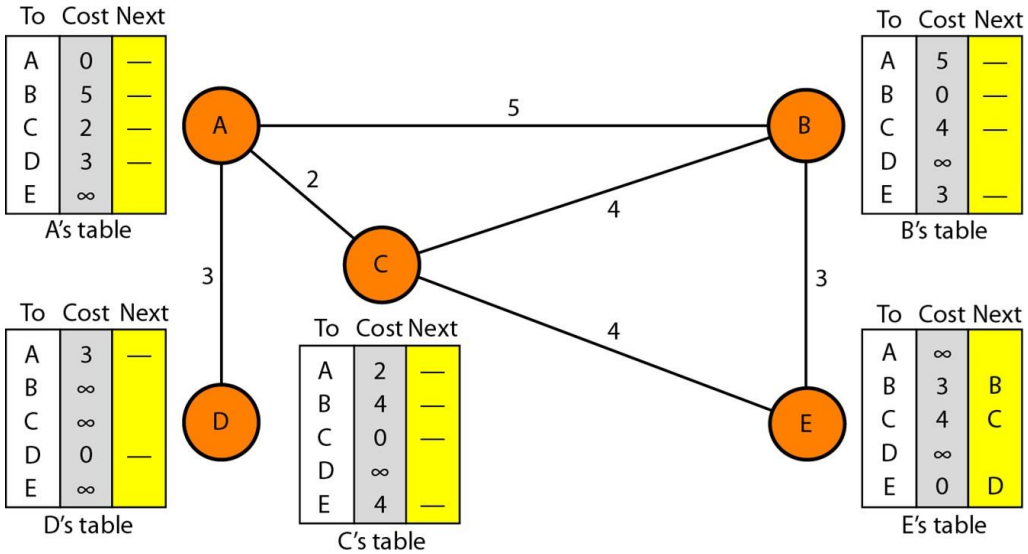
Ans-

Q 6 b) Discuss D-V routing highlighting the importance of distance vector

In this, the least-cost route between any two nodes is the route with minimum distance.

In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.

The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).



Ans-

Intialization

Each node knows how to reach any other node and the cost. At the beginning, however, this is not the case.

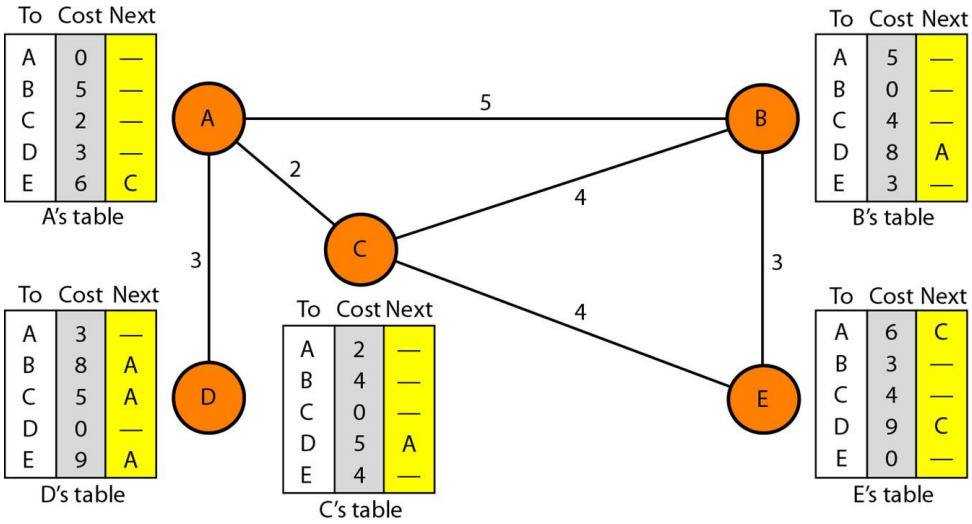
Each node can know only the distance between itself and its immediate neighbors, those directly connected to it.

So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors.

The whole idea of distance vector routing is the sharing of information between neighbors.

Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E.In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

There is only one problem.A node is not aware of a neighbor's table.he best solution for each node is to send its entire table to the neighbor and let the neighbor decide what part to use and what part to discard.



In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

a)The receiving node needs to add the cost between itself and the sending node to each value in the second column.

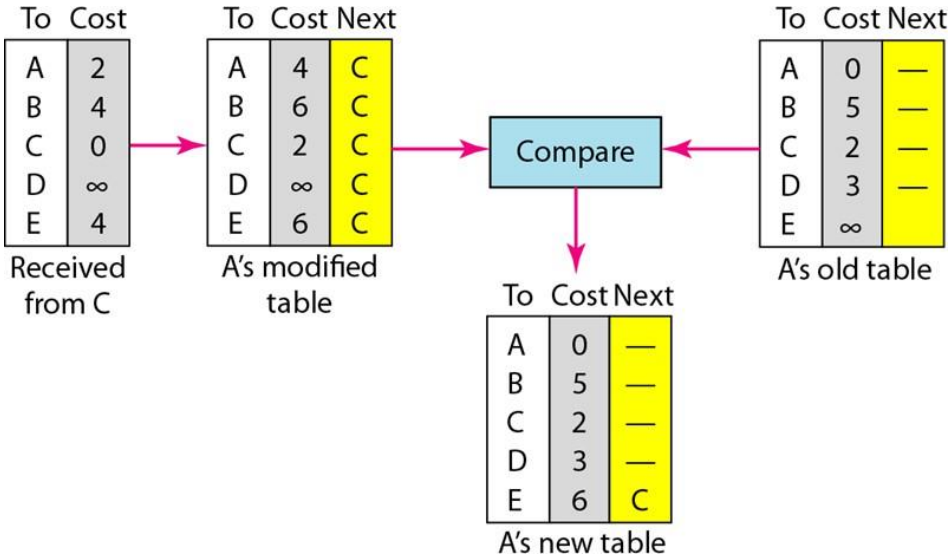
b)The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.

c) The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

If the next-node entry is the same, the receiving node chooses the new row.

For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.



The question now is, When does a node send its partial routing table (only two columns) to all its immediate neighbors?

The table is sent both periodically and when there is a change in the table.

Periodic Update A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

Triggered Update A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update.

Distance Vector routing is important because it provides a simple, distributed, and low-overhead method for routing data in computer networks, especially in small to medium-sized networks.

Key importance points:

- Simplicity of implementation
  Each router only maintains a table of distances (costs) to reach other networks, making it easy to design, configure, and maintain.

- Low computational overhead
  Routers exchange routing information only with their immediate neighbors, reducing processing and memory requirements.

- Dynamic route adaptation
  Automatically updates routes when network topology changes (link failure or recovery), ensuring continued connectivity.

- Foundation for popular protocols
  Forms the basis of widely used routing protocols like RIP (Routing Information Protocol), helping students understand core routing concepts.

- Scalability for smaller networks
  Works efficiently in small or less complex networks where frequent full-table updates are acceptable.

- Decentralized operation
  No central controller is required; each router independently computes routes based on neighbor information, improving fault tolerance.

Q 6 c) Describe BGP protocol in detail

Ans-Border Gateway Protocol (BGP) is the core routing protocol of the Internet. It is an inter-domain, path-vector routing protocol used to exchange routing information between Autonomous Systems (ASes).

---

1. Definition and Purpose

BGP enables routers in different autonomous systems (such as ISPs, enterprises, and cloud providers) to discover reachable networks and select the best path based on policies rather than shortest distance.

Key purpose:

- Exchange routing information between ASes
- Enforce routing policies
- Ensure scalable and stable Internet routing

---

2. Autonomous System (AS)

An Autonomous System is a group of IP networks under a single administrative control, identified by a unique AS Number (ASN).

Types:

- Public ASN – Used on the Internet
- Private ASN – Used within organizations

---

3. Types of BGP

a) External BGP (eBGP)

- Runs between routers in different ASes
- Used for Internet routing
- Higher administrative distance

b) Internal BGP (iBGP)

- Runs within the same AS
- Distributes external routes internally
- Requires full mesh or route reflectors

---

4. BGP Characteristics

- Path-vector protocol (stores complete AS path)
- Uses TCP port 179
- Incremental updates (only changes sent)
- Policy-based routing
- Classless (supports CIDR and VLSM)

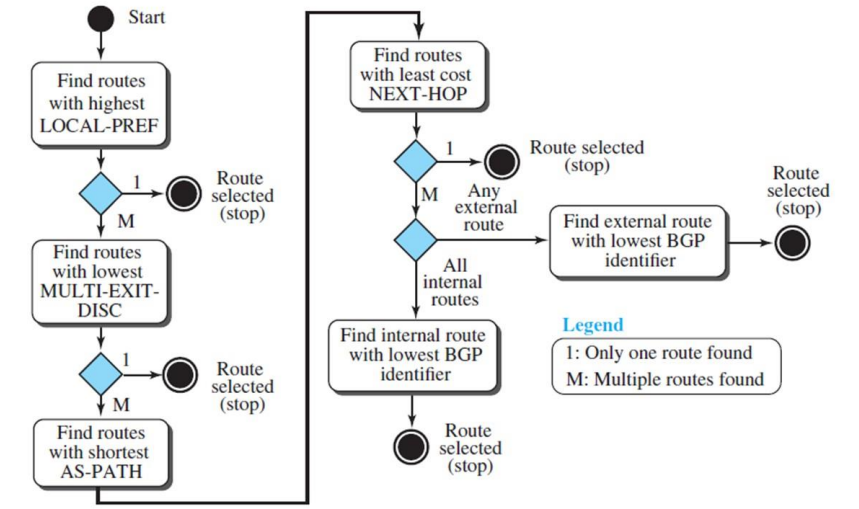
---

5. BGP Message Types

BGP uses four types of messages:

1. OPEN
   Establishes a BGP session and negotiates parameters.
2. UPDATE
   Advertises new routes or withdraws unreachable routes.
3. KEEPALIVE
   Maintains the BGP session.
4. NOTIFICATION
   Reports errors and closes the session.

Figure 20.30 *Flow diagram for route selection*



Q 7 a) Explain the concept of port numbers mentioning ICANN ranges

Ans-Port numbers are logical communication endpoints used in TCP/IP networking to identify specific applications or services running on a host. While an IP address identifies a device on a network, a port number identifies the particular process or service on that device.

Port numbers are 16-bit values, ranging from 0 to 65,535, and are used along with IP addresses to enable multiple applications to communicate simultaneously over the network.

ICANN Port Number Ranges

The Internet Corporation for Assigned Names and Numbers (ICANN) classifies port numbers into three main ranges:

1. Well-Known Ports (0–1023)
    - Reserved for standard and widely used services
    - Typically require administrative privileges
    - Examples:
        - HTTP – 80
        - HTTPS – 443
        - FTP – 21
        - SMTP – 25
        - DNS – 53

2. Registered Ports (1024–49151)
    - Assigned to specific applications by ICANN
    - Used by user-level processes
    - Examples:
        - MySQL – 3306
        - PostgreSQL – 5432
        - HTTPS (alt) – 8443
3. Dynamic / Private Ports (49152–65535)
    - Not assigned to any specific service
    - Used temporarily for client-side communication
    - Also called ephemeral ports

Q 7 b) Explain Go-Back-N Protocol

In Go-Back-N, the sender can transmit **multiple frames continuously** without waiting for an acknowledgment for each frame, up to a fixed **window size (N)**.
However, the receiver **accepts frames only in order**.

If a frame is lost or received with an error, the receiver discards that frame and all subsequent frames, and the sender retransmits the erroneous frame and all following frames.

Working Principle

1. The sender maintains a sliding window of size N.
2. Frames are numbered sequentially.
3. The sender sends up to N frames without waiting for acknowledgments.
4. The receiver sends cumulative acknowledgments.
5. If a frame is lost or damaged:
    - The receiver sends an acknowledgment for the last correctly received fram**e**
    - The sender goes back to the missing frame and retransmits all frames from that point

onward

Example: Go-Back-N ARQ

Sequence numbers: 0 to 10
Window size (N): 4
Lost frame: 2

Step-by-Step Transmission

Sender window initially:
 Frames 0, 1, 2, 3

1. Sender sends frames:
   $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$

2. Receiver actions:
   - Frame 0 received → ACK 1
   - Frame 1 received → ACK 2
   - Frame 2 is lost ✕
   - Frame 3 arrives but is discarded (out of order)

3. Receiver sends:
   - Repeated ACK 2 (last correctly received frame is 1)

Sender Reaction

4. Sender detects missing ACK for frame 2 (timeout occurs)

5. Sender goes back to frame 2 and retransmits:
   $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$

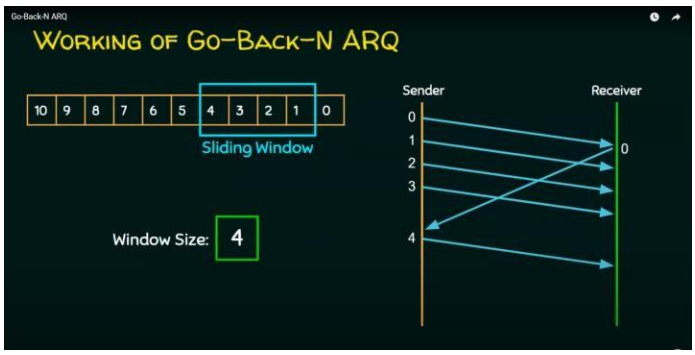
Successful Reception

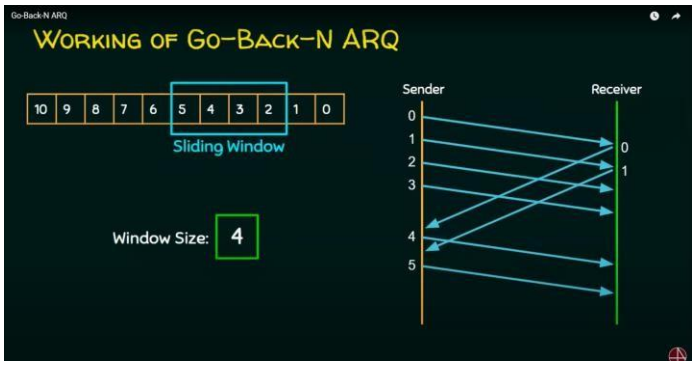
6. Receiver now receives in order:
   - Frame 2 → ACK 3
   - Frame 3 → ACK 4
   - Frame 4 → ACK 5
   - Frame 5 → ACK 6

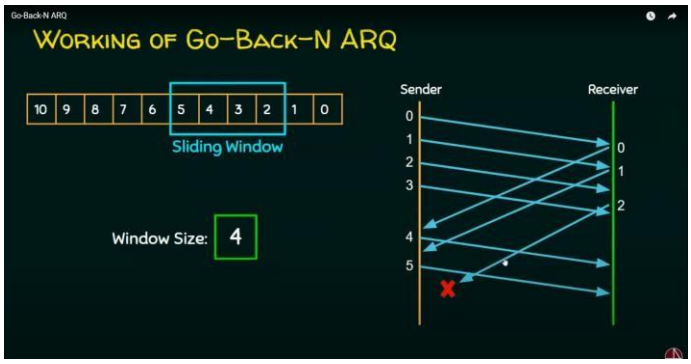

Go-Back-N ARQ
WORKING OF GO-BACK-N ARQ
10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Window Size: 4
Sender
Receiver
0
1
2
3


Go-Back-N ARQ
WORKING OF GO-BACK-N ARQ
10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Window Size: 4
Sender
Receiver
0
1
2
3
0


Go-Back-N ARQ
WORKING OF GO-BACK-N ARQ
10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Window Size: 4
Sender
Receiver
0
1
2
3
4
0


Go-Back-N ARQ
WORKING OF GO-BACK-N ARQ
10 9 8 7 6 5 4 3 2 1 0
Sliding Window
Window Size: 4
Sender
Receiver
0
1
2
3
4
0

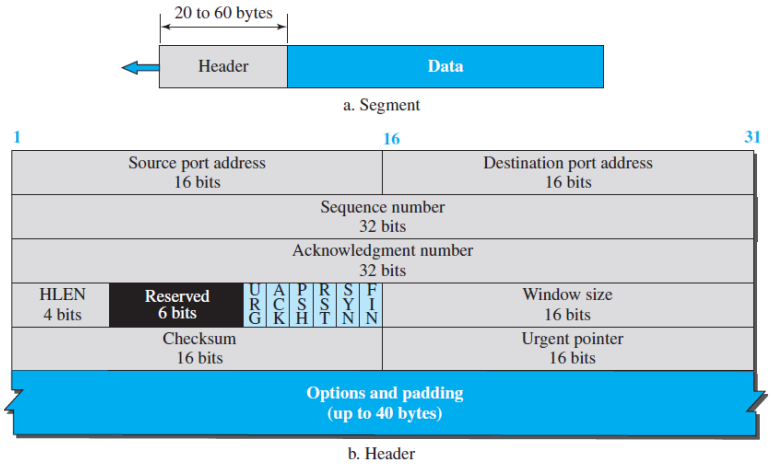




Q 7 c) Explain TCP segment format with a neat diagram

Ans-

Figure 24.7 *TCP segment format*



Source port address. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

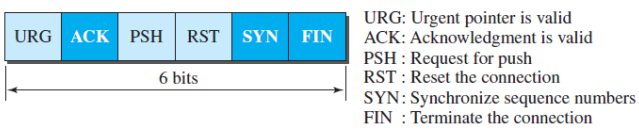
❑ Destination port address. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment. ❑ Sequence number. This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence is the first byte in the segment. During connection establishment (discussed later) each party uses a random number generator to create an initial sequence number (ISN), which is usually different in each direction.

❑ Acknowledgment number. This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number x from the other party, it returns x + 1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.

❑ Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5 × 4 = 20) and 15 (15 × 4 = 60).

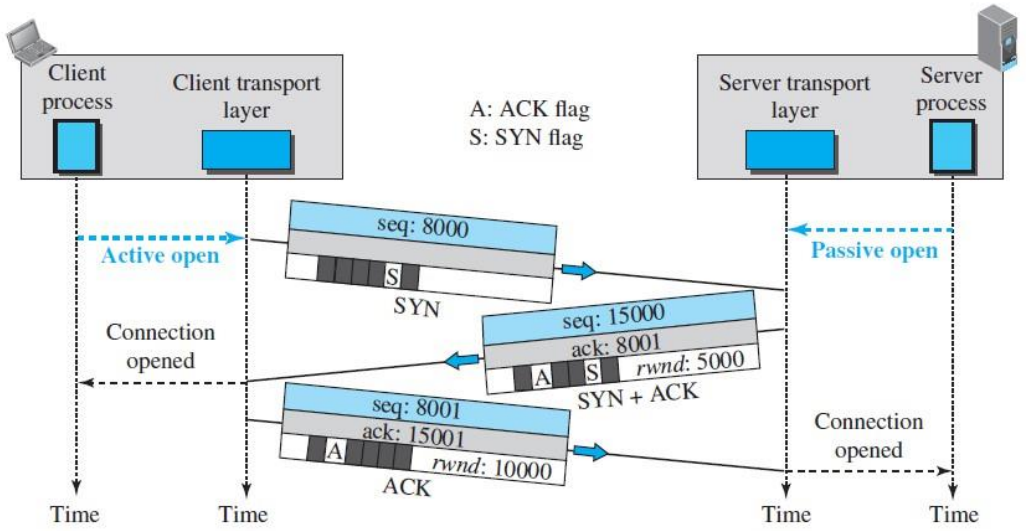
❑ Control. This field defines 6 different control bits or flags, as shown in Figure 24.8. One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP.

Figure 24.8 *Control field*



Q 8 a) Discuss the connection establishment in TCP

Figure 24.10 *Connection establishment using three-way handshaking*



Three-Way Handshaking

The connection establishment in TCP is called three-way handshaking. In our example, an application program, called the client, wants to make a connection with another application program, called the server, using TCP as the transport-layer protocol. The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a passive open. Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself. The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP to connect to a particular server. TCP can now start the three-way handshaking process, as shown in Figure 24.10.

The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers. The client in our example chooses a random number as the first sequence number and sends this number to the server. This sequence number is called the initial sequence number (ISN). Note that this segment does not contain an acknowledgment number. It does not define the window size either; a window size definition makes sense only when a segment includes an acknowledgment. Note that the SYN segment is a control segment and carries no data. However, it consumes one sequence number because it needs to be acknowledged. We can say that the SYN segment carries one imaginary byte. 2. The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. This segment has a dual purpose. First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client. The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client. Because the segment contains an acknowledgment, it also needs to define the receive window size, rwnd (to be used by the client), as we will see in the flow control section. Since this segment is playing the role of a SYN segment, it needs to be acknowledged. It, therefore, consumes one sequence number.

3. The client sends the third segment. This is just an ACK segment. It acknowledges the receipt of the second segment with the ACK flag and acknowledgment number field. Note that the ACK segment does not consume any sequence numbers if it does not carry data, but some implementations allow this third segment in the connection phase to carry the first chunk of data from the client.In this case, the segment consumes as many sequence numbers as the number of data bytes.

SYN Flooding Attack

The connection establishment procedure in TCP is susceptible to a serious security problem called SYN flooding attack. This happens when one or more malicious attackers send a large number of SYN segments to a server pretending that each of them is coming from a different client by faking the source IP addresses in the datagrams. The server, assuming that the clients are issuing an active open, allocates the necessary resources, such as creating transfer control block (TCB) tables and setting timers. The A SYN segment cannot carry data, but it consumes one sequence number. A SYN 1 ACK segment cannot carry data, but it does consume one sequence number. An ACK segment, if carrying no data, consumes no sequence number.

Ans-

Q 8 b) Explain Error control in TCP using acknowledgements

Ans- Error Control in TCP Using Acknowledgements

Transmission Control Protocol (TCP) provides reliable data transfer by detecting errors and ensuring that lost or corrupted data is retransmitted. One of the key mechanisms used for error control in TCP is acknowledgements (ACKs).

Concept of Acknowledgements in TCP

TCP uses positive acknowledgements with retransmission. This means the receiver sends an ACK to confirm successful receipt of data. If an ACK is not received within a certain time, the sender assumes an error occurred and retransmits the data.

How Error Control Works in TCP

1. Sequence Numbers
    - Each byte of data is assigned a sequence number.
    - Sequence numbers help detect missing, duplicate, or out-of-order segments.
2. Acknowledgements (ACKs)
    - TCP uses cumulative acknowledgements.
    - An ACK number indicates the next byte expected by the receiver.

    - Example: ACK = 1001 means all bytes up to 1000 have been received correctly.
3. Error Detection
    - TCP uses a checksum to detect corrupted segments.
    - If a segment is corrupted, it is discarded, and no ACK is sent.
4. Retransmission Mechanism
    - The sender starts a timer after transmitting a segment.
    - If the ACK is not received before the timer expires, the sender retransmits the segment.
5. Duplicate ACKs
    - When a segment is lost, the receiver repeatedly sends the same ACK for the last correctly received byte.
    - After receiving three duplicate ACKs, the sender performs fast retransmission without waiting for timeout.
6. Selective Acknowledgement (SACK) (optional extension)
    - Allows the receiver to inform the sender about non-contiguous blocks of data received.
    - Improves efficiency by reducing unnecessary retransmissions.

Example

- Sender sends segments with sequence numbers 1–1000 and 1001–2000.
- If the second segment is lost:
    - Receiver sends ACK = 1001 repeatedly.
    - Sender retransmits the lost segment after duplicate ACKs or timeout.

Importance of Acknowledgement-Based Error Control

- Ensures reliable, ordered, and error-free delivery
- Detects and recovers from packet loss
- Improves performance through fast retransmission

Q 8 c) Discuss three algorithms for handling congestion in TCP

Ans-

Three Algorithms for Handling Congestion in TCP

TCP uses congestion control algorithms to prevent overloading the network and to ensure fair and efficient data transmission. The three main algorithms are Slow Start, Congestion Avoidance, and Fast Recovery.

---

1. Slow Start Algorithm

Purpose:
To gradually probe the available network capacity when a connection begins or after a timeout.

Working:

TCP starts with a small congestion window (cwnd), usually 1 MSS.

For every ACK received, cwnd increases exponentially (roughly doubles every RTT).

Growth continues until:

A packet loss occurs, or

cwnd reaches the slow start threshold (ssthresh).

Significance:

Prevents sudden congestion

Quickly discovers available bandwidth

---

2. Congestion Avoidance Algorithm

Purpose:
To avoid congestion once the network capacity is nearly reached.

Working:

Activated when cwnd ≥ ssthresh.

cwnd increases linearly (additive increase).

Typically, cwnd increases by 1 MSS per RTT.

If congestion is detected (packet loss):

ssthresh is reduced

cwnd is adjusted accordingly

Significance:

Maintains network stability

Prevents congestion collapse

---

3. Fast Recovery Algorithm

Purpose:
To recover quickly from packet loss without reducing throughput drastically.

Working:

Triggered after receiving three duplicate ACKs.

TCP assumes a segment is lost but the network is still active.

Actions taken:

ssthresh = cwnd / 2

cwnd is reduced but not reset to 1

Lost segment is retransmitted immediately (fast retransmit)

After recovery, TCP enters congestion avoidance.

Significance:

Avoids unnecessary slow start

Improves performance during minor congestion

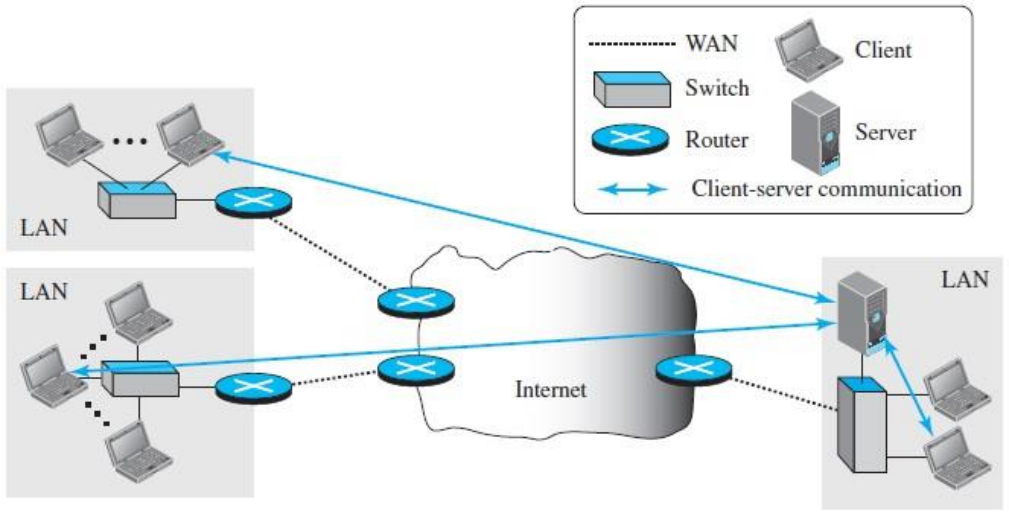
Q 9 a) Discuss application layer paradigms with neat diagrams

Ans-

Q 9 b) Explain Traditional Paradigm: Client-Server

The traditional paradigm is called the client-server paradigm. It was the most popular paradigm until a few years ago. In this paradigm, the service provider is an application program, called the server process; it runs continuously, waiting for another application program, called the client process, to make a connection through the Internet and ask for service. There are normally some server processes that can provide a specific type of service, but there are many clients that request service from any of these server processes. The server process must be running all the time; the client process is started when the client needs to receive service. The client-server paradigm is similar to some available services out of the territory of the Internet. For example, a telephone directory center in any area can be thought of as a server; a subscriber that calls and asks for a specific telephone number can be thought of as a client. The directory center must be ready and available all the time; the subscriber can call the center for a short period when the service is needed. Although the communication in the client-server paradigm is between two application programs, the role of each program is totally different. In other words, we cannot the use of sockets in process to process communication run a client program as a server program or vice versa. Later in this chapter, when we talk about client-server programming in this paradigm, we show that we always need to write two application programs for each type of service. Figure 25.2 shows an example of a client-server communication in which three clients communicate with one server to receive the services provided by this server. One problem with this paradigm is that the concentration of the communication load is on the shoulder of the server, which means the server should be a powerful computer. Even a powerful computer may become overwhelmed if a large number of clients try to connect to the server at the same time. Another problem is that there should be a service provider willing to accept the cost and create a powerful server for a specific service, which means the service must always return some type of income for the server in order to encourage such an arrangement. Several traditional services are still using this paradigm, including the World Wide Web (WWW) and its vehicle HyperText Transfer Protocol (HTTP), file transfer protocol (FTP), secure shell (SSH), e-mail, and so on.

Figure 25.2 *Example of a client-server paradigm*

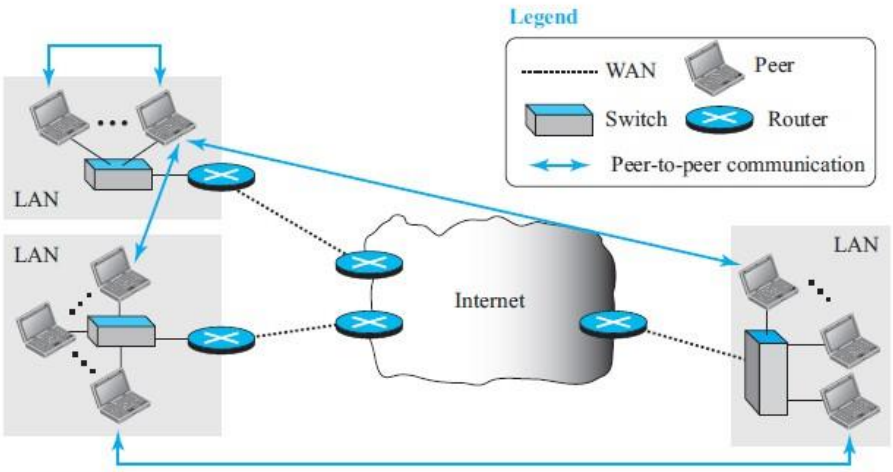


New Paradigm: Peer-to-Peer

A new paradigm, called the peer-to-peer paradigm (often abbreviated P2P paradigm) has emerged to respond to the needs of some new applications. In this paradigm, there is no need for a server process to be running all the time and waiting for the client processes to connect. The responsibility is shared between peers. A computer connected to the Internet can provide service at one time and receive service at another time. A computer can even provide and receive services at the same time. Figure 25.3 shows an example of communication in this paradigm. One of the areas that really fits in this paradigm is the Internet telephony. Communication by phone is indeed a peer-to-peer activity; no party needs to be running forever waiting for the other party to call. Another area in which the peer-to-peer paradigm can be used is when some computers connected to the Internet have something to share with each other. For example, if an Internet user has a file available to share with other Internet users, there is no need for the file holder to become a server and run a server process all the time waiting for other users to connect and retrieve the file.

Although the peer-to-peer paradigm has been proved to be easily scalable and cost-effective in eliminating the need for expensive servers to be running and maintained all the time, there are also some challenges. The main challenge has been security; it is more difficult to create secure communication between distributed services than between those controlled by some dedicated servers. The other challenge is applicability; it appears that not all applications can use this new paradigm. For example, not many Internet users are ready to become involved, if one day the Web can be implemented as a peer-to-peer service. An application may choose to use a mixture of the two paradigms by combining the advantages of both. For example, a light-load client-server communication can be used to find the address of the peer that can offer a service. When the address of the peer is found, the actual service can be received from the peer by using the peer-to peer Paradigm.

Figure 25.3 *Example of a peer-to-peer paradigm*



Ans-

Q 9 c) Discuss the connection types in HTTP along with formats of message

Ans-1. Connection Types in HTTP

Non-Persistent Connection (HTTP/1.0)

Concept:

- A separate TCP connection is opened for each HTTP request/response pair.
- After the response is sent, the connection is closed.

Working:

- Client opens TCP connection
- Sends request
- Receives response
- Connection closed

Advantages:

- Simple to implement

Disadvantages:

- High overhead due to repeated TCP connection setup
- Increased latency

Persistent Connection (HTTP/1.1)

Concept:

- A single TCP connection is used for multiple HTTP requests and responses.
- Connection remains open unless explicitly closed.

Working:

- Client sends multiple requests on the same connection
- Server sends multiple responses

Advantages:

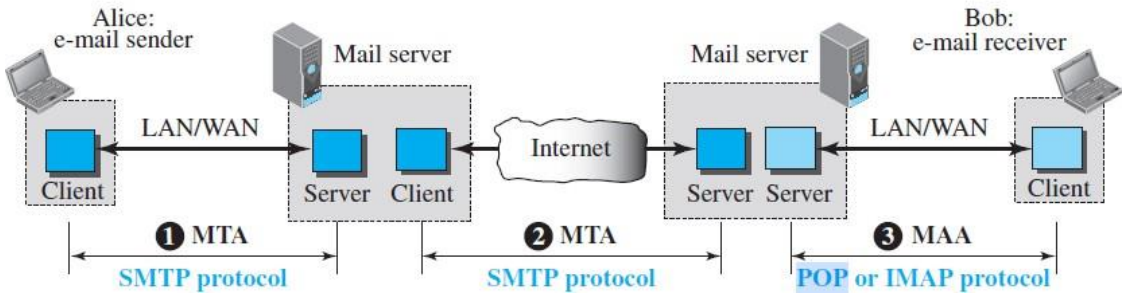
- Reduced latency
- Lower network overhead
- Better performance

Disadvantages:

- Server resources occupied longer

Q 10 a) Explain POP and IMAP protocols

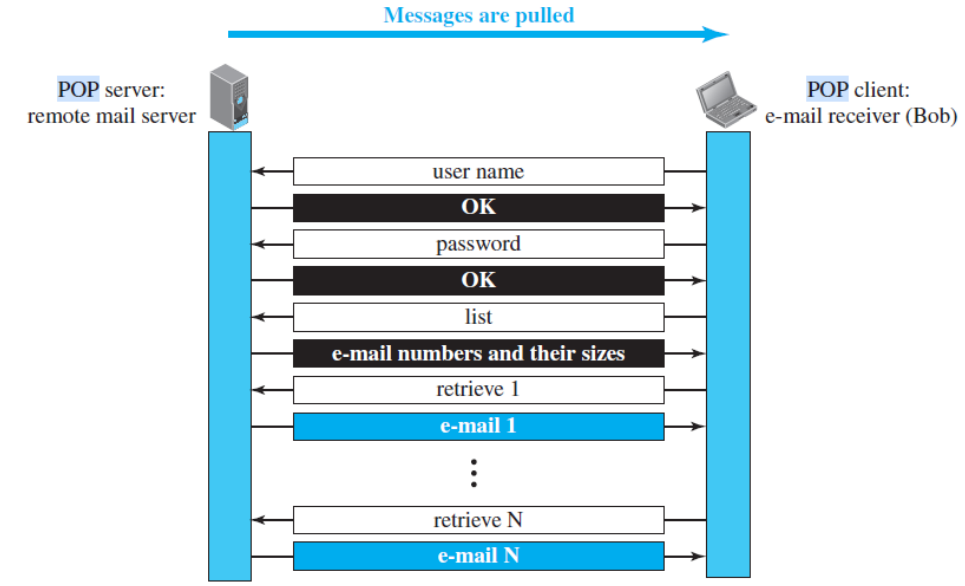
Figure 26.15 *Protocols used in electronic mail*



Ans-POP and IMAP Protocols

POP (Post Office Protocol) and IMAP (Internet Message Access Protocol) are application-layer email retrieval protocols used by mail clients to access emails from a mail server.

Figure 26.17 *POP3*



---

1. POP (Post Office Protocol)

Definition

POP allows a user to download emails from the mail server to the local device.

Current version: POP3
Port numbers:

- 110 (unencrypted)

- 995 (POP3S – encrypted)

Working of POP

1. Client connects to mail server
2. Authenticates using username/password
3. Downloads emails to local system
4. Emails are usually deleted from the server

Features

- Simple and easy to use
- Offline access to emails
- One-way communication

Advantages

- Saves server storage
- Works well with limited internet access
- Simple implementation

Limitations

- Emails not synchronized across devices
- Risk of data loss if device fails
- No server-side folder management

Example Use Case

- Personal email access on a single device

---

2. IMAP (Internet Message Access Protocol)

Definition

IMAP allows users to access and manage emails directly on the mail server.

Current version: IMAP4
Port numbers:

- 143 (unencrypted)
- 993 (IMAPS – encrypted)

Working of IMAP

1. Client connects to mail server
2. Authenticates
3. Emails remain on the server
4. Actions are synchronized across devices

Features

- Two-way communication
- Server-side folder management
- Supports multiple devices
- Partial message download

Advantages

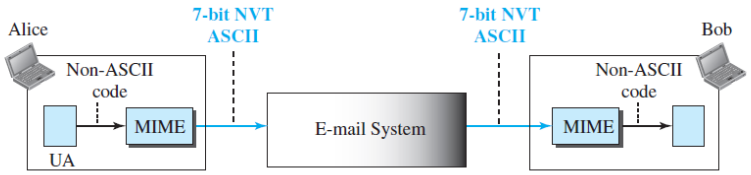
- Emails accessible from anywhere

- Automatic synchronization
- Better suited for mobile and cloud environments

Limitations

- Requires continuous internet access
- Uses more server storage
- More complex than POP

Figure 26.18 *MIME*



MIME

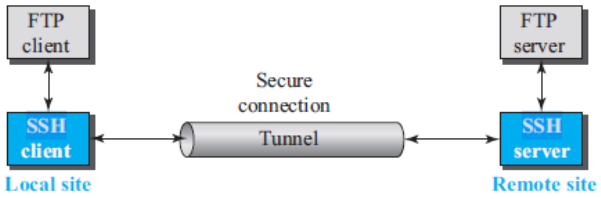
Electronic mail has a simple structure. Its simplicity, however, comes with a price. It can send messages only in NVT 7-bit ASCII format. In other words, it has some limitations. It cannot be used for languages other than English (such as French, German, Hebrew, Russian, Chinese, and Japanese). Also, it cannot be used to send

binary files or video or audio data. Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that allows non-ASCII data to be sent through e-mail. MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data. We can think of MIME as a set of software functions that transforms non-ASCII data to ASCII data and vice versa, as shown in Figure 26.18.

Q 10 b) Discuss the applications of SSH protocol

Ans-Although SSH is often thought of as a replacement for TELNET, SSH is, in fact, a general-purpose protocol that provides a secure connection between a client and server. SSH for Remote Logging Several free and commercial applications use SSH for remote logging. Among them, we can mention PuTTy, by Simon Tatham, which is a client SSH program that can be used for remote logging. Another application program is Tectia, which can be used on several platforms.

Figure 26.26 *Port forwarding*



SSH for File Transfer

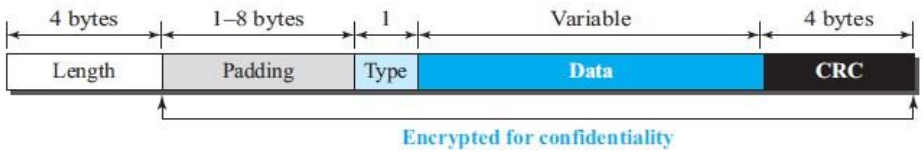
One of the application programs that is built on top of SSH for file transfer is the Secure File Transfer Program (sftp). The sftp application program uses one of the channels provided by the SSH to transfer files. Another common application is called Secure Copy (scp). This application uses the same format as the UNIX copy command, cp, to copy files.

Port Forwarding One of the interesting services provided by the SSH protocol is port forwarding. We can use the secured channels available in SSH to access an application program that does not provide security services. Applications such as TELNET and Simple Mail Transfer Protocol (SMTP), which are discussed above, can use the services of the SSH port forwarding mechanism. The SSH port forwarding mechanism creates a tunnel through which the messages belonging to other protocols can travel. For this reason, this mechanism is sometimes referred to as SSH tunneling. Figure 26.26 shows the The FTP client can use the SSH client on the local site to make a secure connection with the SSH server on the remote site. Any request from the FTP client to the FTP server is carried through the tunnel provided by the SSH client and server. Any response from the FTP server to the FTP client is also carried through the tunnel provided by the SSH client and server. Format of the SSH Packets Figure 26.27 shows the format of packets used by the SSH protocols. The length field defines the length of the packet but does not include the padding.One to eight bytes of padding is added to the packet to make the attack on the security provision more difficult. The cyclic redundancy check (CRC) field is used for error detection. The type field designates the type of the packet used in different SSH protocols. The data field is the data transferred by the packet in different protocols.

Format of the SSH Packets

Figure 26.27 shows the format of packets used by the SSH protocols.

Figure 26.27 *SSH packet format*



Q 10 c) Explain resolution in DNS

Ans- DNS Resolution

DNS (Domain Name System) resolution is the process of translating a human-readable domain name (like www.example.com) into its corresponding IP address (like 93.184.216.34) so that computers can

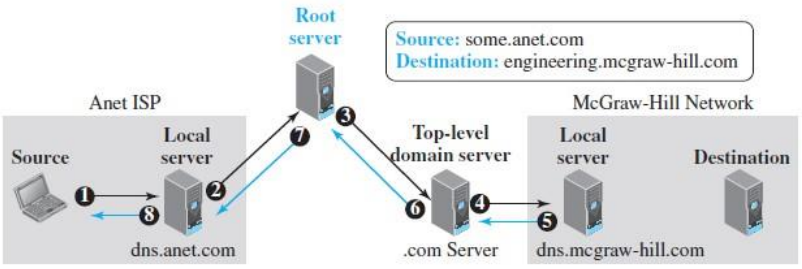
locate each other on the network.

Types of DNS Resolution

1. Recursive Resolution

- The client requests the DNS resolver to fully resolve the domain name.
- The resolver queries other DNS servers on behalf of the client and returns the final IP address.

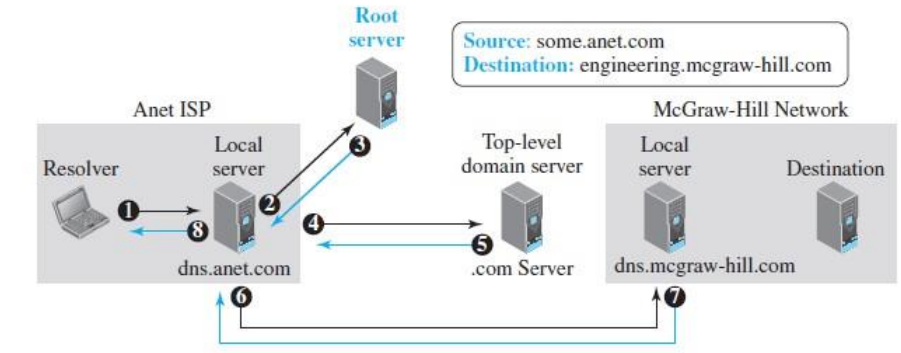
Figure 26.36 *Recursive resolution*



2. Iterative Resolution

- Each DNS server returns the best information it has.
- The client (or resolver) continues querying the next server until the IP address is found.

Figure 26.37 *Iterative resolution*



DNS Resolution Process (Step-by-Step)

1. User Request
   User enters a domain name (e.g., www.example.com) in a browser.

2. Local DNS Cache Check
   The system checks:
   - Browser cache
   - OS cache
   - Local DNS resolver cache
3. Query to Recursive Resolver
   If not found, the request is sent to a recursive DNS resolver (usually ISP or public DNS).
4. Root DNS Server Query
   Resolver asks the root server, which replies with the address of the TLD server (e.g., .com).
5. TLD DNS Server Query
   Resolver queries the Top-Level Domain (TLD) server, which responds with the authoritative DNS server.
6. Authoritative DNS Server Query
   Authoritative server returns the IP address of the domain.
7. Response to Client
   Resolver sends the IP address back to the client and stores it in cache.