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.

			M	L	C
		Module – 1			
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

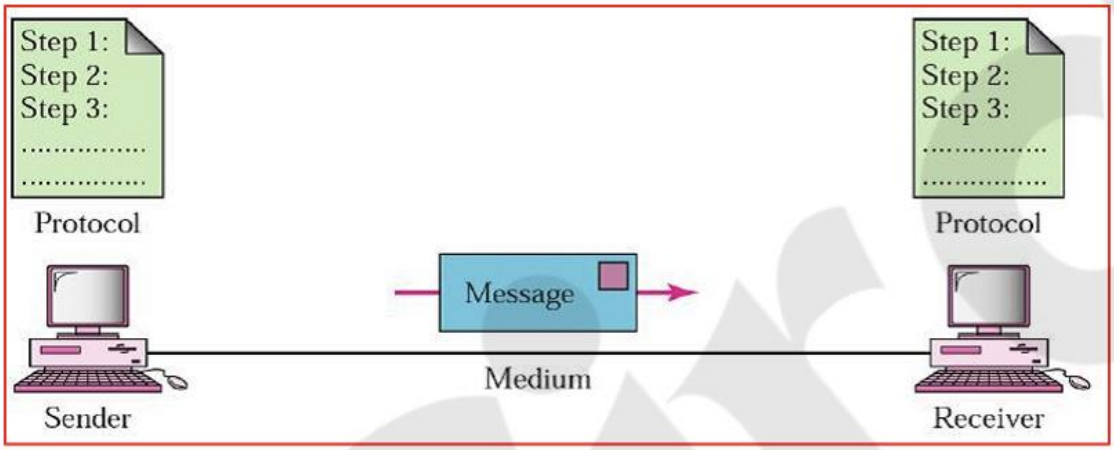
		OR			
Q.6	a.	Explain IPv6 packet format in detail.	6	L2	CO2
	b.	Discuss D-V routing highlighting the importance of distance vector.	7	L2	CO2
	c.	Describe BGP protocol in detail.	7	L2	CO4
		Module – 4			
Q.7	a.	Explain the concept of port numbers mentioning ICANN ranges.	5	L2	CO3
	b.	Explain Go-Back-N 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

* * * * *

Q1 a) Explain the fundamental characteristics and data representation in data communication.

DATA COMMUNICATIONS

Data communication is the process of transferring data from one point to another using a communication system. It involves several essential components and mechanisms to ensure the accurate and timely delivery of data.



Fundamental Characteristics of Data Communication

For a data communication system to be effective, it must possess the following fundamental characteristics:

1.1 Delivery

- **Data must be delivered to the correct destination device.**
- **Incorrect delivery makes communication meaningless.**

1.2 Accuracy

- **Data must be delivered without errors.**
- **Any alteration of data leads to incorrect information.**

1.3 Timeliness

- **Data must be delivered within an acceptable time limit.**
- **Important for real-time applications like video conferencing and online calls.**

1.4 Jitter

- **Jitter is the variation in packet arrival time.**
- **Excessive jitter degrades the quality of audio and video transmission.**

2. Data Representation in Data Communication

Data can be represented in the following forms:

2.1 Text

- **Represented using character encoding schemes such as ASCII and Unicode.**
- **Each character is represented by a unique binary code.**

2.2 Numbers

- **Numbers are represented in binary format.**
- **Used for mathematical and computational operations.**

2.3 Images

- **Images are represented as a matrix of pixels.**
- **Each pixel value represents color intensity.**

2.4 Audio

- **Audio is a continuous signal converted into digital form using sampling and quantization.**

2.5 Video

- **Video is a combination of images (frames) and audio, represented digitally.**

Conclusion

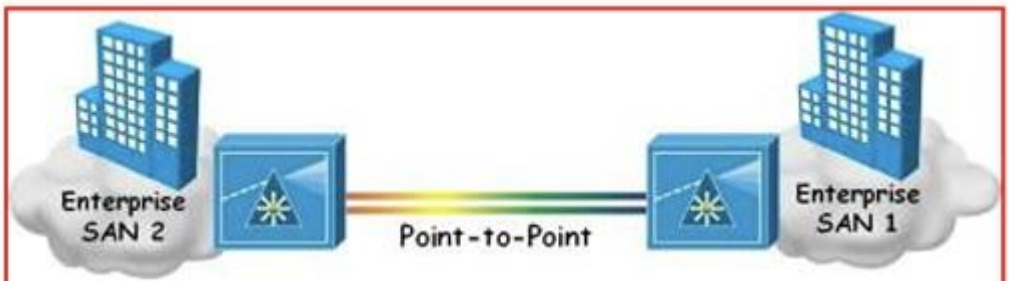
Efficient data communication depends on fundamental characteristics such as delivery, accuracy, timeliness, and jitter, while different types of data are represented digitally in the form of text, numbers, images, audio, and video.

Q1 b) Discuss the types of connections and basic network topologies.

Types of Connections

A connection refers to how devices are linked to communicate data. There are two main types:

1.1 Point-to-Point Connection



- **A dedicated link exists between two devices.**
- **Data flows directly from sender to receiver.**

- **Entire bandwidth is used by the two connected devices.**

Example:
Computer connected to a printer, router-to-router link.

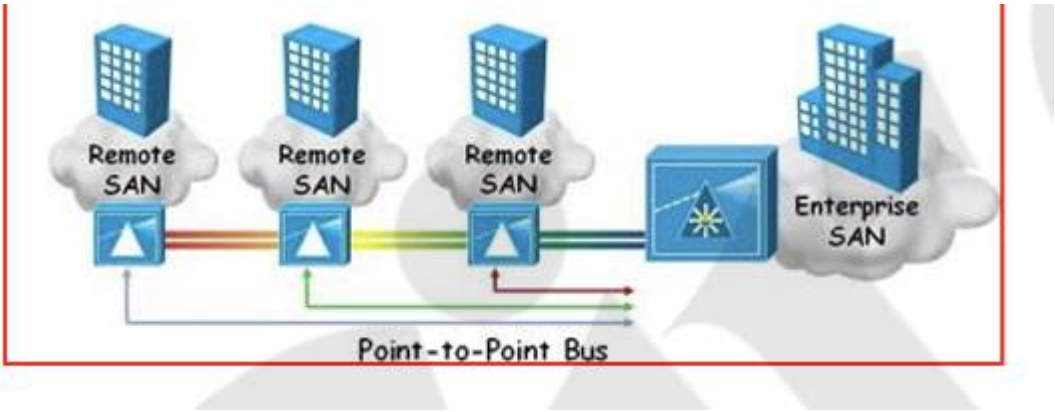
Advantages:

- **High data speed**
- **Secure and reliable**

Diagram:

Device A ———— Device B

1.2 Multipoint (Multidrop) Connection



- **A single communication link is shared by multiple devices.**
- **Bandwidth is shared among connected devices.**

Example:
Traditional bus network.

Advantages:

- **Cost-effective**
- **Easy to expand**

Disadvantages:

- **Lower performance**
- **Possible data collision**

Diagram:

Device A

|

Device B ———— Shared Link ———— Device C

2. Basic Network Topologies

A network topology defines the physical or logical arrangement of devices in a network.

2.1 Bus Topology

- **All devices are connected to a single backbone cable.**
- **Data travels in both directions.**

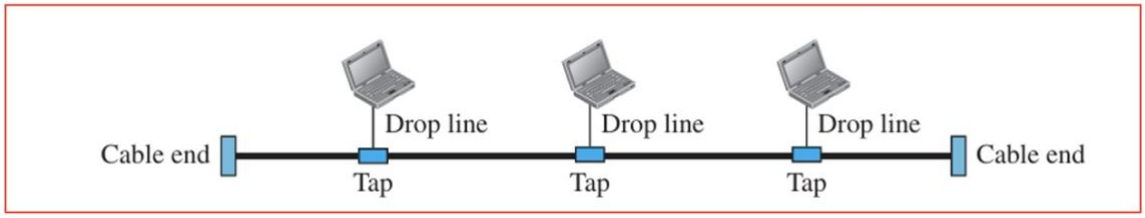
Advantages:

- **Easy to install**
- **Low cost**

Disadvantages:

- **Backbone failure brings down the network**
- **Difficult fault isolation**

Diagram:



2.2 Star Topology

- **All devices are connected to a central hub or switch.**
- **Most commonly used topology today.**

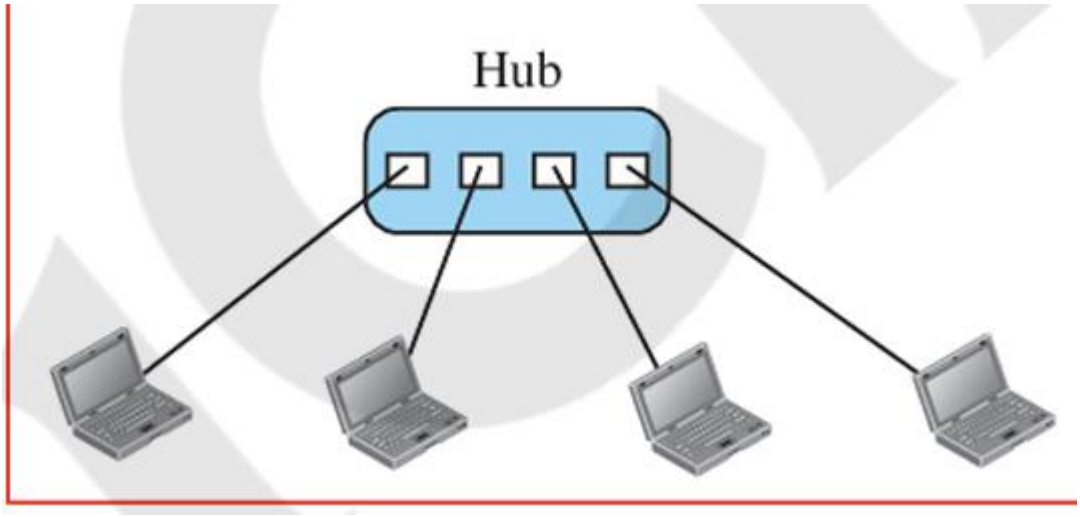
Advantages:

- **Easy fault detection**
- **Scalable and reliable**

Disadvantages:

- **Central device failure stops the network**
- **Higher cabling cost**

Diagram:



2.3 Ring Topology

- **Devices are connected in a circular fashion.**
- **Data flows in one direction (usually).**

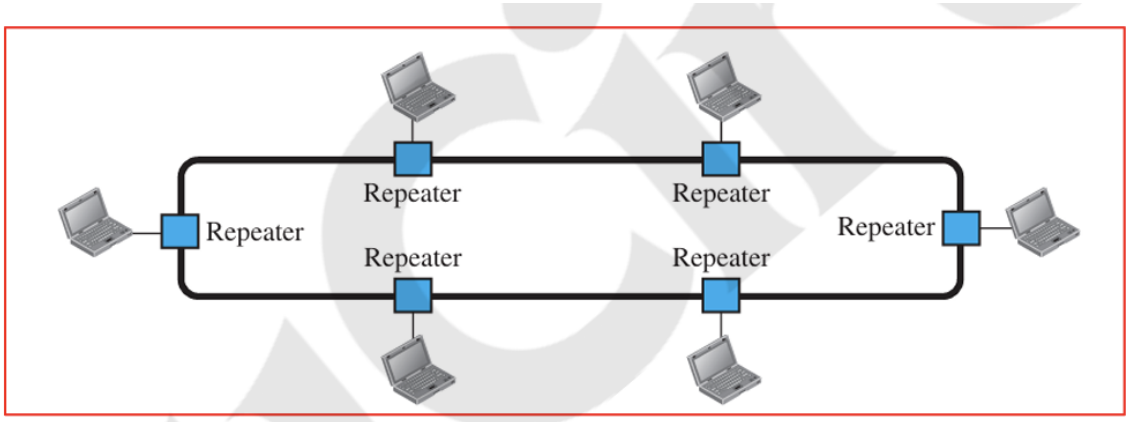
Advantages:

- **No data collision**
- **Equal access to network**

Disadvantages:

- **Failure of one node affects entire network**
- **Difficult to troubleshoot**

Diagram:



2.4 Mesh Topology

- **Every device is connected to every other device.**
- **Can be full mesh or partial mesh.**

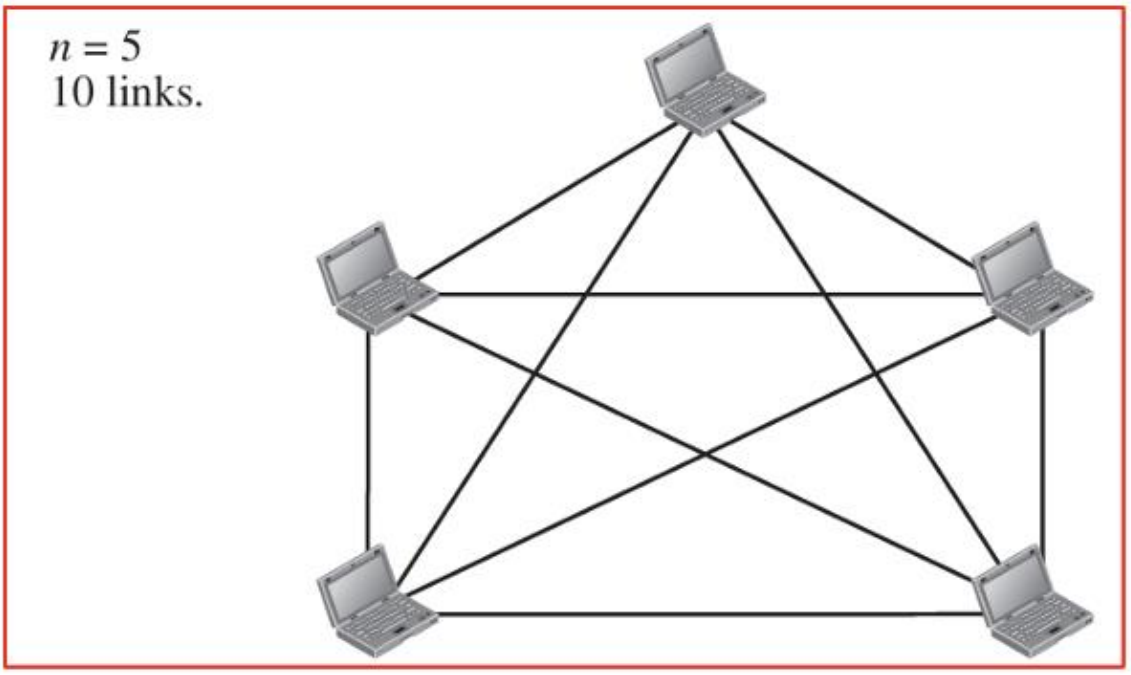
Advantages:

- **Highly reliable**
- **No single point of failure**

Disadvantages:

- **Expensive**
- **Complex installation**

Diagram:



Conclusion

Point-to-point and multipoint connections define how devices share links, while bus, star, ring, and mesh topologies define how networks are structured. Each topology has its own advantages and is chosen based on cost, reliability, and scalability.

Q1 c) Explain packet switching and circuit switching with neat diagrams.
Definition of Switching

Switching is a technique used in telecommunication and networking to route data or voice signals

between devices in a network. It ensures efficient utilization of resources by directing data packets or

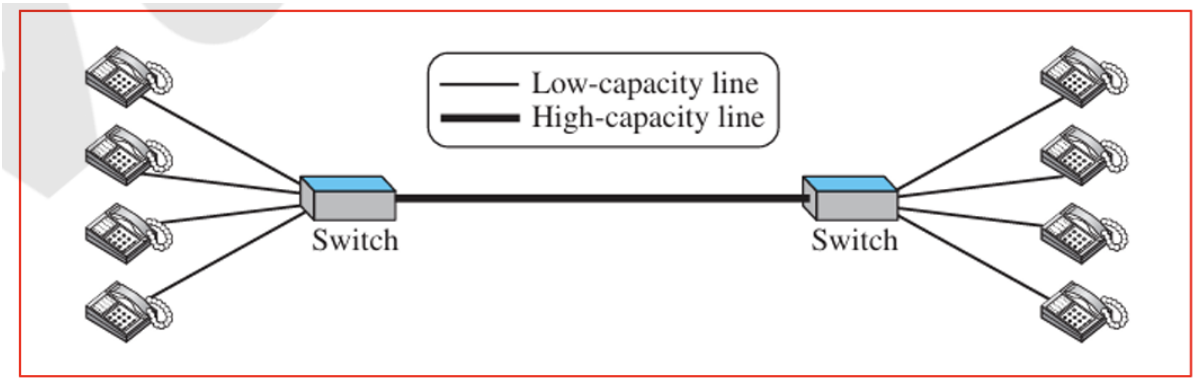
establishing dedicated communication paths between the sender and receiver.

Types of Switching

1. Circuit-Switched Network

2. Packet-Switched Network

1. Circuit-Switched Network



In a circuit-switched network, a dedicated communication path or circuit is established between the

sender and receiver for the duration of the communication session. This type of switching is

commonly used in traditional telephone networks.

Features:

• Dedicated Path: A single, exclusive channel is established for communication.

• Real-Time Communication: Suitable for applications like voice calls.

• Resource Usage: Resources are reserved and cannot be shared with others until the session

ends.

• Latency: Minimal latency once the circuit is established.

Advantages:

• Reliable and consistent data transfer.

• Predictable performance due to a fixed data transfer path.

Disadvantages:

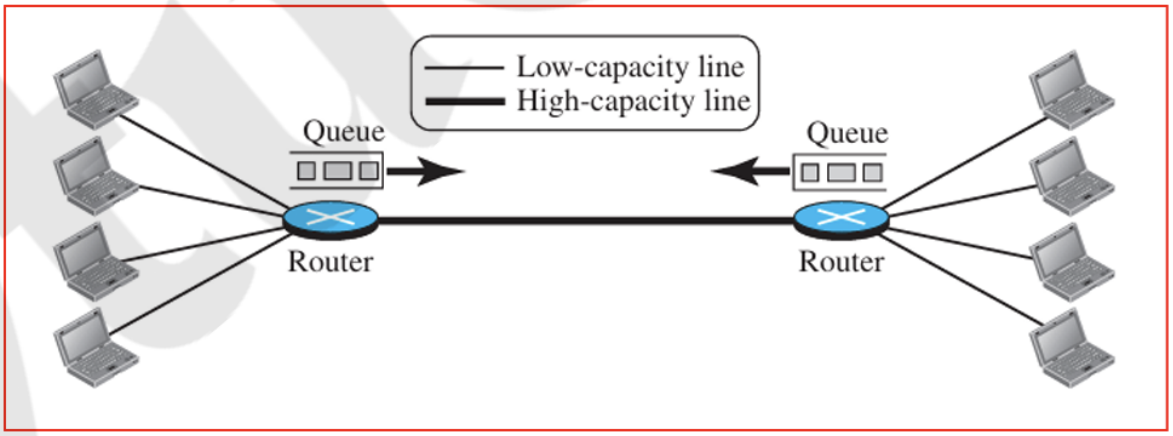
• Inefficient use of resources as the dedicated channel remains idle when not in use.

• Establishing a circuit takes time.

Example:

Traditional Public Switched Telephone Network (PSTN).

2. Packet-Switched Network



In a packet-switched network, data is divided into packets, and each packet is transmitted independently over shared network resources. Packets may take different routes to reach the destination, where they are reassembled in the correct order.

Features:

- **No Dedicated Path: Packets are routed dynamically based on availability.**
- **Efficient Resource Use: Resources are shared among multiple users.**
- **Store-and-Forward Mechanism: Routers store packets temporarily before forwarding them.**
- **Data Integrity: Packets are checked for errors and retransmitted if needed.**

Advantages:

- **Efficient resource utilization.**
- **Scalable and adaptable to varying traffic conditions.**
- **Lower cost compared to circuit-switched networks.**

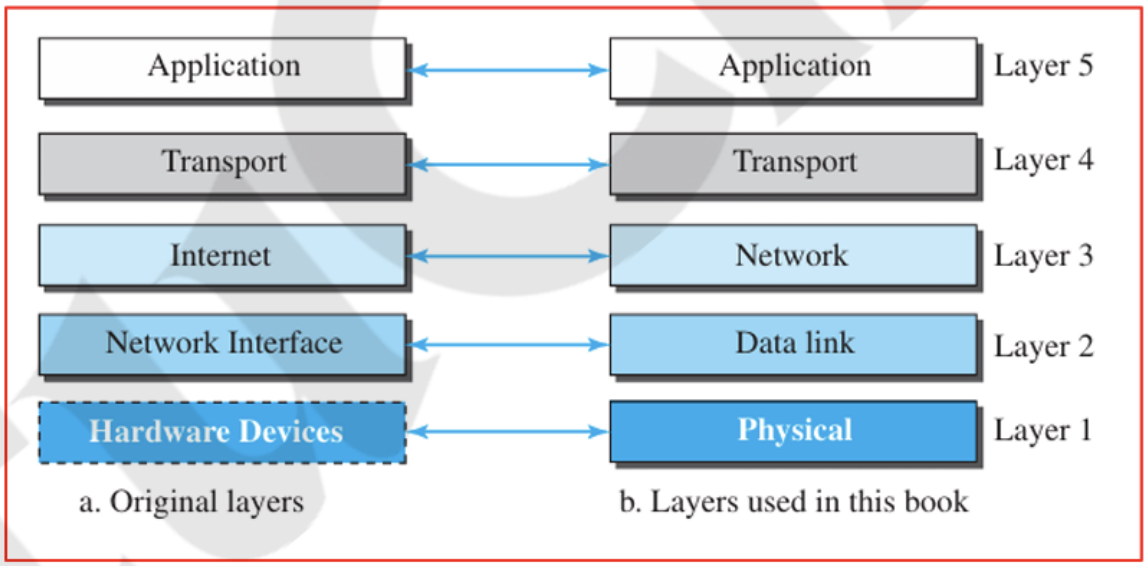
Disadvantages:

- **Latency due to packet reassembly and routing delays.**
- **Potential for packet loss or out-of-order delivery.**

Example:

The Internet, which uses protocols like TCP/IP.

Q2 a) Explain the layers of the TCP/IP protocol suite.



Layers of TCP/IP Protocol Suite

The TCP/IP protocol suite is a set of communication protocols used to connect network devices on the internet. It is organized into four abstraction layers, each responsible for specific functionalities.

1. Application Layer

• Function: Provides network services to applications. It handles high-level protocols and user interfaces.

• Protocols:

o HTTP, HTTPS (web browsing)

o SMTP (email sending)

o FTP (file transfer)

o DNS (domain name resolution)

• Examples: Web browsers, email clients, and file transfer applications.

2. Transport Layer

• Function: Ensures reliable data delivery and provides end-to-end communication between devices.

• Key Features:

o Segmentation: Divides large data into smaller segments.

o Flow Control: Manages the data transmission rate to avoid congestion.

o Error Control: Ensures data integrity by retransmitting lost or corrupted packets.

• Protocols:

o TCP (reliable, connection-oriented)

o UDP (faster, connectionless)

3. Internet Layer

• Function: Handles logical addressing and routing of data packets between devices.

• Key Features:

o Logical IP addressing for unique identification of devices.

o Routing of packets across networks.

• Protocols:

o IP (IPv4, IPv6): Provides addressing and routing.

o ICMP: Used for error reporting (e.g., ping command).

o ARP: Resolves IP addresses to MAC addresses.

4. Network Access Layer

• Function: Deals with the hardware and physical transmission of data.

• Key Features:

o Converts packets into frames for physical transmission.

o Responsible for Media Access Control (MAC) addressing.

• Protocols:

o Ethernet, Wi-Fi, DSL.

o Manages data transmission over various physical media.

Key Points

• The TCP/IP model maps to the OSI model but has fewer layers:

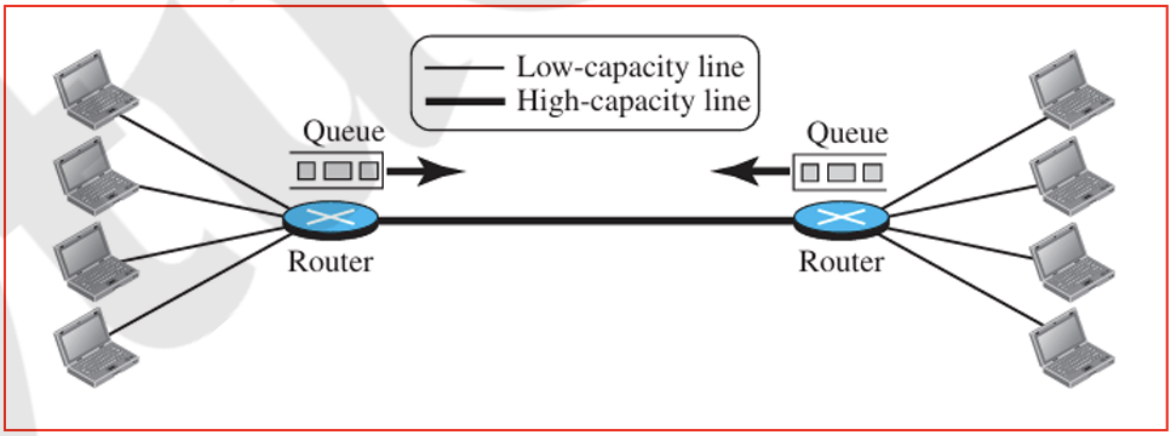
o Application Layer combines the Application, Presentation, and Session layers of OSI.

o Network Access Layer corresponds to the Data Link and Physical layers of OSI.

• It is widely used because it is simpler and directly maps to internet communication.

Q2 b) Explain the types of packet-switched networks and evaluate the total delay time of both.

1. Packet-Switched Networks

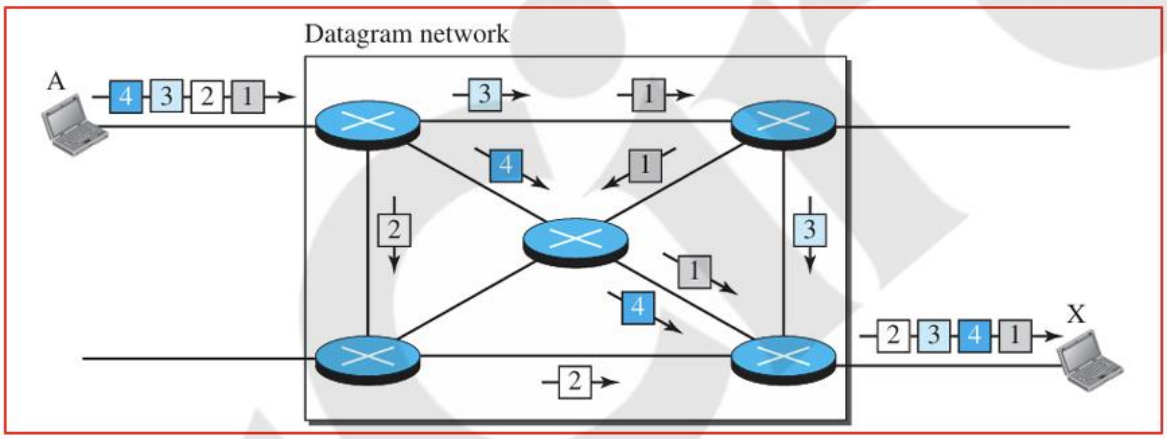


In packet-switched networks, data is divided into small packets, and each packet is transmitted independently through the network.
Resources are shared, making communication efficient.

Packet-switched networks are broadly classified into two types:

1. **Datagram Packet-Switched Networks**
2. **Virtual Circuit Packet-Switched Networks**

2. Datagram Packet-Switched Network



Explanation

- **No pre-established path between sender and receiver.**
- **Each packet is treated as an independent unit.**
- **Packets may follow different routes to reach the destination.**
- **Order of packets is not guaranteed.**

Characteristics

- **No setup phase**
- **Routing decision made for each packet**
- **Packet loss and delay variation possible**

Example

- **Internet (IP protocol)**

Total Delay in Datagram Network

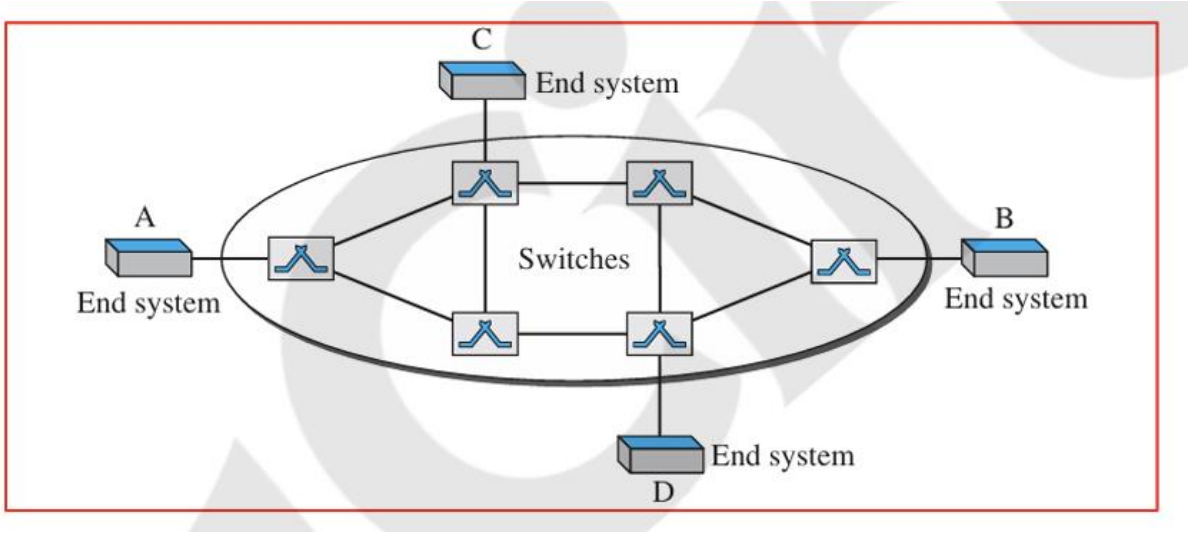
Total delay is the sum of delays for each packet at every hop.

$$\text{Total Delay} = \sum(\text{Processing Delay} + \text{Queuing Delay} + \text{Transmission Delay} + \text{Propagation Delay})$$

- **Delay may vary packet to packet**
- **Queuing delay is unpredictable**

Overall: Delay is variable and non-deterministic

3. Virtual Circuit Packet-Switched Network



Explanation

- **A logical path (virtual circuit) is established before data transmission.**
- **All packets follow the same route.**
- **Packets arrive in order.**
- **Similar to circuit switching but without reserving bandwidth.**

Phases

1. **Setup phase**

2. **Data transfer phase**
3. **Teardown phase**

Examples

- **X.25**
- **Frame Relay**
- **ATM**

Total Delay in Virtual Circuit Network

$$\text{Total Delay} = \text{Setup Delay} + \sum(\text{Transmission Delay} + \text{Propagation Delay}) + \text{Teardown Delay}$$

- **Routing decision made only once**
- **Queuing delay is lower and more predictable**

Overall: Delay is more stable and ordered

4. Comparison of Total Delay

Feature	Datagram Network	Virtual Circuit Network
Connection setup	Not required	Required
Routing	Per packet	Once per session
Delay	Variable	Predictable
Packet order	Not guaranteed	Guaranteed
Queuing delay	High	Low
Example	Internet	ATM, Frame Relay

5. Conclusion

- **Datagram networks offer flexibility but suffer from variable delay.**
- **Virtual circuit networks provide ordered delivery and stable delay at the cost of setup overhead.**
- **Choice depends on application requirements (real-time vs best-effort).**

Q3 a) Explain types of errors and Hamming distance.
Find the Hamming distance between the following:

i) d(000, 011)
ii) d(10101, 11110)

1. Types of Errors in Data Communication

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

1.1 Single-Bit Error

- **Only one bit in the data unit is changed.**
- **Example:**
- **Sent: 100101**
- **Received: 100111**

1.2 Multiple-Bit Error

- **More than one bit is altered, but not necessarily adjacent.**
- **Example:**
- **Sent: 1100110**
- **Received: 1000011**

1.3 Burst Error

- **Two or more consecutive bits are changed.**
- **Length of burst = distance between first and last corrupted bit.**
- **Most common and dangerous type of error.**

Example:

Sent: 101100010

Received: 101011110

2. Hamming Distance

Definition

Hamming distance is the number of bit positions in which two data strings of equal length differ.

Mathematical Representation

$$d(x, y) = \text{number of differing bits between } x \text{ and } y$$

Importance of Hamming Distance

- **Determines error-detection capability.**
- **Determines error-correction capability.**
- **Minimum Hamming distance:**
    - **To detect d errors → distance ≥ d + 1**
    - **To correct t errors → distance ≥ 2t + 1**

3. Hamming Distance Calculations

i) d(000, 011)

000

011

Differences at positions 2 and 3

$$d(000, 011) = 2$$

ii) d(10101, 11110)

10101

11110

Differences at positions 2, 4, and 5

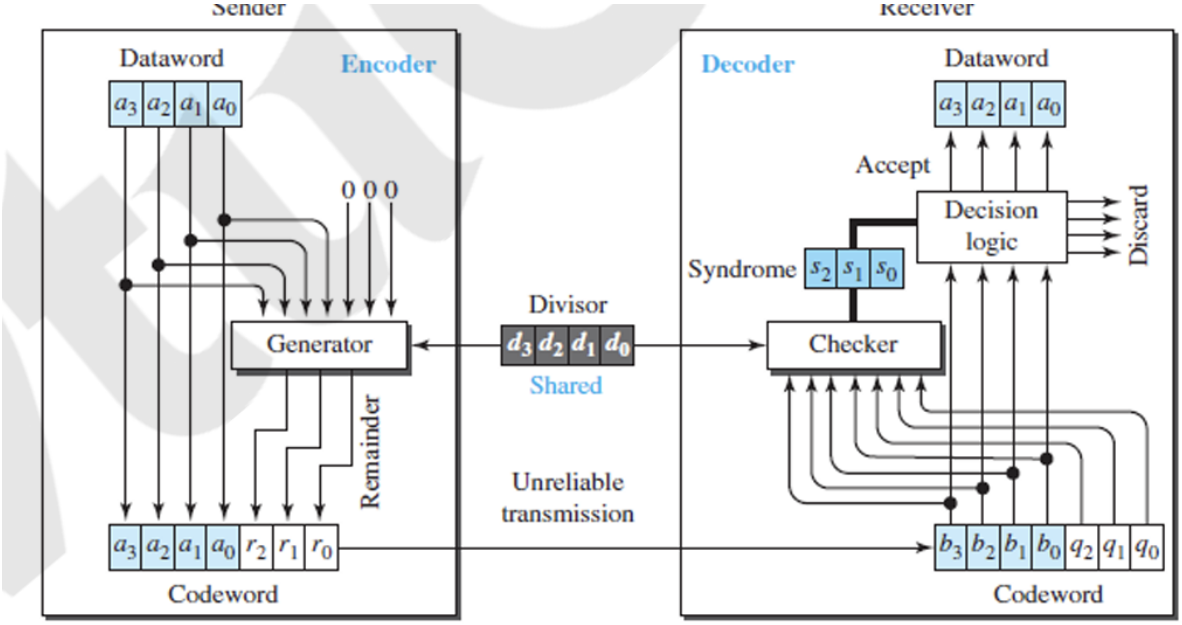
$$d(10101, 11110) = 3$$

Q3 b) Describe the working of CRC encoder and decoder.
Perform CRC division for the following:

- **Data word: 1001**
- **Divisor: 1011**

1. Cyclic Redundancy Check (CRC)

CRC is an error-detection technique used in the Data Link Layer.
It treats data as a binary polynomial and performs modulo-2 division to generate a redundant check value (CRC bits).



2. Working of CRC Encoder

Steps at Sender (Encoder):

1. **Let**
    - **Data word = D**
    - **Divisor (Generator polynomial) = G (length = r+1)**
2. **Append r zeros to the data word, where**
3. **r = length of divisor – 1**
4. **Perform modulo-2 division (XOR-based division) of the appended data by the divisor.**
5. **The remainder obtained is the CRC bits.**
6. **Append CRC bits to the original data → Codeword.**

CRC ENCODER FLOW

Data → Append zeros → Mod-2 Divider → Remainder (CRC)

↓

Codeword sent

3. Working of CRC Decoder

Steps at Receiver (Decoder):

1. **Receiver receives the codeword.**
2. **Divides the received codeword by the same divisor.**

3. **Checks the remainder:**
    - **Remainder = 0 → No error**
    - **Remainder ≠ 0 → Error detected**

CRC DECODER FLOW

Received Codeword → Mod-2 Divider → Remainder

↓

Error / No Error

4. CRC Division (Solved Example)

Given:

- **Data word = 1001**
- **Divisor = 1011**

Step 1: Append zeros

Divisor length = 4
So, append 3 zeros to data word:

1001 → 1001000

Step 2: Perform Modulo-2 Division

1011) 1001000

1011

0010

0000

0100

0000

1000

1011

---

011 ← Remainder

Step 3: CRC Bits

CRC = 011

Step 4: Codeword

Append CRC to original data:

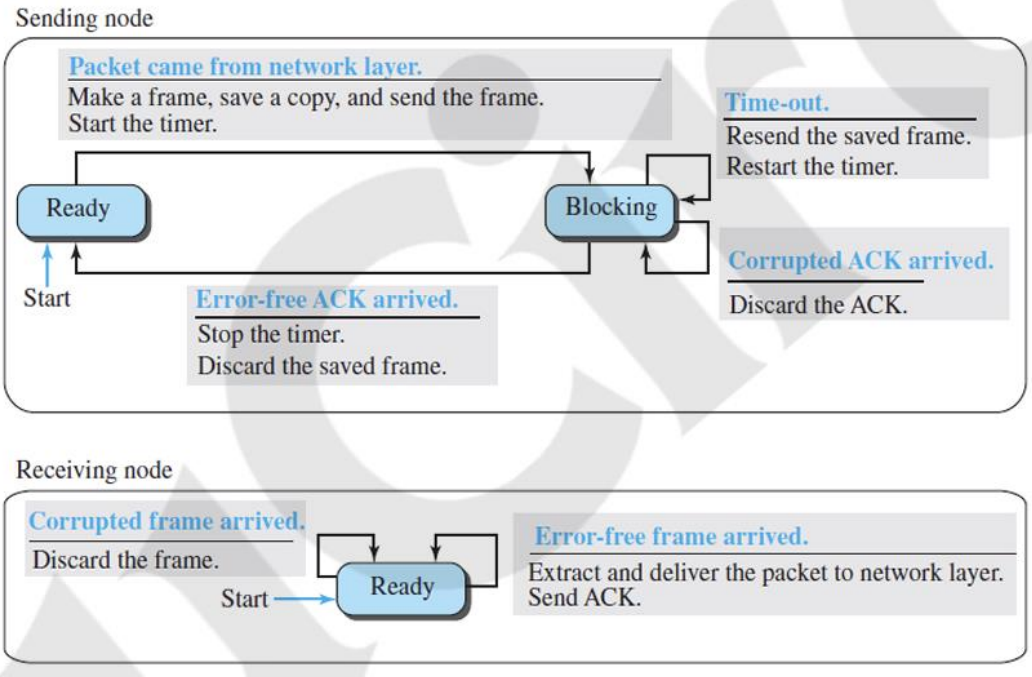
Codeword = 1001 + 011 = 1001011

5. Verification at Receiver

1001011 ÷ 1011 → Remainder = 000

✓ No error detected

Q4 a) Explain the Stop-and-Wait protocol using FSM (Finite State Machine).



Stop-and-Wait Protocol using FSM (Finite State Machine)

Introduction

The Stop-and-Wait protocol is a simple flow and error control protocol used in the data link layer. In this protocol, the sender transmits one frame and waits for an acknowledgment (ACK) before sending the next frame.

A Finite State Machine (FSM) is used to represent the behavior of the sender and receiver using states and transitions.

1. Sender FSM

States of Sender

1. **Ready State (Wait for Call)**
    - **Sender is ready to send a frame.**
2. **Wait for ACK State**
    - **Sender waits for acknowledgment after sending a frame.**

Sender FSM Operation

- **Sender sends a frame with sequence number 0 or 1.**
- **Starts a timer.**
- **If ACK is received before timeout, sender switches sequence number and sends next frame.**
- **If ACK is lost or timeout occurs, sender retransmits the same frame.**

Sender FSM:

[Ready]

|

| send frame (seq=0/1)

v

[Wait for ACK]

**| **

ACK Timeout

**| **

v v

[Ready] Retransmit frame

2. Receiver FSM

States of Receiver

1. **Wait for Frame 0**
2. **Wait for Frame 1**

Receiver FSM Operation

- **Receiver checks sequence number of incoming frame.**
- **If correct frame is received:**
    - **Delivers data to upper layer.**
    - **Sends ACK.**
    - **Switches state.**
- **If duplicate frame is received:**
    - **Discards frame.**
    - **Resends previous ACK.**

Receiver FSM:

[Wait for Frame 0]

|

Frame 0 received

|

Send ACK0

v

[Wait for Frame 1]

[Wait for Frame 1]

|

Frame 1 received

|

Send ACK1

v

[Wait for Frame 0]

3. Features of Stop-and-Wait Protocol

- **Uses sequence numbers (0 and 1).**
- **Uses ACK and timeout mechanism.**
- **Ensures reliable data transmission.**
- **Simple but inefficient for high-speed networks.**

4. Advantages

- **Simple to implement**
- **Ensures reliability**
- **Easy error detection**

5. Disadvantages

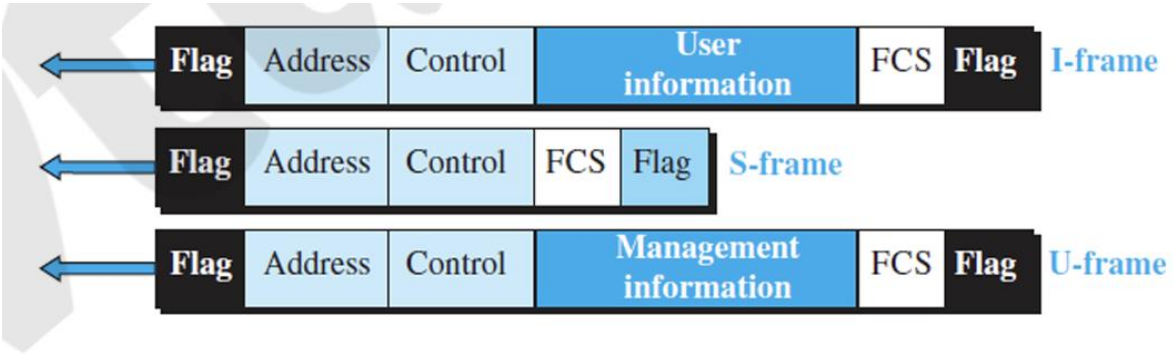
- **Poor channel utilization**
- **High waiting time**
- **Not suitable for high bandwidth-delay networks**

Conclusion

The Stop-and-Wait protocol uses a finite number of states to control data transmission and acknowledgments, ensuring reliable communication by allowing only one outstanding frame at a time

Q4 b) Explain the three types of frames in HDLC.

Control Fields in Frames (1-frames, S-frames, and U-frames)



In the HDLC (High-Level Data Link Control) protocol, control fields are used to manage and control the communication between devices. The HDLC protocol defines three types of frames:

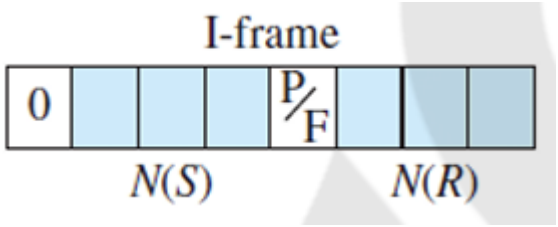
1. I-Frames (Information Frames): Used for data transmission.

2. S-Frames (Supervisory Frames): Used for flow control and error control.

3. U-Frames (Unnumbered Frames): Used for network management and control.

Each frame type has a specific control field structure that serves different purposes.

1. I-Frames (Information Frames)



I-frames carry user data and control information for flow and error control.

Control Field Structure for I-Frames:

Bits

N(S)

N(R)

Description

Sequence number of the transmitted frame.

Acknowledgment number for the received frame.

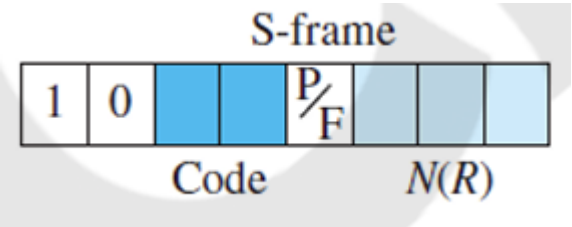
P/F (Poll/Final) Indicates whether the frame is a poll or final frame.

• Purpose:

o Transmit user data between devices.

o Include acknowledgment for previously received data frames.

2. S-Frames (Supervisory Frames)



S-frames are used for flow control and error control. They do not carry user data but manage the communication session.

Control Field Structure for S-Frames:

Bits

Control Type (2 bits)

N(R)

P/F

• Purpose:

Description

Indicates the type of S-frame: - 00: Receive Ready (RR) - 01: Receive Not Ready (RNR) - 10: Reject (REJ) - 11: Selective Reject (SREJ).

Acknowledgment number for the last correctly received I-frame.

Poll/Final bit for session management.

o Manage flow control:

▪ RR: Indicates the receiver is ready to accept more frames.

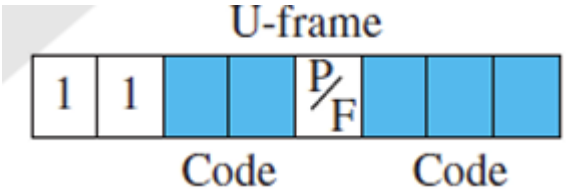
▪ RNR: Indicates the receiver is not ready to accept frames.

o Manage error control:

▪ REJ: Signals a negative acknowledgment for a lost or erroneous frame.

▪ SREJ: Used for selective retransmission of a specific frame.

3. U-Frames (Unnumbered Frames)



U-frames are used for network management and control functions, such as connection establishment, disconnection, and error reporting.

Control Field Structure for U-Frames:

Bits

Description

Control Type (5 bits) Specifies the type of U-frame operation (e.g., SABME, DISC, UA).

P/F

Poll/Final bit for session management.

Command/Response Differentiates between command and response frames.

• Purpose:

o Establish, manage, and terminate communication sessions.

o Handle special control signals like Set Asynchronous Balanced Mode Extended

(SABME), Disconnect (DISC), and Unnumbered Acknowledgment (UA).

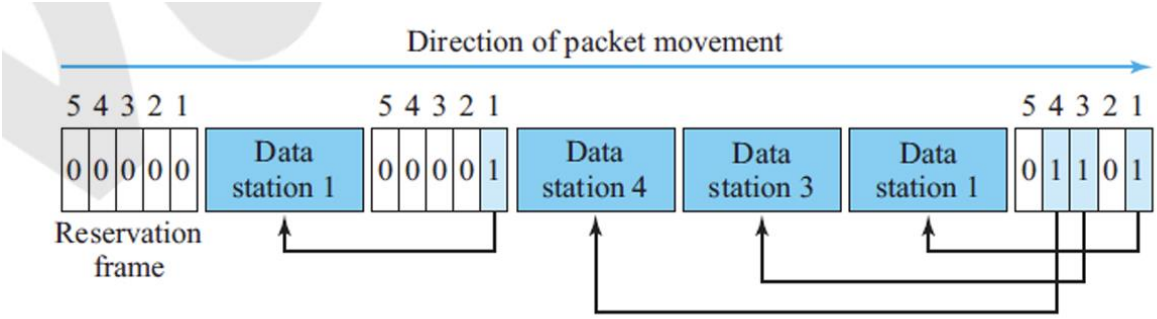
Q4 c) Discuss the controlled-access protocol using the reservation method.

1. Controlled-Access Protocol (Overview)

In controlled-access protocols, a station is allowed to transmit only when it has explicit permission. This avoids collisions and ensures orderly and fair access to the shared medium.

Examples of controlled-access protocols:

- **Reservation**
- **Polling**
- **Token Passing**



2. Reservation Method (Definition)

The reservation method is a controlled-access protocol in which stations reserve time slots in advance before transmitting data.

- **Time is divided into fixed cycles.**
- **Each cycle consists of:**
    - **Reservation phase**
    - **Data transmission phase**

Only stations that have successfully reserved a slot are allowed to transmit.

3. Working of Reservation Method

Step 1: Reservation Phase

- **The channel is divided into N reservation mini-slots.**

- **Each station is assigned one reservation slot.**
- **If a station wants to transmit data, it sets its bit to 1 in its reservation slot.**

Reservation Frame:

| S1 | S2 | S3 | S4 |

| 1 | 0 | 1 | 0 |

Stations S1 and S3 have reserved the channel.

Step 2: Data Transmission Phase

- **Only the stations that reserved slots transmit data.**
- **Transmission happens in order of reservation slots.**
- **No collision occurs during data transmission.**

Data Transmission Order:

S1 → S3

4. Characteristics of Reservation Method

- **Collision-free data transmission**
- **Fair access to all stations**
- **Requires synchronization among stations**
- **Suitable for systems with regular and heavy traffic**

5. Advantages

- **Eliminates collisions**
- **Efficient under high load**
- **Predictable delay**
- **Fair bandwidth distribution**

6. Disadvantages

- **Overhead due to reservation slots**
- **Inefficient when very few stations want to transmit**
- **Complex implementation**

7. Applications

- **Wireless networks**
- **Satellite communication**
- **Real-time data transmission systems**

Conclusion

The reservation method is an efficient controlled-access protocol that prevents collisions by allocating transmission slots in advance, making it suitable for high-traffic and time-critical networks.

Q5 a) Explain the services offered by the Network Layer.

1. Logical Addressing

- **Assigns logical addresses (IP addresses) to devices.**
- **Logical addresses uniquely identify a device across the entire network.**
- **Enables communication beyond the local network.**

Example: IPv4, IPv6 addresses.

2. Routing

- **Determines the best path for data to travel from source to destination.**
- **Uses routing algorithms and routing tables.**
- **Routers operate at the network layer.**

Example: Shortest path routing.

3. Packet Forwarding

- **Transfers packets from one network to another using routers.**
- **Decides the next hop based on the destination IP address.**
- **Ensures packets reach the correct destination network.**

4. Internetworking

- **Enables communication between different networks (LAN, MAN, WAN).**
- **Hides differences in hardware and protocols.**

- **Makes the Internet possible.**

5. Fragmentation and Reassembly

- **Breaks large packets into smaller fragments to match MTU (Maximum Transmission Unit).**
- **Reassembles fragments at the destination.**

Important when networks have different frame sizes.

6. Congestion Control (Basic Level)

- **Controls traffic flow to avoid network congestion.**
- **Uses techniques like packet dropping and routing adjustments.**

7. Error Handling and Diagnostics

- **Reports errors using protocols like ICMP.**
- **Helps detect unreachable hosts and network failures.**

8. Quality of Service (QoS) Support

- **Manages traffic priorities.**
- **Ensures better service for real-time data (audio, video).**

Conclusion

The Network Layer provides essential services such as logical addressing, routing, forwarding, fragmentation, and internetworking, ensuring efficient and reliable packet delivery across interconnected networks.

Q5 b) Define address space. Differentiate between classful and classless addressing.

1. Address Space

Address space is the total number of unique addresses that can be generated by an addressing scheme and used to identify devices on a network.

- **In computer networks, address space refers to all possible IP addresses available in a given IP version.**
- **The size of the address space depends on the number of bits used for addressing.**

Example:

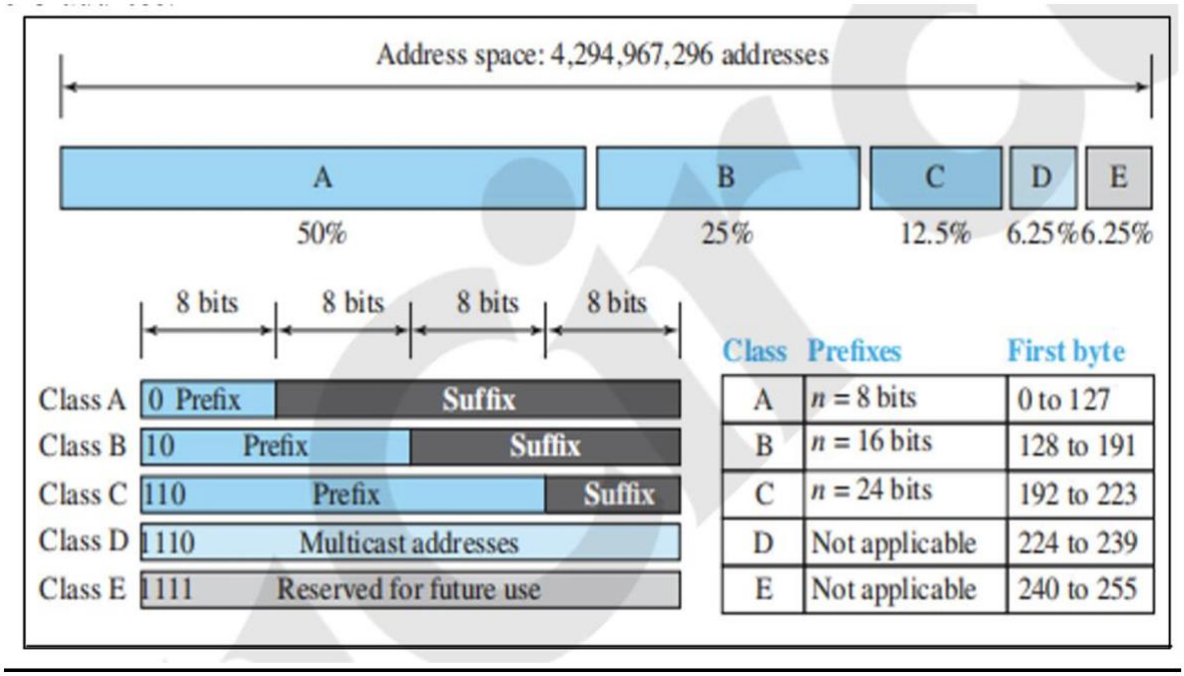
- **IPv4 uses 32 bits, so total address space =**

$$2^{32} = 4,294,967,296 \text{ addresses}$$

- **IPv6 uses 128 bits, providing a much larger address space.**

Address space ensures unique identification, routing, and scalability of networks.

2. 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

Definition

Classful addressing is an IP addressing method where the address space is divided into fixed classes based on the first few bits of the IP address.

IP Address Classes

Class	First Bits	Network Bits	Host Bits	Range
A	**0**	**8**	**24**	**1.0.0.0 – 126.255.255.255**
B	**10**	**16**	**16**	**128.0.0.0 – 191.255.255.255**
C	**110**	**24**	**8**	**192.0.0.0 – 223.255.255.255**

(Class D – Multicast, Class E – Experimental)

Characteristics

- **Fixed network and host boundaries**
- **No subnet mask required**

- **Leads to wastage of IP addresses**
- **Not scalable for modern networks**

3. Classless Addressing (CIDR)

Definition

Classless addressing, also known as CIDR (Classless Inter-Domain Routing), is an IP addressing method where the address space is divided without fixed classes, using a variable-length subnet mask (VLSM).

Representation

- **Written as:**
- **IP Address / Prefix Length**
- **Example:**
- **192.168.1.0 / 26**

Characteristics

- **Flexible network size**
- **Efficient utilization of IP addresses**
- **Uses subnet masks**
- **Supports route aggregation**

4. Differences Between Classful and Classless Addressing

Feature	Classful Addressing	Classless Addressing
Address division	Fixed classes	No classes
Flexibility	Low	High
IP utilization	Poor (wastage)	Efficient
Subnet mask	Not used	Used
Scalability	Limited	Highly scalable
Routing	Complex	Simplified (aggregation)

Feature	Classful Addressing	Classless Addressing
Example	Class A, B, C	CIDR, VLSM

5. Conclusion

- **Classful addressing is simple but inefficient.**
- **Classless addressing overcomes IP wastage and supports modern Internet routing.**
- **Hence, classless addressing is widely used today**

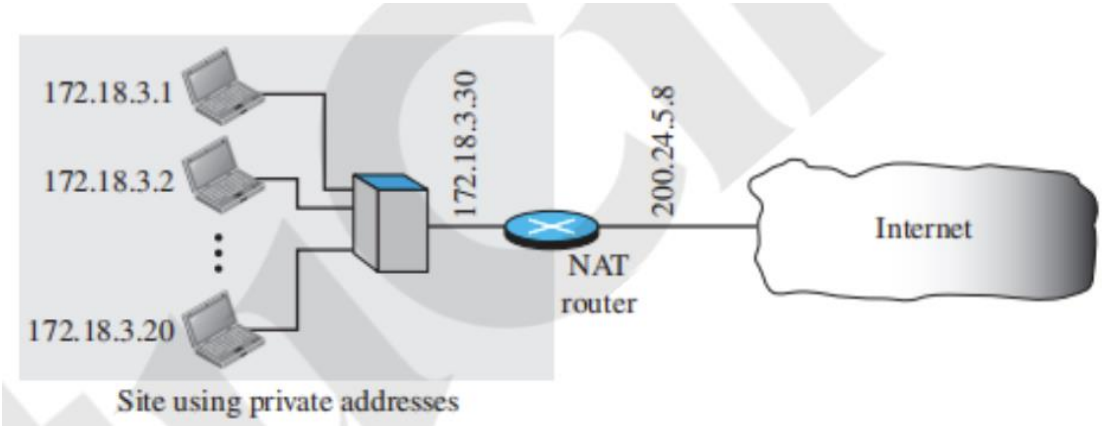
Q5 c) Explain Network Address Translation (NAT) with a neat diagram.

Network Address Translation (NAT)

Concept:
ISPs allocate limited IP address ranges to small businesses or households, but expanding these ranges is often difficult due to neighboring allocations. Not all devices in a network require simultaneous Internet access. For example, a small business with 20 computers may need only 4 to access the Internet at once, while the rest handle internal tasks.

Solution:

- **Use private IP addresses for internal communication.**
- **Use a few global IP addresses from the ISP for external communication.**
- **NAT allows a network to use private addresses internally and maps them to global addresses for Internet access through a NAT-enabled router.**



Example IPs in a Private Network:

- **172.18.3.1**
- **172.18.3.2**
- **172.18.3.20**

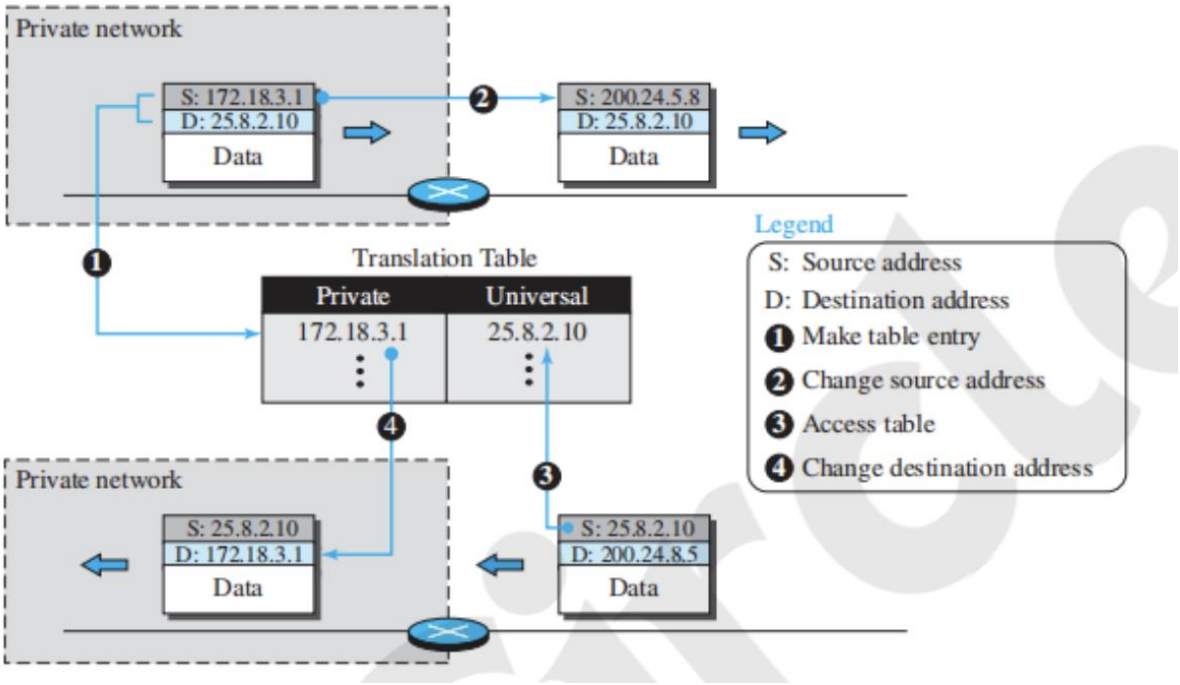
- **172.18.3.30**

NAT Router:

- **Private side: one private address from the internal network**
- **Public side: one global address from ISP (e.g., 200.34.5.8)**

Network Visibility:

- **The private network is invisible to the rest of the Internet.**
- **The Internet sees only the NAT router with the global IP address (e.g., 200.34.5.8).**



Working of NAT:

1. **Outgoing Packets:**
    - **All outgoing packets pass through the NAT router.**
    - **The router replaces the source address (private IP) with the global NAT address.**
    - **Records the destination address in a translation table.**
2. **Incoming Packets:**
    - **Incoming packets pass through the NAT router.**
    - **The router replaces the destination (global) address with the corresponding private address using the translation table.**

Key Points:

- **NAT enables multiple devices in a private network to share a single global IP address.**
- **Translation tables maintain the mapping between private and global addresses.**

- **Helps conserve global IP addresses and secures internal networks.**

Advantages of NAT

- **Conserves public IP addresses**
- **Provides basic security**
- **Easy to implement**
- **Reduces ISP dependency**

Disadvantages of NAT

- **Breaks end-to-end connectivity**
- **Causes issues with some protocols**
- **Additional processing overhead**

Conclusion

NAT plays a crucial role in modern networks by enabling private networks to communicate with the Internet efficiently while conserving IPv4 addresses and enhancing security.

Q6 a) Explain the IPv6 packet format in detail.

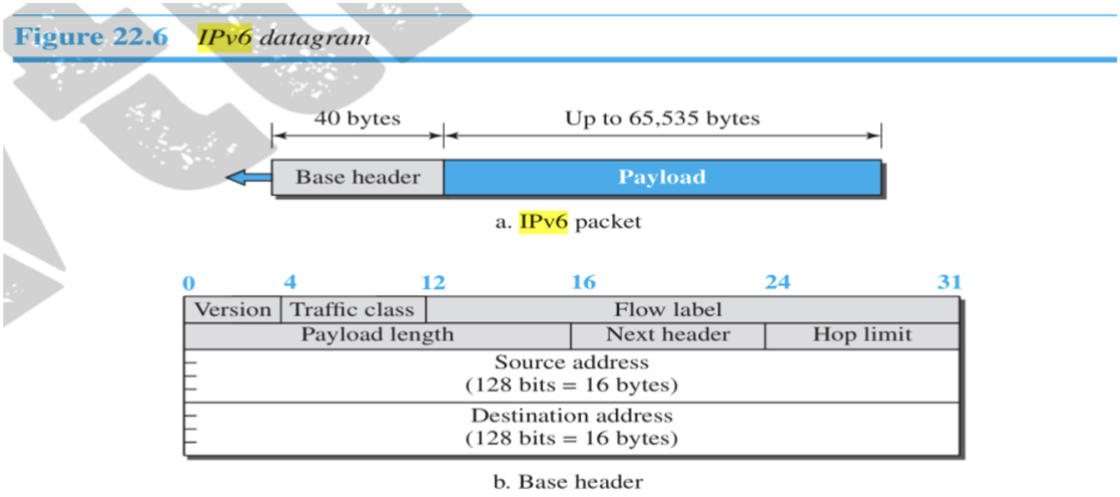
Introduction

An IPv6 datagram is the basic unit of data transmission in an IPv6 network.
IPv6 was designed to overcome the limitations of IPv4, especially address exhaustion, and to improve routing efficiency, scalability, and QoS support.

An IPv6 datagram consists of:

- **A fixed-length header (40 bytes)**
- **A payload containing upper-layer data**

IPv6 Datagram Structure



IPv6 Header Fields

1. Version (4 bits)

- **Indicates the IP version.**
- **Value is 6 for IPv6 (0110 in binary).**

2. Traffic Class (8 bits)

- **Used for Quality of Service (QoS).**
- **Helps prioritize traffic such as voice, video, or data.**
- **Similar to the Type of Service (ToS) field in IPv4.**

3. Flow Label (20 bits)

- **Identifies packets belonging to the same flow.**
- **Enables special handling for real-time applications like VoIP and video streaming.**

4. Payload Length (16 bits)

- **Specifies the length of the payload in bytes.**
- **Does not include the 40-byte IPv6 header.**

5. Next Header (8 bits)

- **Indicates the type of header that follows the IPv6 header.**

- **Can specify:**
    - **Upper-layer protocols (TCP, UDP, ICMPv6)**
    - **Extension headers**

Examples:

- **6 → TCP**
- **17 → UDP**
- **58 → ICMPv6**

6. Hop Limit (8 bits)

- **Prevents packets from looping indefinitely.**
- **Decremented by each router.**
- **Packet is discarded when hop limit reaches 0.**
- **Similar to TTL in IPv4.**

7. Source Address (128 bits)

- **IPv6 address of the sender.**
- **Written in hexadecimal format.**

8. Destination Address (128 bits)

- **IPv6 address of the receiver.**
- **Identifies the final destination of the packet.**

IPv6 Datagram Payload

- **Contains the actual data being transmitted.**
- **Structure depends on the Next Header field.**
    - **TCP → TCP segment**
    - **UDP → UDP datagram**
    - **ICMPv6 → Control message**

Key Features of IPv6 Datagram Format

- **Fixed 40-byte header**

- **No header checksum → faster processing**
- **Supports extension headers**
- **Efficient routing**
- **Better support for QoS and security**
- **128-bit addressing enables massive address space**

Conclusion

The IPv6 datagram format uses a simplified and fixed-length header that improves routing efficiency, scalability, and performance. With features like flow labeling, large address space, and QoS support, IPv6 is well-suited for modern Internet applications.

Q6 b) Discuss Distance Vector (D-V) routing, highlighting the importance of distance vectors.

1. Definition

Distance Vector (D-V) routing is a dynamic routing algorithm in which each router:

- **Maintains a routing table containing the distance (cost) to each destination and the next hop.**
- **Periodically shares its routing table with its immediate neighbors.**
- **Updates its table based on the information received from neighbors.**

Key idea: "Tell me what you know about distances to all destinations, and I'll update my table accordingly."

2. Working of Distance Vector Routing

1. **Initialization:**
   Each router knows the cost to its direct neighbors and sets the cost to all other destinations as infinity (∞).
2. **Distance Vector Exchange:**
   - **Routers periodically send their distance vectors (routing tables) to neighboring routers.**
3. **Update Rule (Bellman-Ford Algorithm):**
   For each destination D, the router calculates:

$$\text{Cost to D via neighbor N} = \text{Cost to N} + \text{Neighbor's Cost to D}$$

   - **If this new cost is less than the current cost, the routing table is updated.**
4. **Iteration:**

- Repeat exchanges until routing tables converge (no more changes).

3. Distance Vector Table Example

Destination	Cost	Next Hop
A	0	—
B	1	B
C	3	B
D	2	D

- The distance vector sent to neighbors is essentially the cost to reach all destinations.

4. Importance of Distance Vectors

1. Efficient Routing:
   - Routers know the shortest path to all destinations based on neighbor information.
2. Simplicity:
   - Easy to implement using Bellman-Ford algorithm.
3. Scalability:
   - Works well in small to medium-sized networks.
4. Dynamic Adaptation:
   - Can adapt to topology changes (link failures or new links).
5. Foundation for Protocols:
   - Basis for protocols like RIP (Routing Information Protocol).

5. Advantages of D-V Routing

- Simple and easy to implement
- Requires limited computational resources
- Routers only need information from direct neighbors

6. Disadvantages

- Slow convergence in large networks (count-to-infinity problem)
- Routing loops possible if not carefully handled

- **Not suitable for very large networks**

Q6 c) Describe the BGP (Border Gateway Protocol) in detail.

1. Definition

BGP (Border Gateway Protocol) is a path-vector routing protocol used to exchange routing information between autonomous systems (ASes) on the Internet.

- **It is classified as an Exterior Gateway Protocol (EGP).**
- **It enables the Internet to function as a network of networks.**
- **BGP makes routing decisions based on path, policies, and rules, rather than just distance.**

2. Purpose of BGP

- **Connect multiple autonomous systems (AS) efficiently.**
- **Determine the best path for data packets across the Internet.**
- **Handle policy-based routing, allowing organizations to control traffic flow.**

3. Key Features

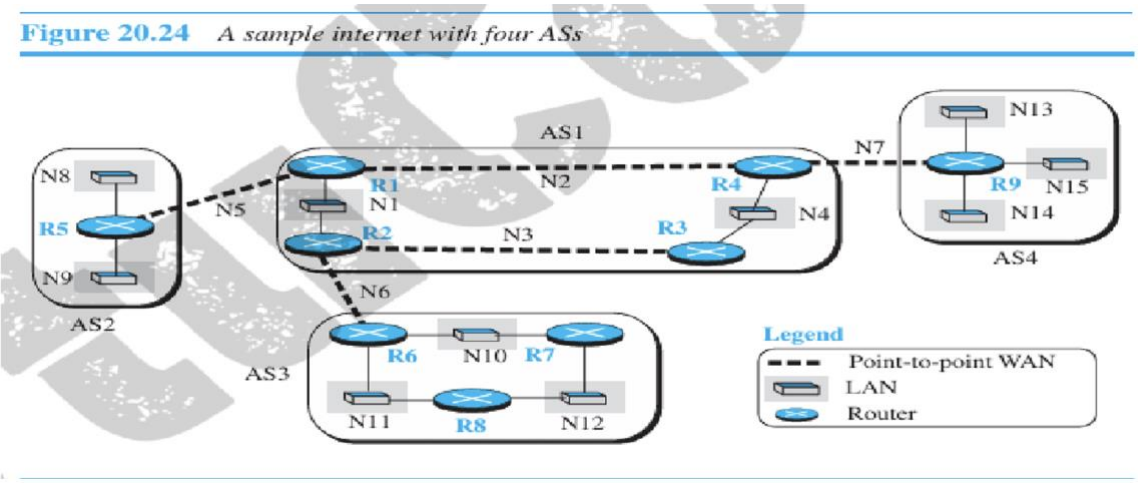
1. **Path-Vector Protocol**
    - **Keeps a list of ASes (AS_PATH) a route traverses.**
    - **Helps prevent routing loops.**
2. **TCP-based**
    - **BGP uses TCP port 179 for reliable communication between routers.**
3. **Policy-Based Routing**
    - **Network administrators can define routing policies (prefer one path over another).**
4. **Scalable**
    - **Designed for large networks, including the global Internet.**
5. **Loop-Free**
    - **Uses AS_PATH information to avoid loops between ASes.**

4. Types of BGP

1. **Internal BGP (iBGP)**
    - **BGP between routers within the same AS.**

- Used for internal routing consistency.

2. External BGP (eBGP)
   - BGP between routers in different ASes.
   - Used to exchange routing information between organizations or ISPs.



5. How BGP Works

1. Neighbor Establishment
   - Routers form BGP peers (neighbors) using TCP.
   - Exchange full routing tables initially.
2. Route Advertisement
   - Each router advertises networks it can reach and AS_PATH.
3. Route Selection
   - Router selects the best path using BGP rules:
     - Highest weight
     - Shortest AS_PATH
     - Local preference
     - Lowest origin type
     - Lowest MED (Multi-Exit Discriminator)
4. Route Update
   - Updates are sent only when there is a change.
   - Reduces unnecessary network traffic.

6. BGP Message Types

Message Type	Purpose
OPEN	Establishes a connection with neighbor
UPDATE	Advertises new routes or withdraws old routes
KEEPALIVE	Maintains session between peers
NOTIFICATION	Reports errors and closes session

7. Advantages of BGP

- **Scalable for global Internet routing**
- **Policy-based routing gives flexibility**
- **Prevents routing loops with AS_PATH**
- **Reliable due to TCP connection**

8. Disadvantages

- **Complex configuration and management**
- **Slower convergence compared to IGP (Interior Gateway Protocols)**
- **Requires more memory and processing power**

9. Summary

BGP is a path-vector protocol that connects multiple autonomous systems, ensuring loop-free, policy-based, and scalable Internet routing. Its AS_PATH mechanism and TCP reliability make it the backbone of Internet routing.

Module -4

7a. Explain the concept of port numbers mentioning ICANN ranges.

In the transport layer, communication takes place between processes running on different hosts. To identify a particular process on a host, the transport layer uses **port numbers** in addition to IP addresses.

A **port number** is a 16-bit integer ranging from **0 to 65,535** that uniquely identifies a process in a host. While the IP address identifies the destination host, the port number identifies the **specific application process** running on that host. Thus, process-to-process communication is achieved using the combination of IP address and port number, called a **socket address**.

Client processes usually use **ephemeral (temporary) port numbers**, whereas server processes use **well-known port numbers** so that clients can easily locate them.

According to ICANN, port numbers are divided into three ranges:

1. **Well-known ports (0 – 1023):**
   These ports are assigned and controlled by ICANN and are reserved for standard services such as HTTP, FTP, SMTP, etc.
2. **Registered ports (1024 – 49,151):**
   These ports are not controlled by ICANN but can be registered to avoid duplication. They are generally used by user applications and specific services.
3. **Dynamic or Private ports (49,152 – 65,535):**
   These ports are neither controlled nor registered. They are used as temporary ports, mainly by client processes during communication.

Thus, port numbers play an important role in identifying processes and enabling correct delivery of data at the transport layer.

7b. Explain go back n protocol

Go-Back-N is a **connection-oriented transport layer protocol** that provides both **flow control and error control** using the concept of **pipelining**. It allows the sender to transmit multiple packets before receiving acknowledgments, thereby improving channel utilization compared to the Stop-and-Wait protocol.

1. Basic Idea

In Go-Back-N, the sender can send several packets without waiting for individual acknowledgments. However, the receiver can accept only **one packet in order** and discards all out-of-order packets. If an error or loss occurs, the sender **goes back and retransmits all outstanding packets** starting from the first unacknowledged packet.

2. Sequence Numbers

Packets are numbered using **modulo 2^m sequence numbers**, where *m* is the number of bits in the sequence number field. This numbering helps in identifying lost, duplicate, and out-of-order packets.

3. Acknowledgment Numbers

Acknowledgments are **cumulative**.
If the receiver sends **ackNo = n**, it means that **all packets up to (n–1) have been received correctly** and the next expected packet is *n*.

4. Send Window

The sender maintains a **sliding send window** of maximum size **$2^m - 1$**.
The window contains:

- Packets already acknowledged
- Outstanding packets (sent but not yet acknowledged)
- Packets that can be sent next

Three variables define the send window:

- **Sf** – sequence number of first outstanding packet
- **Sn** – sequence number of next packet to send
- **Ssize** – size of the send window

When an acknowledgment arrives, the window **slides forward** accordingly

5. Receive Window

The receive window size is always **1**.
The receiver accepts only the packet with the **expected sequence number (Rn)**.
Any out-of-order packet is discarded and an acknowledgment for the expected packet is sent again.

6. Timers and Retransmission

Only **one timer** is used for the oldest outstanding packet.
If a **time-out occurs**, the sender **retransmits all outstanding packets** starting from Sf. This behavior gives the protocol its name "Go-Back-N".

7. Finite State Machine (FSM)

- The sender has **Ready** and **Blocking** states.
- The receiver always remains in the **Ready** state.
  FSMs control sending packets, receiving acknowledgments, sliding windows, and retransmissions.

8. Window Size Restriction

The size of the send window must be **less than 2^m**.
If the window size equals 2^m, duplicate packets may be **mistaken as new packets**, causing errors.

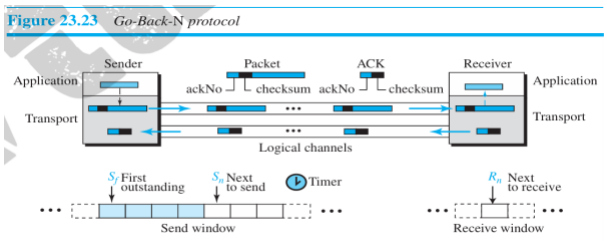
9. Advantages and Limitation

Advantages:

- Better channel utilization than Stop-and-Wait
- Supports pipelining and higher throughput

Limitation:

- On error, many correctly sent packets are retransmitted, causing inefficience



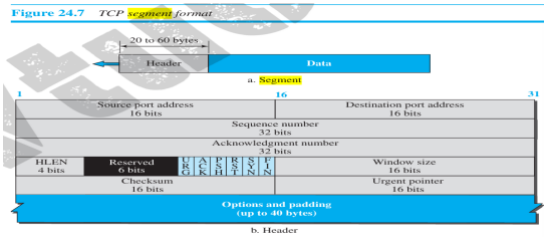
7c. Explain TCP SEGMENT format with neat diagram

The TCP segment consists of a **header** followed by **data (payload)**.
The header contains control information required for reliable, ordered and error-free delivery of data.

Explanation of Fields

1. **Source Port (16 bits)**
   Identifies the sending application process.

2. **Destination Port (16 bits)**
   Identifies the receiving application process.
3. **Sequence Number (32 bits)**
   Indicates the sequence number of the first data byte in this segment.
   Used for **ordering and reliability**.
4. **Acknowledgment Number (32 bits)**
   Contains the next sequence number expected from the sender.
   Used for **error control and flow control**.
5. **Header Length (HLEN / Data Offset)**
   Specifies the length of the TCP header in 32-bit words.
6. **Reserved Field**
   Reserved for future use and set to zero.
7. **Flags (Control Bits)**
   Includes control bits such as:
   - **URG** – Urgent pointer valid
   - **ACK** – Acknowledgment valid
   - **PSH** – Push data to application
   - **RST** – Reset connection
   - **SYN** – Synchronize sequence numbers (connection setup)
   - **FIN** – Finish connection
8. **Window Size (16 bits)**
   Specifies the number of bytes the receiver is willing to accept.
   Used for **flow control**.
9. **Checksum (16 bits)**
   Used to detect errors in header and data.
10. **Urgent Pointer (16 bits)**
    Indicates the end of urgent data when URG flag is set.
11. **Options and Padding**
    Used for additional features such as maximum segment size.
    Padding ensures the header ends on a 32-bit boundary.
12. **Data (Payload)**
    Contains the actual application data.



8a. Connection Establishment in TCP

*TCP is a **connection–oriented protocol**. Before data transmission can begin, a logical connection must be established between the client and the server. This is done using a procedure called the **three–way handshake**.*

Step 1: SYN (Connection Request)

- The **client sends a SYN segment** to the server.
- The sequence number is set to an initial value (say x).
- This indicates a request to establish a connection and synchronize sequence numbers.

Client state: SYN-SENT
Server state: LISTEN

Step 2: SYN + ACK (Acknowledgment + Request)

- The server receives the SYN and replies with a **SYN + ACK segment**.
- The acknowledgment number is set to **x + 1** (next expected sequence).
- The server also sends its own initial sequence number (say y).

Server state: SYN-RECEIVED

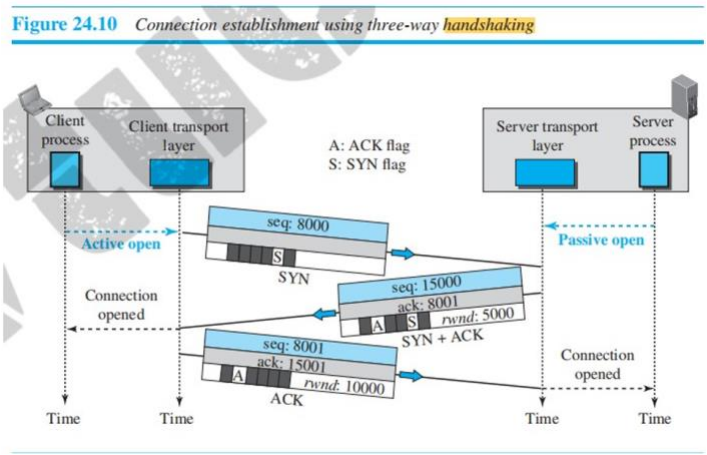
Step 3: ACK (Final Acknowledgment)

- The client sends an **ACK segment** back to the server.
- The acknowledgment number is set to **y + 1**.
- After this, the connection is successfully established.

Client state: ESTABLISHED
Server state: ESTABLISHED

Purpose of Three–Way Handshake

1. Synchronizes **sequence numbers** between sender and receiver.
2. Confirms that **both sides are ready** for data transfer.
3. Prevents **old or duplicate connection requests** from causing errors.
4. Establishes a **reliable full–duplex connection**.



8b. Explain error control in TCP acknowledgements

TCP provides reliable data transfer by using **error control mechanisms** based on **sequence numbers, acknowledgments, timers and retransmissions**.

1. **Sequence Numbers**
   Each byte of data in TCP is assigned a sequence number. This helps the receiver detect **lost, duplicate or out-of-order segments**.
2. **Acknowledgments (ACKs)**
   TCP uses **positive cumulative acknowledgments**.

An acknowledgment number indicates the **next byte expected** by the receiver.
If ACK = n, it means all bytes up to (n – 1) have been received correctly.

3. **Timeout and Retransmission**
   The sender starts a **timer** after sending a segment.
   If an acknowledgment is not received before the timer expires, the sender **retransmits the lost segment**.
4. **Duplicate ACKs and Fast Retransmission**
   When out-of-order segments are received, the receiver sends **duplicate ACKs**.
   After receiving three duplicate ACKs, the sender performs **fast retransmission** without waiting for timeout.

8c. discuss THREE algorithms for handling congestion in TCP

TCP controls congestion using a set of algorithms that regulate the sender's transmission rate according to the network condition. The main algorithms used are **Slow Start, Congestion Avoidance, and Fast Recovery (with Fast Retransmit)**.

1. Slow Start Algorithm

Slow Start is used at the beginning of a connection or after a timeout.

- TCP maintains a variable called **congestion window (cwnd)**.
- Initially, cwnd is set to **1 Maximum Segment Size (MSS)**.
- For every acknowledgment received, cwnd is **doubled every round-trip time** (exponential growth).
- Transmission rate increases rapidly until cwnd reaches a threshold called **slow start threshold (ssthresh)** or congestion occurs.

Purpose:
To probe the network capacity gradually and avoid sudden congestion.

2. Congestion Avoidance Algorithm

When cwnd reaches the slow start threshold, TCP enters congestion avoidance mode.

- In this phase, cwnd increases **linearly** instead of exponentially.
- For every round-trip time, cwnd is increased by **one MSS**.
- If congestion is detected (timeout or packet loss), TCP reduces cwnd and updates ssthresh.

Purpose:
To maintain stable transmission and prevent congestion by increasing the sending rate slowly.

3. Fast Retransmit and Fast Recovery

These algorithms are used when congestion is detected through **duplicate acknowledgments**.

Fast Retransmit

- If the sender receives **three duplicate ACKs**, it assumes a packet is lost.
- The sender **retransmits the missing segment immediately** without waiting for timeout.

Fast Recovery

- After fast retransmission, cwnd is reduced to **half of its previous value**.
- TCP does not enter slow start again. Instead, it continues with **congestion avoidance**.

Purpose:
To quickly recover from packet loss and avoid unnecessary slow start, thereby improving performance.

Module – 5

9a. Discuss application layer paradigms with neat diagram

Introduction

Application layer paradigms define the relationship and communication style between two application programs that exchange messages over the Internet. These paradigms determine how services are requested and provided between processes. The two main paradigms are **Client–Server paradigm** and **Peer–to–Peer paradigm**. A third approach called the **Mixed paradigm** combines features of both.

1. Client–Server Paradigm

In the client–server paradigm, one process acts as a **server** and provides services, while the other acts as a **client** and requests services.

Characteristics:

- The server runs continuously and waits for client requests.
- The client is started only when service is required.
- The server must be powerful since many clients may connect simultaneously.
- Communication load is concentrated on the server.

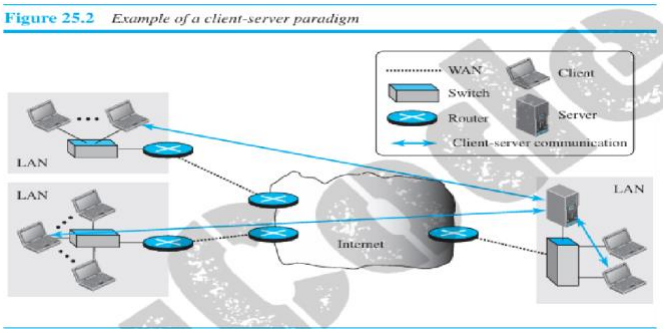
Examples: HTTP (Web), FTP, E-mail, SSH.

Advantages:

- Easy to manage and control.
- Better security since server controls access.

Disadvantages:

- Server overload possible.
- High cost of maintaining powerful servers.



2. Peer–to–Peer (P2P) Paradigm

In the peer–to–peer paradigm, there is **no permanent server**. Each computer (peer) can both request and provide services.

Characteristics:

- Responsibility is shared among peers.
- Any peer can act as a client or a server at different times.
- Scalable and cost-effective.
- Security is more difficult to maintain.

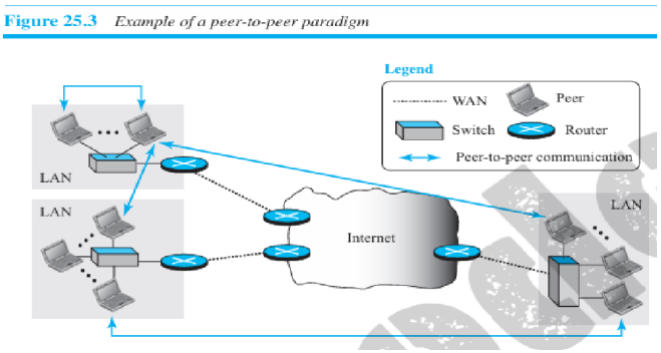
Examples: BitTorrent, Skype, Internet telephony, IPTV.

Advantages:

- No need for expensive dedicated servers.
- Highly scalable.

Disadvantages:

- Difficult to implement security.
- Not suitable for all applications.



3. Mixed Paradigm

In the mixed paradigm, both client–server and peer–to–peer models are combined.

Explanation:

- Initially, a light client–server communication is used to locate a peer.

- After locating the peer, the actual data transfer takes place using peer–to–peer communication.

Advantage:

- Combines the control of client–server with the scalability of peer–to–peer.

9b. Explain the use of sockets in process-to-process communication

Introduction

In client–server communication, interaction takes place between two running application programs called **processes**. This communication is achieved using an interface known as a **socket**. A socket acts as an end-point of communication between two processes running on different hosts in a network.

Definition of Socket

A socket is an **abstraction** created by an application program to send and receive data. It behaves like a source or sink similar to a file or terminal, but it is not a physical entity. It provides an interface between the **application layer** and the **transport layer** through the operating system.

Position of Socket in Protocol Stack

The socket interface lies between the **application layer** and the **transport layer**. It allows application programs to access the services of the transport layer (TCP or UDP) through the operating system.

Process-to-Process Communication Using Sockets

Communication between a client process and a server process is carried out through **two sockets**, one at each end.

- The client process creates a socket and sends a request through it.
- The server process also creates a socket, receives the request, processes it, and sends back a response.
- Each process considers the socket as the entity that sends and receives messages.
- The operating system and TCP/IP protocol handle the actual data transmission.

Socket Address

Each socket is identified by a **socket address**, which consists of:

- **IP address (32 bits)** – identifies the host
- **Port number (16 bits)** – identifies the process

Thus,
Socket Address = IP Address + Port Number

A complete communication is identified by a **pair of socket addresses**:

- Local socket address
- Remote socket address

Finding Socket Addresses

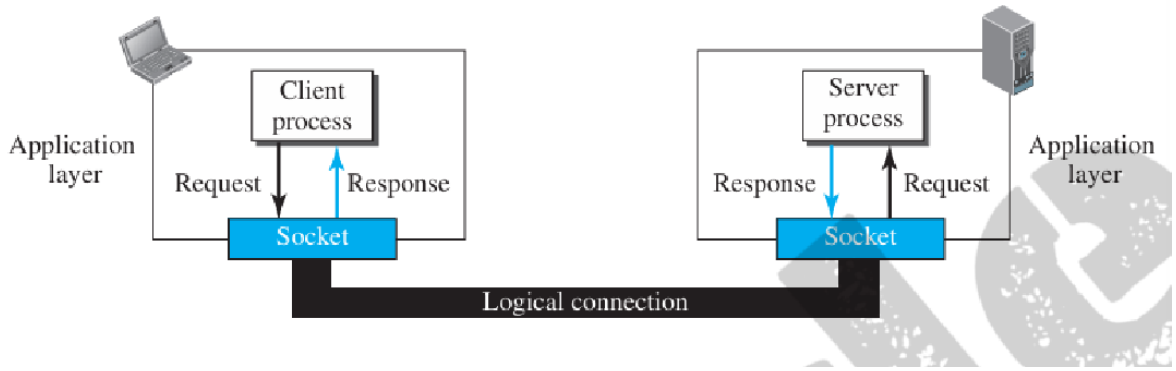
At Server Side:

- Local socket address is fixed and known (well-known port number).
- Remote socket address is obtained when a client sends a request.

At Client Side:

- Local socket address uses a temporary (ephemeral) port number assigned by the OS.
- Remote socket address is obtained using the server's name and DNS service.

Figure 25.6 *Use of sockets in process-to-process communication*



9c. Discuss Connection Types in HTTP along with Format of Messages

Introduction

The HyperText Transfer Protocol (HTTP) is a client–server protocol used to retrieve web pages from the World Wide Web. HTTP uses the services of **TCP**, which is a reliable and connection-oriented protocol. Depending on how connections are maintained, HTTP supports **two types of connections**:

1. **Nonpersistent connections**
2. **Persistent connections**

HTTP also defines standard **request and response message formats** for communication between client and server.

1. Connection Types in HTTP

(a) Nonpersistent Connection

In a nonpersistent connection, **one TCP connection is used for only one request and one response**.

Working:

1. Client opens a TCP connection to the server.
2. Client sends a request.
3. Server sends the response and closes the connection.
4. Client closes the connection after receiving data.

If a web page contains multiple objects (images, files, etc.), a **separate connection is opened for each object**.

Characteristics:

- Default in HTTP version 1.0
- High overhead due to repeated connection establishment and termination
- More delay because of multiple handshakes
- Requires more server resources

b) Persistent Connection

In a persistent connection, **the same TCP connection is used for multiple requests and responses**.

Working:

- After sending a response, the server keeps the connection open.
- The client can send multiple requests over the same connection.
- Connection is closed only when requested or after a time-out.

Characteristics:

- Default in HTTP version 1.1
- Reduces overhead and delay
- Saves time and resources
- Only one connection setup and termination needed

2. Format of HTTP Messages

HTTP defines two types of messages:

1. **Request Message**
2. **Response Message**

Each message consists of **four parts**:

- Start line
- Header lines
- Blank line
- Body (optional)

2. Format of HTTP Messages

HTTP defines two types of messages:

1. **Request Message**
2. **Response Message**

Each message consists of **four parts**:

- Start line
- Header lines
- Blank line

- Body (optional)

The first line is called the **Request Line**. **Fields in Request Line:**

- **Method** – GET, POST, PUT, HEAD, DELETE, etc.
- **URL** – Address of the requested resource
- **Version** – HTTP/1.1

Common Request Methods:

- GET – Requests a document
- HEAD – Requests header information only
- PUT – Sends a document to server
- POST – Sends data to server
- DELETE – Removes a web page

(b) HTTP Response Message Format

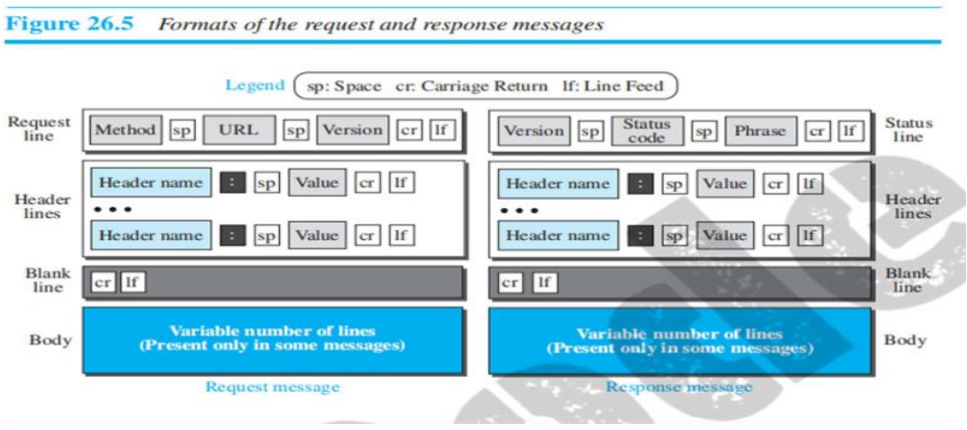
The first line is called the **Status Line**.

Fields in Status Line:

- **Version** – HTTP/1.1
- **Status Code** – Indicates result of request
  - 200 – OK (Success)
  - 304 – Not Modified
  - 404 – Not Found
  - 500 – Server Error
- **Phrase** – Text explanation of status

Common Response Headers:

- Date
- Server
- Content-Length
- Content-Type
- Last-Modified



10a. Explain POP and IMAP protocols.

Introduction

Electronic mail systems use specific protocols to **retrieve e-mails from a mail server to a client**. Two important mail access protocols are:

- **POP (Post Office Protocol)**
- **IMAP (Internet Message Access Protocol)**

These protocols operate at the **application layer** and are used after the mail is delivered to the receiver's mail server using SMTP.

1. POP (Post Office Protocol)

POP is a simple protocol used by a mail client to **download e-mails from the mail server to the local computer**.

Features of POP:

- Works in **download-and-delete** mode by default.
- Once messages are downloaded, they are normally **removed from the server**.
- Suitable for users who access mail from **one computer only**.
- Uses **TCP port number 110** (POP3).

Phases of POP Operation

POP works in **three phases**:

(a) Authorization Phase

- Client sends username and password to the server.
- Server authenticates the user.

(b) Transaction Phase

- Client retrieves messages from the mailbox.
- Messages can be marked for deletion.

(c) Update Phase

- Client quits the session.
- Server deletes the marked messages permanently.

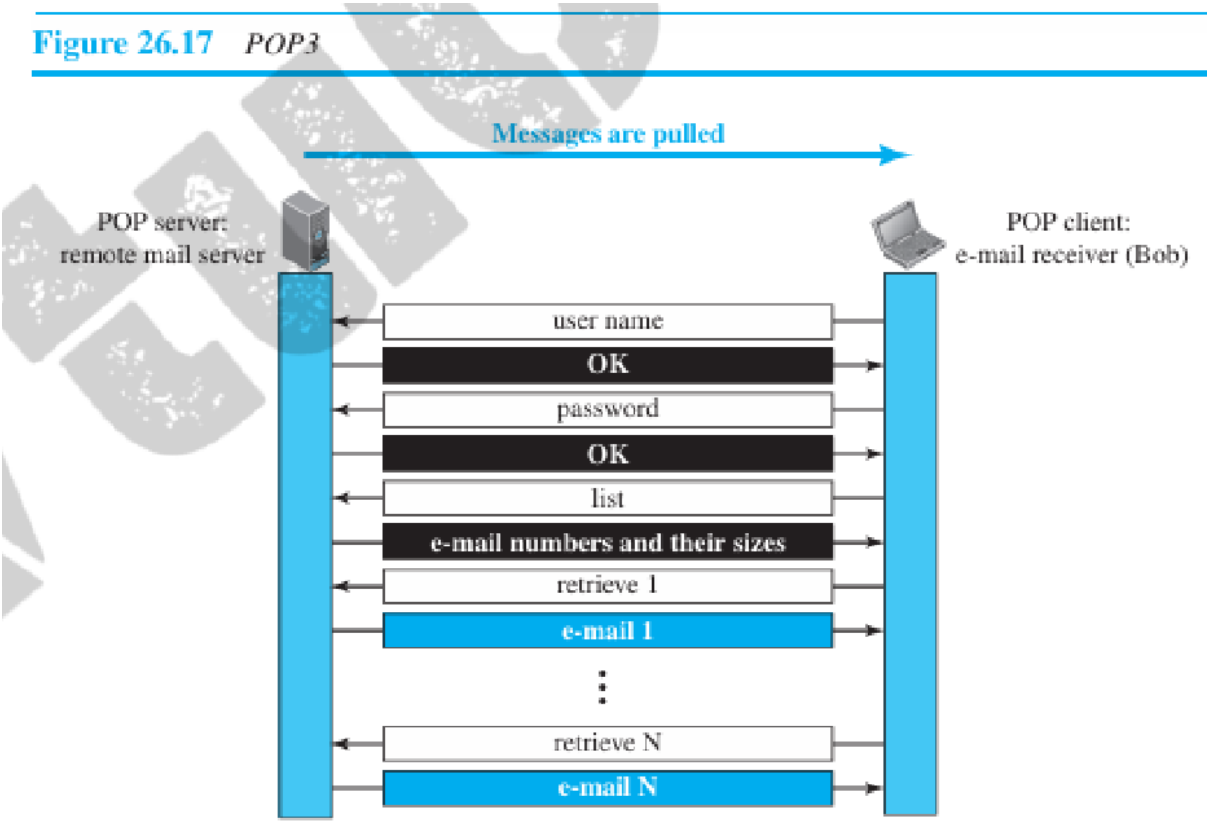
Advantages of POP

- Simple and easy to implement.
- Allows offline reading of mails.

Disadvantages of POP

- Messages are removed from server (may cause data loss).
- Not suitable for accessing mail from multiple devices.

- No support for folder management on server.



2. IMAP (Internet Message Access Protocol)

IMAP is an advanced mail access protocol that allows the client to **manage and access e-mails directly on the mail server**.

Features of IMAP:

- Messages remain **stored on the server**.
- Supports **multiple folders and mailboxes**.
- Suitable for users who access mail from **multiple devices**.
- Allows selective downloading of headers or parts of messages.
- Uses **TCP port number 143** (IMAP4).

Working of IMAP

- Client connects to the server and authenticates.
- Messages are kept on the server and synchronized with the client.
- Users can create, delete, and rename folders on the server.
- Same mailbox view is available from different devices.

Advantages of IMAP

- Mail remains safely stored on the server.
- Supports multi-device access.
- Provides folder management and message searching.

Disadvantages of IMAP

- Requires continuous server connection.
- More complex and consumes more server storage.

10b. Discuss Applications of SSH Protocol

Applications of SSH

1. **Remote Login**
   SSH is used to log into a remote computer securely. It replaces insecure protocols like Telnet by encrypting usernames, passwords, and commands during transmission.
2. **Remote Command Execution**
   SSH allows users to execute commands on a remote system securely and receive the output without exposing data to attackers.
3. **Secure File Transfer**
   SSH supports secure file transfer protocols such as **SCP (Secure Copy Protocol)** and **SFTP (Secure File Transfer Protocol)** to transfer files safely between systems.
4. **Tunneling and Port Forwarding**
   SSH is used to create encrypted tunnels for forwarding ports and securing other application communications such as database access and web traffic.

10c. Explain Resolution in DNS

Introduction

The Domain Name System (DNS) is a distributed database that maps **domain names to IP addresses**. The process of translating a domain name into its corresponding IP address is called **name resolution** or **DNS resolution**. This process is required whenever a client wants to communicate with a host using its domain name.

1. Recursive Resolution

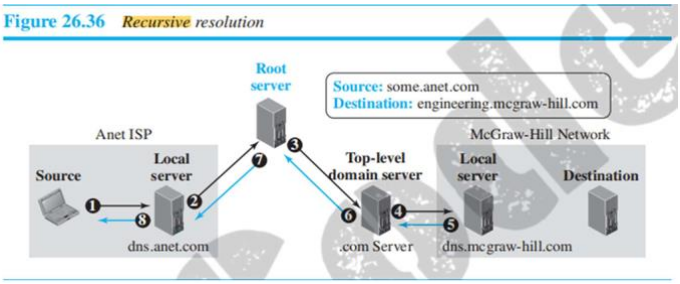
In recursive resolution, the **client requests the DNS server to find the complete answer**.

Working:

- The client sends a query to the local DNS server.
- The local DNS server is responsible for contacting other DNS servers (root, TLD, and authoritative servers).
- The final IP address is returned directly to the client.

Characteristics:

- Entire responsibility lies with the DNS server.
- Simple for the client.
- Increases load on DNS servers.



2. Iterative Resolution

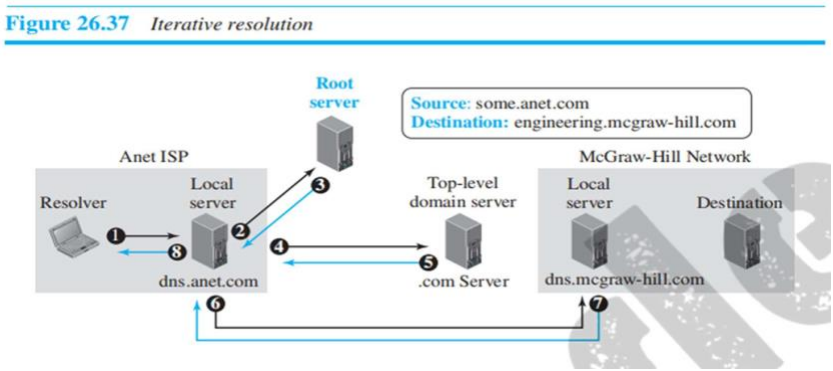
In iterative resolution, the DNS server gives the **best possible answer at each step**, and the client continues querying other servers.

Working:

1. Client sends a query to the local DNS server.
2. Local DNS server replies with the address of a root server.
3. Client queries the root server, which returns the address of the TLD server.
4. Client queries the TLD server, which returns the address of the authoritative server.
5. Client queries the authoritative server to get the final IP address.

Characteristics:

- Client performs most of the work.
- Reduces load on DNS servers.
- More messages are exchanged.



Complete DNS Resolution Process

When a client enters a domain name (e.g., www.example.com), the following steps occur:

1. The client sends a query to the **local DNS server**.
2. If the local DNS server has the answer in its cache, it replies immediately.
3. If not, it contacts the **root DNS server**.
4. The root server returns the address of the appropriate **TLD server** (.com, .org, etc.).
5. The TLD server returns the address of the **authoritative DNS server** for the domain.
6. The authoritative server returns the **IP address** of the requested host.
7. The local DNS server sends the IP address back to the client and stores it in cache.