

1.

a) What is computer network? Describe the Network Criteria.[3]

A **computer network** is a system of interconnected computers and other devices (like printers, servers, etc.) that are linked together to share resources, data, and applications. These connections can be made using wired (like Ethernet cables) or wireless (like Wi-Fi) communication channels.

Network Criteria

To ensure a computer network functions effectively, it must meet the following **three key criteria**:

1. Performance

- Refers to how well the network operates.
- Depends on:
 - **Transmission time** – time to transfer data from source to destination.
 - **Response time** – time between sending a request and receiving a response.
 - **Throughput** – number of messages successfully delivered per unit time.
 - **Latency** – delay between sending and receiving data.

2. Reliability

- Measures the dependability of the network.
- Includes:
 - **Downtime** – how often the network fails.
 - **Failure recovery** – how fast the network recovers from a fault.
 - **Accuracy** – ensuring data is transmitted without errors.

3. Security

- Ensures that data is protected from unauthorized access and threats.
- Involves:
 - **Confidentiality** – keeping data private.
 - **Integrity** – ensuring data is not altered.
 - **Availability** – ensuring network services are accessible when needed.

b) What are the advantages of a multi-point connection over a P2P connection?[2]**Advantages of a Multi-point Connection over a Point-to-Point (P2P) Connection**

In a **point-to-point (P2P)** connection, a single link connects exactly two devices.

In a **multipoint** connection, more than two devices share a single link.

Advantages of Multi-point Connection over P2P:**1. Cost-Efficient**

- Fewer cables and ports are required, reducing hardware and installation costs.
- Ideal for setups where many devices need to communicate.

2. Resource Sharing

- Multiple devices can share the same medium (bandwidth, printer, storage, etc.), making better use of available resources.

3. Simplified Wiring

- Less cabling compared to multiple P2P connections, especially in large networks.

4. Easier Expansion

- New devices can be added to the network without significant changes to the existing infrastructure.

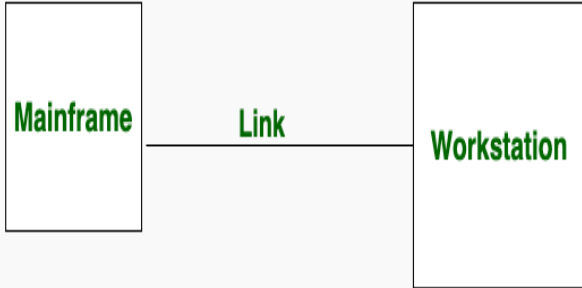
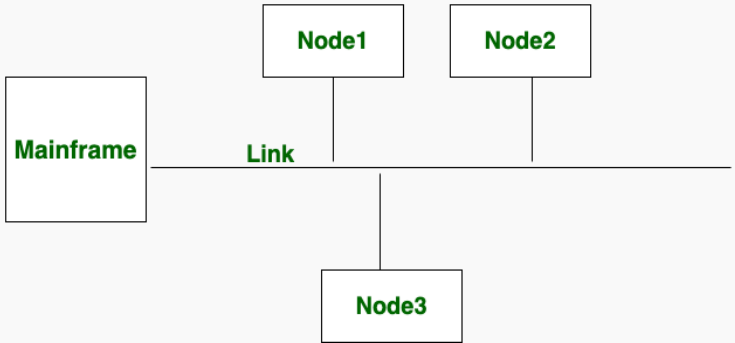
5. Space Saving

- Reduces physical clutter due to fewer wires and connectors.

Example:

- A classroom network where multiple computers are connected to a single projector using a shared line (multipoint) instead of connecting each computer directly to the projector (P2P).

Extra

S.NO	Point to point communication	Multipoint Communication
1.	Point to point communication means the channel is shared between two devices.	Multipoint Communication means the channel is shared among multiple devices or nodes.
2.	In this communication, There is dedicated link between two nodes.	In this communication, link is provided at all times for sharing the connection among nodes.
3.	In this communication, the entire capacity is reserved between these connected two devices with the possibility of waste of network bandwidth/ resources.	In this communication, the entire capacity isn't reserved by any two nodes and the network bandwidth is maximum utilized.
4.	In this communication, there is one transmitter and one receiver.	In this communication, there is one transmitter and many receivers.
5.	In point-to-point connections, the smallest distance is most important to reach the receiver.	In Multi-point connections, the smallest distance is not important to reach the receiver.
6.	Point-to-point communication provides security and privacy because communication channel is not shared.	Multi-point communication does not provide security and privacy because communication channel is shared.
7.		

c) Define protocol and Standards in Computer networks.[3]**Protocol**

A **protocol** is a set of rules and conventions that define how devices on a network communicate with each other. It governs the format, timing, sequencing, and error control of messages exchanged between devices.

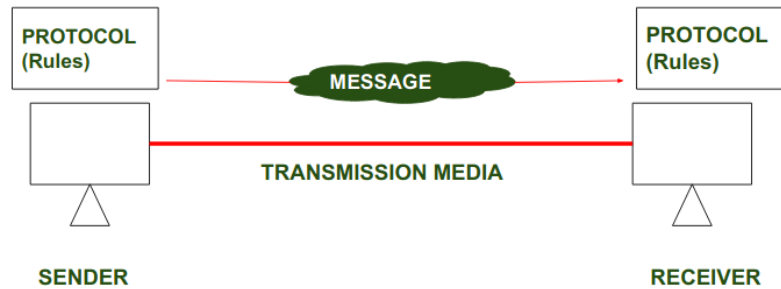
Key Functions of Protocols:

- Data formatting and encoding
- Error detection and correction
- Data compression
- Flow and congestion control
- Ensuring successful message delivery

Examples:

- **TCP (Transmission Control Protocol)** – reliable data transfer

- **IP (Internet Protocol)** – addressing and routing
- **HTTP (HyperText Transfer Protocol)** – web communication



Protocol

Standards

Standards are established guidelines developed by recognized organizations to ensure **interoperability**, **compatibility**, and **uniformity** among different hardware and software systems in networking.

Purpose of Standards:

- Allow devices from different manufacturers to work together
- Ensure consistent quality and performance
- Promote technology growth and adoption

Standard Organizations:

- **IEEE** (Institute of Electrical and Electronics Engineers)
- **ISO** (International Organization for Standardization)
- **IETF** (Internet Engineering Task Force)
- **ITU-T** (International Telecommunication Union – Telecommunication Standardization Sector)

Examples:

- **IEEE 802.3** – Ethernet standard
- **IEEE 802.11** – Wi-Fi standard
- **ISO/OSI Model** – network communication framework

d)

What do you mean by ARPANET? Describe the physical topology of computer networks.[4]

ARPANET (Advanced Research Projects Agency Network) was the **first operational packet-switching network** and the **precursor to the modern Internet**.

It was developed in the **late 1960s** by the **U.S. Department of Defense's ARPA (now DARPA)** to connect universities and research centers, enabling resource and information sharing.

Key Features of ARPANET:

- Introduced **packet switching**, which breaks data into packets before transmission.
- Initially connected four nodes: UCLA, Stanford, UC Santa Barbara, and University of Utah (1969).
- Led to the development of **TCP/IP**, which became the foundation of the Internet.
- Officially decommissioned in **1990**, but its legacy continues as the backbone of Internet architecture.

Characteristics of ARPANET

1. It is basically a type of WAN.
2. It used the concept of a packet-switching network.
3. It used Interface Message Processors(IMP)s for sub-netting.
4. ARPANETs software was split into two parts- a host and a subnet.

Advantages of ARPANET

- ARPANET was designed to serve even in a Nuclear Attack.
- It was used for collaborations through E-mails.
- It created an advancement in the transfer of important files and data for defense.

Limitations of ARPANET

- Increased number of LAN connections resulted in difficulty handling.
- It was unable to cope-up with advancement in technology.

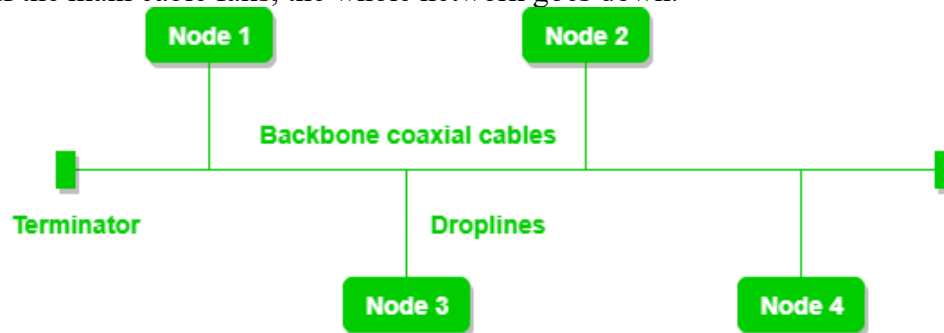
Physical Topology of Computer Networks(<https://www.geeksforgeeks.org/types-of-network-topology/>)

Physical topology refers to the **actual layout** of devices (nodes) and cables (media) in a network.

Common Types of Physical Topologies:

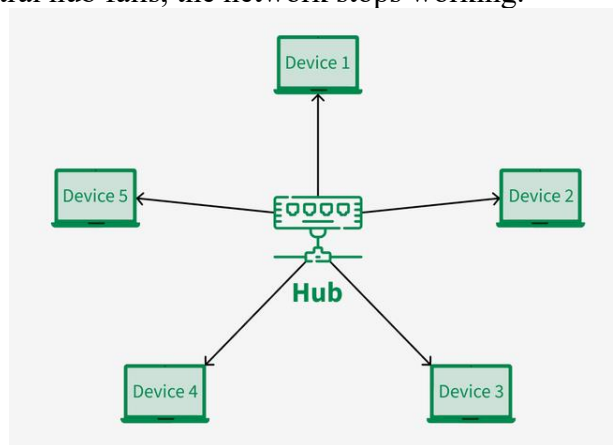
1. Bus Topology

- All devices are connected to a single backbone cable.
- **Pros:** Easy to set up, requires less cable.
- **Cons:** If the main cable fails, the whole network goes down.



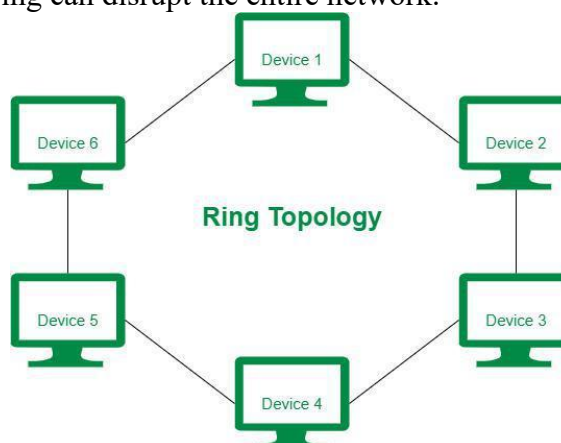
2. Star Topology

- All devices are connected to a central hub or switch.
- **Pros:** Easy to manage and troubleshoot.
- **Cons:** If the central hub fails, the network stops working.



3. Ring Topology

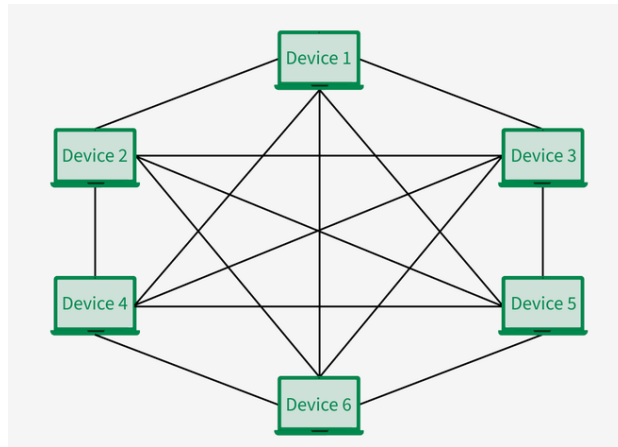
- Devices are connected in a circular loop.
- **Pros:** Simple data flow, good performance under load.
- **Cons:** A break in the ring can disrupt the entire network.



AVAILABLE AT:

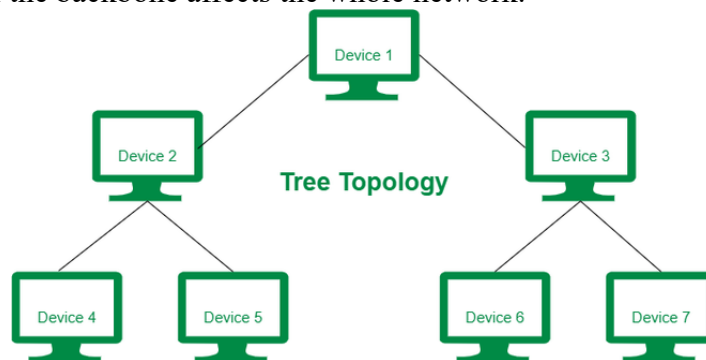
4. Mesh Topology

- Every device is connected to every other device.
- **Pros:** High fault tolerance and reliability.
- **Cons:** Expensive and complex to install and maintain.



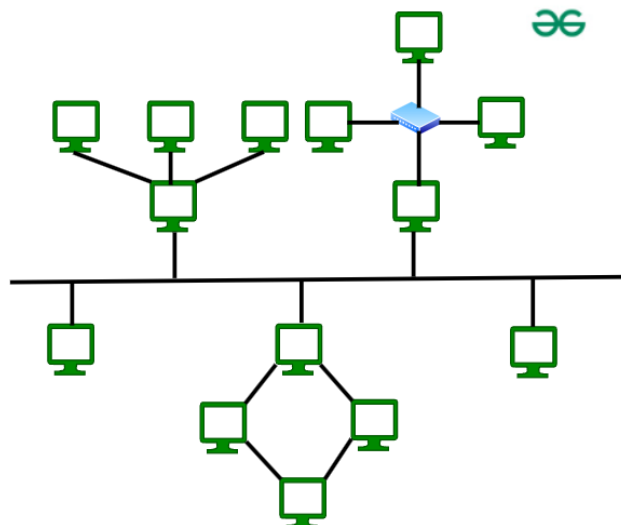
5. Tree Topology

- Combination of star and bus topologies (hierarchical).
- **Pros:** Scalable and easy to manage.
- **Cons:** Failure of the backbone affects the whole network.



6. Hybrid Topology

- Combination of two or more topologies.
- **Pros:** Flexible and reliable.
- **Cons:** Complex design and costly.



2. a) In the Go-Back-N protocol, the size of the send window can be $2^m - 1$, while the size of the receive window is only 1. How can flow control be accomplished when there is a big difference between the size of the send and receive windows? [5]

In the **Go-Back-N (GBN) ARQ protocol**, the sender is allowed to send up to $2^m - 1$ frames without waiting for acknowledgment. However, the receiver can only accept **one frame at a time** — its **receive window size is 1**. Despite this asymmetry, **flow control** is still effectively maintained.

How Flow Control Works in GBN:

Send Window (Sender Side)

The sender keeps a **window** of up to $2^m - 1$ frames that it can transmit before requiring ACKs. It maintains a timer for the earliest unacknowledged frame.

Receive Window = 1 (Receiver Side)

The receiver only accepts the **next expected frame**. Any out-of-order frames are **discarded**, enforcing strict sequence.

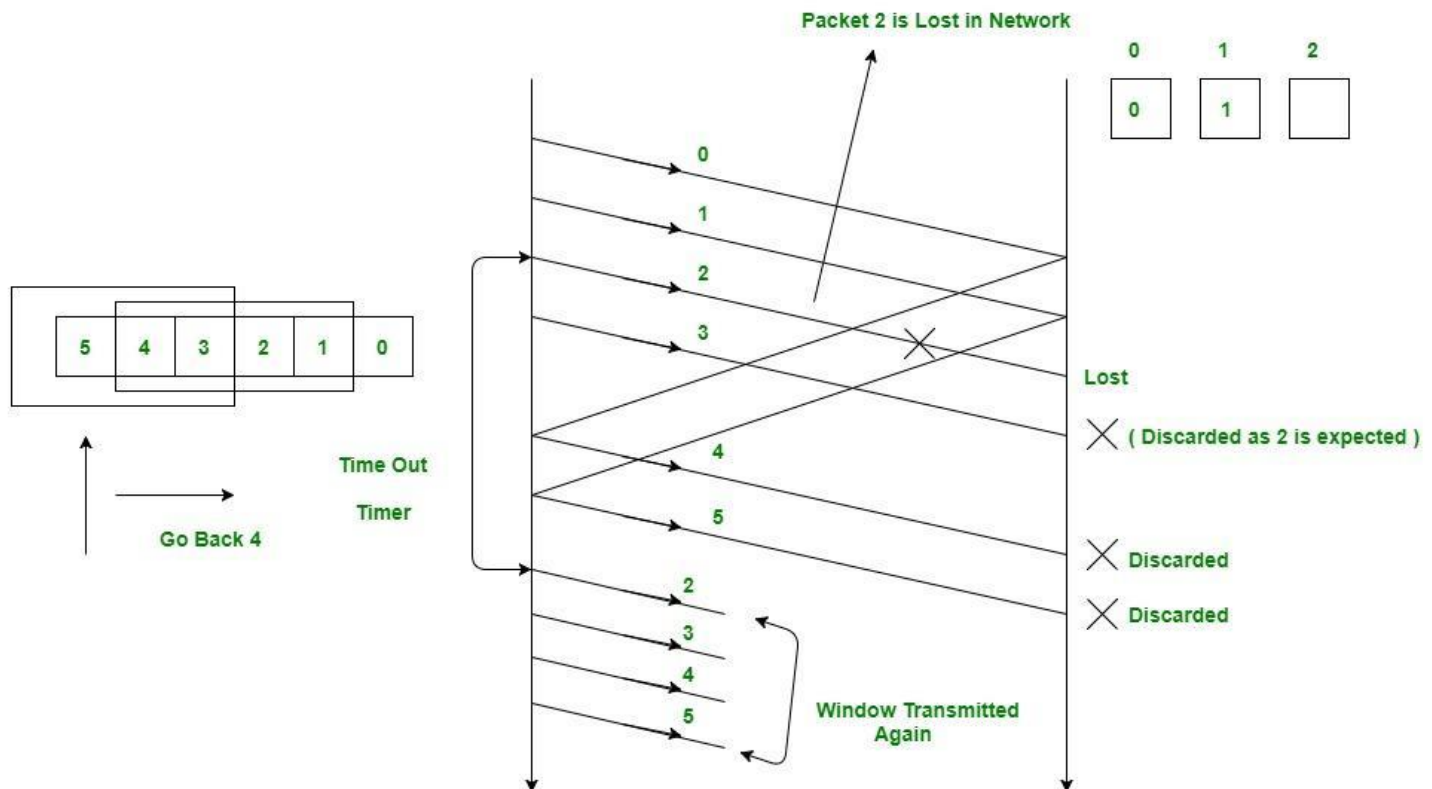
Use of Acknowledgments (ACKs)

The receiver sends an ACK for each correct frame. If a frame is missing, no ACK is sent, and the sender waits until timeout.

Timeout and Retransmission

If the sender doesn't receive ACK for a frame before the timer expires, it **goes back** and retransmits **that frame and all subsequent ones** — even if some of them were received and discarded.

Explanation with Diagram:



- Frames **0 and 1** are sent and acknowledged.
- **Frame 2** is **lost** in the network.

- Frames **3, 4, and 5** are sent but **discarded** by the receiver because it is still waiting for frame 2 (receive window = 1).
- After **timeout**, the sender **goes back to 2** and retransmits frames **2, 3, 4, and 5**.
- This ensures the receiver gets the correct frame and maintains proper flow.

Advantages of GBN Protocol

- Simple to implement and effective for reliable communication.
- Better performance than stop-and-wait protocols for error-free or low-error networks.

Disadvantages of GBN Protocol

- Inefficient if errors are frequent, as multiple frames might need to be retransmitted unnecessarily.
- Bandwidth can be wasted due to redundant retransmissions.

2. b) If the data link layer can detect errors between hops, why do you think we need another checking mechanism at the transport layer in OSI Model?[4]

Even though the **data link layer** provides **error detection and correction** at each **hop** (i.e., node-to-node), we still need **error checking at the transport layer** for the following important reasons:

1. End-to-End Reliability

The **data link layer** only checks data **between adjacent nodes**.

It **cannot guarantee** that data remains error-free **from the sender to the final destination** across all intermediate hops.

The **transport layer (like TCP)** ensures the **entire end-to-end communication** is reliable and error-free.

2. Undetected Errors at Lower Layers

Sometimes, **bit errors** may go **undetected** at the data link layer.

Transport layer adds an **extra layer of verification** to catch such errors using checksums.

3. End-to-End Data Integrity

The transport layer verifies that:

Entire messages arrive,

They arrive **in order**,

And they are **not corrupted**.

It handles **retransmission, sequencing, and error correction** that lower layers don't.

4. Application-Level Assurance

Applications like web browsers, email, or file transfers expect **complete and correct data**.

Transport layer provides **guaranteed delivery services** (e.g., via TCP), which are critical for many application protocols.

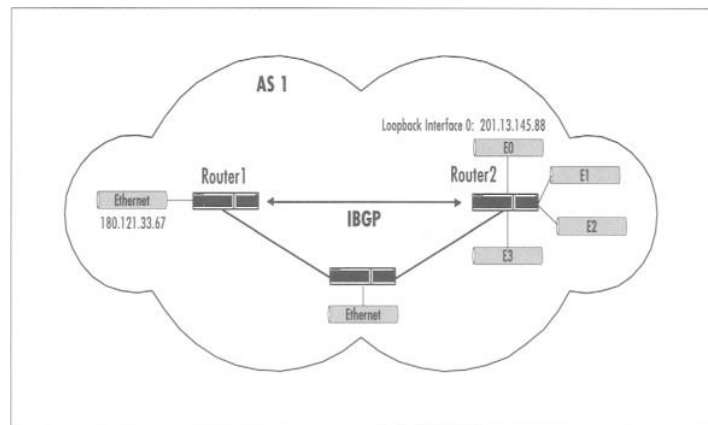
Example:

Suppose a message passes through 5 routers:

- Data link layer checks errors **at each hop**.
- But there's **no guarantee** the message will arrive **correctly at the destination**.
- **Transport layer ensures the entire path** is covered — from sender's app to receiver's app.

he **data link layer** provides **hop-by-hop** error detection, but the **transport layer** provides **end-to-end** reliability and error checking, which is essential for **complete, correct, and ordered delivery** of data in real-world communication systems.

c) What do you mean by loopback interface? An organization is assigned the block 2000:1456:2474/48. What is the IPv6 address of an interface in the third subnet if the IEEE physical address of the computer is (F5-A9-23-14-7A-D2)1616?



The **loopback interface** is a virtual network interface in a computer or device that is used to send network traffic to itself. It's not tied to any physical hardware but acts like a network interface internally.

Key points about loopback interface:

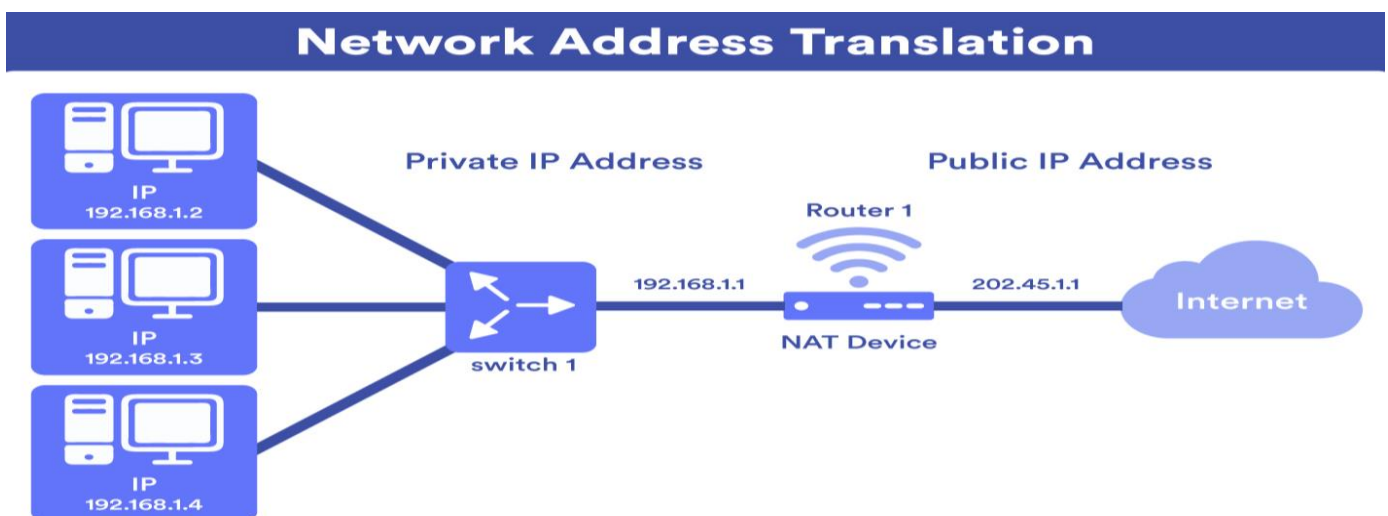
- It allows a device to communicate with itself using network protocols.
- Typically assigned the IP address **127.0.0.1** in IPv4 (called the "localhost" address).
- Used for testing and troubleshooting network software without needing a physical network connection.
- Helps developers test network applications on the local machine.
- Packets sent to the loopback interface never leave the device; they loop back internally.

The loopback interface is a special network interface that lets a device talk to itself over the network protocol stack, mainly for testing and local communication purposes.

3

(C) How can NAT help in address depletion? Explain with necessary diagram.[3]

Network Address Translation (NAT) is a process in which one or more local IP addresses are translated into one or more Global IP addresses and vice versa to provide Internet access to the local hosts. It also does the translation of port numbers, i.e., masks the port number of the host with another port number in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table.



AVAILABLE AT:

IP Address Depletion

IPv4 has a limited number of addresses (~4.3 billion). With billions of devices, public IPv4 addresses are running out. This shortage is called IP address depletion.

How NAT Helps in Address Depletion:

- Instead of assigning a unique public IP to every device, NAT allows many devices to use one public IP.
- Internally, devices use private IPs, which do not consume public address space.
- Only the router's public IP is visible on the internet.
- This conserves public IP addresses significantly.

PCs use private IPs internally. When they access the internet, the NAT router replaces their private IP with the **single public IP** (203.0.113.5). NAT keeps track of each connection so replies go to the correct internal device.

4 Cryptography pdf

5©

list three forwarding techniques and give a brief description of each.

Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a [router](#) to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination. However, this simple solution is impossible today in an internetwork such as the Internet because the number of entries needed in the routing table would make table lookups inefficient.

Forwarding Techniques

Several techniques can make the size of the routing table manageable and also handle issues such as security

a. Next-Hop Method

Definition: The router forwards the packet to a specific **next-hop IP address**, rather than making a complete routing decision.

In this technique, the routing table holds only the address of the next hop instead of information about the complete route (route method). The entries of a routing table must be consistent with one another.

Use case: Efficient in large networks because it reduces the size of the routing table.

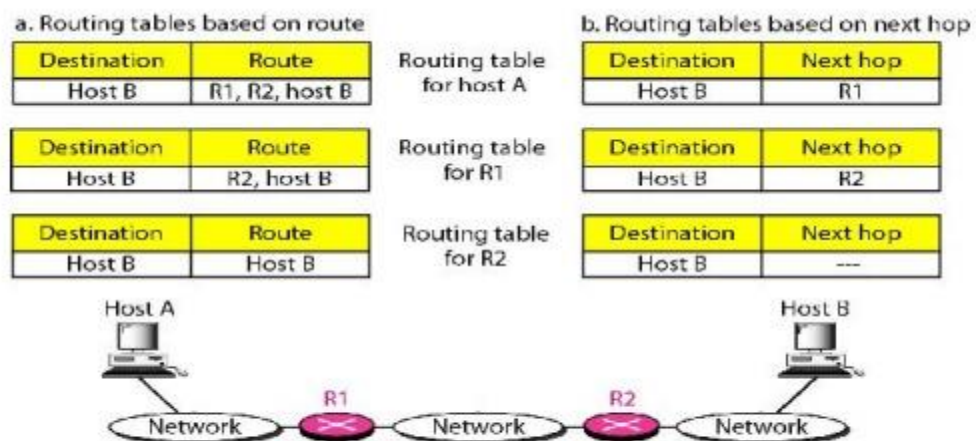


Figure 3.40 Route method versus next-hop method

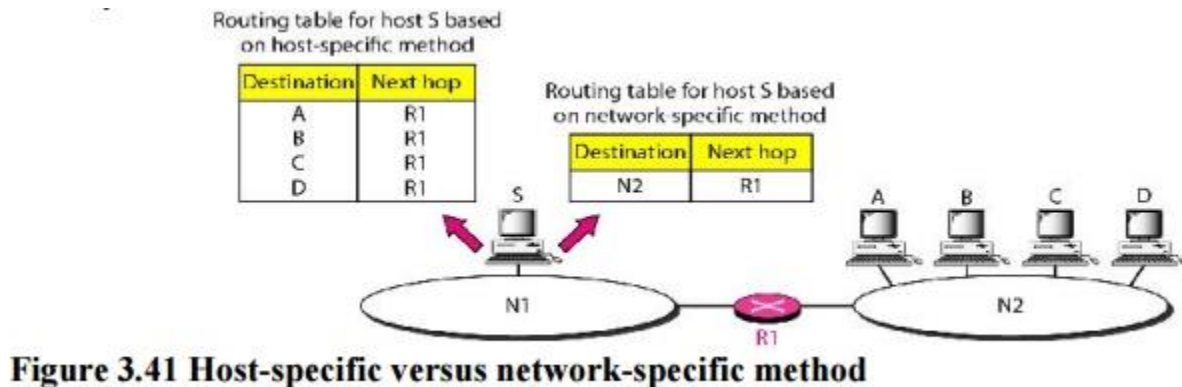
b. Network-Specific Method versus Host-Specific Method

AVAILABLE AT:

Definition: The router forwards packets based on the **entire destination network address**, not individual host addresses.

Here, instead of having an entry for every destination host connected to the same physical network (host-specific method), we have only one entry that defines the address of the destination network itself. Host-specific routing is used for purposes such as checking the route or providing security measures.

Use case: Reduces the number of routing entries and simplifies routing.

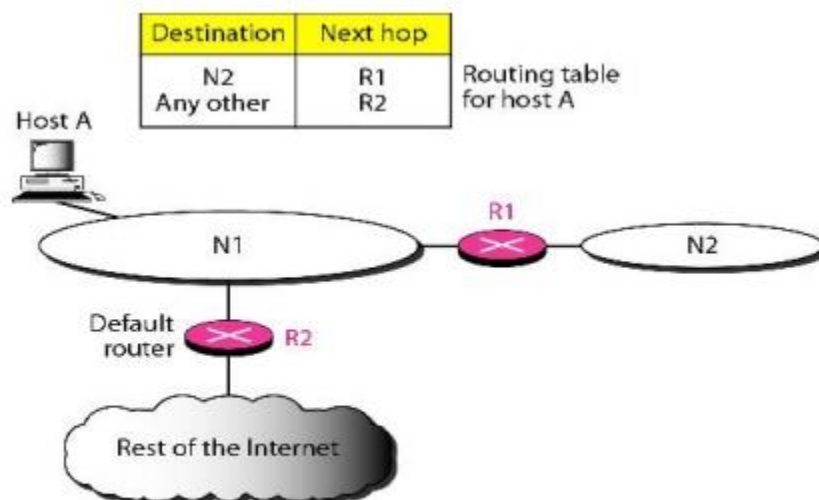


c. Default Method

Definition: A route used when **no other specific route** matches the destination IP.

Host A is connected to a network with two routers. Router R1 routes the packets to hosts connected to network N2. However, for the rest of the Internet, router R2 is used. So instead of listing all networks in the entire Internet, host A can just have one entry called the default (normally defined as network address 0.0.0.0). Acts as a “catch-all” route. If the router doesn't find a match in the routing table, it uses the default.

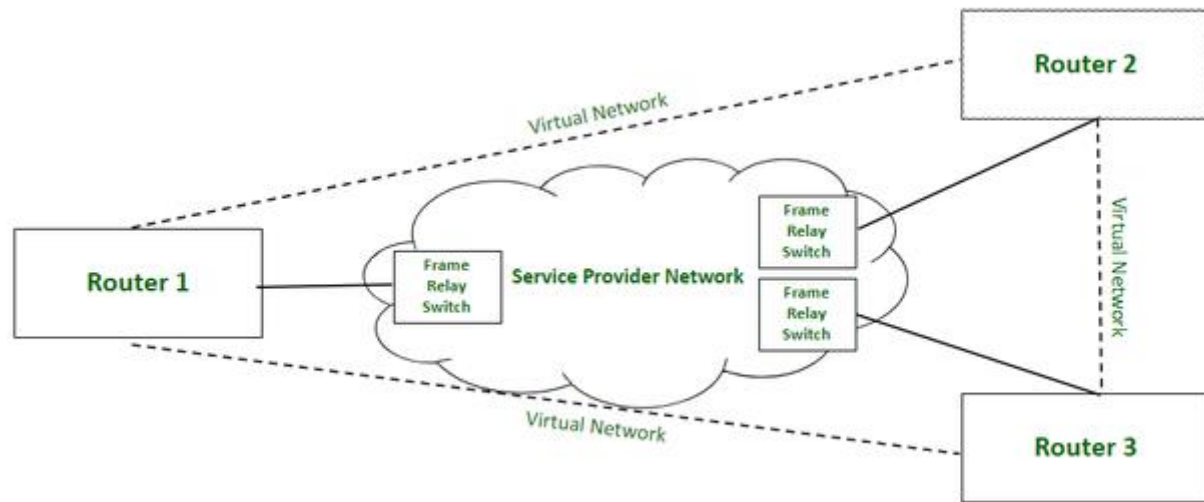
Use case: Useful for small networks or edge routers connecting to the internet.



6(a)

Frame Relay is a **high-speed, packet-switched Wide Area Network (WAN) protocol** designed to connect **Local Area Networks (LANs)** over long distances.

- It works at the **data link layer (Layer 2)** of the OSI model.
- Transmits data in units called **frames**.
- It's used to transmit data over **virtual circuits** between LANs.
- Originally designed for use over **ISDN and leased lines**.



Advantages:

- High speed
- Scalable
- Reduced network congestion
- Cost-efficient
- Secured connection

Disadvantages:

- Lacks error control mechanism
- Delay in packet transfer
- Less reliable

Reasons Frame Relay is a Better Solution than T-1 for Connecting LANs:

Feature	Frame Relay	T-1 Line
Cost	Cheaper	More expensive
Bandwidth Efficiency	Shares bandwidth dynamically	Fixed 1.544 Mbps
Scalability	Can support many virtual circuits	One-to-one connection
Flexibility	Many LANs can be connected over one line	One LAN to another only
Overhead	Lower (no error correction)	Higher due to constant connection
Use of Resources	Efficient (transmits only when needed)	Always consumes full bandwidth

Explanation:

- A **T-1 line** is a **dedicated, point-to-point connection** with a constant data rate (1.544 Mbps). It is expensive and provides **no flexibility**—used for always-on, fixed communication between two locations.
- **Frame Relay**, on the other hand:
 - Uses **virtual circuits**, allowing **many connections** over one physical line.
 - Sends data **only when needed**, reducing cost.
 - Supports **multiple LAN connections**, making it more efficient and scalable for growing networks.

6[c]

Subnetting :

Subnetting is the process of dividing a **larger network** into **smaller subnetworks** (subnets).

- It helps organize a network, improve security, and efficiently use IP addresses.
- Done by **borrowing bits from the host part** of the IP address.

□ Example:

Class B default mask: 255.255.0.0

Subnet mask: 255.255.255.0 (borrowed 8 bits for subnetting)

Supernetting :

Supernetting is the process of combining **multiple smaller networks** (usually Class C) into a **larger network block**.

- It reduces the number of routes in routing tables (used in CIDR).
- Done by **borrowing bits from the network part** and treating them as host bits.

□ Example:

Combining four Class C networks (/24) into one supernet: 192.168.0.0/22

Subnet Mask vs. Supernet Mask vs. Default Mask

Feature	Default Mask	Subnet Mask	Supernet Mask
Purpose	Identifies network and host by class	Divides a large network into subnets	Combines multiple networks into one
Bits Borrowed	None	From host bits	From network bits
# of Hosts/Subnets	Fixed (based on class)	More subnets, fewer hosts per subnet	Fewer subnets, more hosts per subnet
Example (Class C)	255.255.255.0 (/24)	255.255.255.224 (/27)	255.255.254.0 (/23)
Used in	Classful networks	Private networks, enterprise networks	CIDR, ISP routing, internet backbone

Class C Default Mask (/24):

255.255.255.0 → 11111111.11111111.11111111.00000000

Subnet Mask (/27):

255.255.255.224 → 11111111.11111111.11111111.11100000

↑ Borrowed 3 bits from host portion

Supernet Mask (/23):

255.255.254.0 → 11111111.11111111.11111110.00000000

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

↓ Gave back 1 bit to host portion (from network part)

Subnetting = Breaking down a network into smaller segments using more bits for network.

Supernetting = Aggregating smaller networks into one using fewer bits for network.

- Their **masks differ** from the **default classful mask** in terms of **number of leading 1s** in binary.
 - **Subnet mask** → more 1s than default
 - **Supernet mask** → fewer 1s than default

6[d]

Frame Relay uses a **simple and efficient** set of techniques to handle **congestion control** in wide area networks (WANs). It does **not** perform error correction like older protocols (e.g., X.25), but instead provides mechanisms to **detect** and **signal** congestion, allowing **end devices** to respond appropriately.

Frame Relay Congestion Control Techniques:

1. Forward Explicit Congestion Notification (FECN)

- Set by a **Frame Relay switch** when there is congestion in the **forward direction**.
 - It **informs the destination** device that congestion occurred along the path.
 - The destination can then notify the sender to slow down transmission.
- ☐ **Bit Set in Frame** → Receiver knows congestion happened in the forward path.

2. Backward Explicit Congestion Notification (BECN)

- Set by the network when congestion is detected, and sent in the **reverse direction** (back toward the sender).
 - It **alerts the sender** to reduce the data transmission rate.
- ☐ **Bit Set in Frame** → Sender is told to slow down due to congestion on the path.

3. Discard Eligibility (DE) Bit

- Used to mark **less important (low-priority)** frames.
 - When congestion happens, switches are allowed to **drop frames with DE = 1** to preserve high-priority traffic.
- ☐ Helps in **prioritizing** important data under congestion.

Frame Relay uses several **key attributes** to manage and control network traffic efficiently. These attributes help ensure **reliable performance**, manage **congestion**, and allocate **bandwidth** among multiple users.

1. Committed Information Rate (CIR)

- The **average guaranteed data rate** (in bits per second) the network commits to support for a particular **virtual circuit (PVC)**.
 - Frame Relay ensures that this rate is available under normal network conditions.
- ☐ Example: CIR = 64 Kbps means the network guarantees at least 64 Kbps.

2. Committed Burst Size (Bc)

- The **maximum number of bits** the network agrees to transfer during a fixed time interval (**Tc**) under normal conditions.
- It is calculated as:
$$Bc = CIR \times Tc$$

3. Excess Burst Size (Be)

- The number of **additional bits** (above Bc) that can be sent during the interval **Tc**, but **not guaranteed** by the network.
- If the network is congested, frames within Be **may be dropped**.

4. Measurement Interval (Tc)

- The **time window** (in seconds) over which Bc and Be are measured and enforced.

- Typically a small interval (e.g., 10 to 125 milliseconds).

5. Discard Eligibility (DE) Bit

- A **bit in the frame header** used to indicate that a frame is **low priority**.
- Frames marked with DE = 1 are eligible to be **discarded during congestion** to protect higher-priority traffic.

6. Forward Explicit Congestion Notification (FECN)

- A **bit** that signals the **destination** device that congestion was experienced in the forward path.
- The receiver may inform the sender to **reduce transmission speed**.

7. Backward Explicit Congestion Notification (BECN)

- A **bit** that travels in the **opposite direction** (backward path) to alert the **sender** that there is congestion in the network.
- The sender can **slow down** the sending rate accordingly.

7[b]

A **digital signature** is a **mathematical technique** used to ensure the **authenticity, integrity, and non-repudiation** of a message or document in digital communication.

It acts like a handwritten signature but is **much more secure** and **tamper-proof**.

It confirms:



- **Who** sent the message (authentication),
- That the message **was not altered** (integrity),
- And the sender **cannot deny** sending it (non-repudiation).

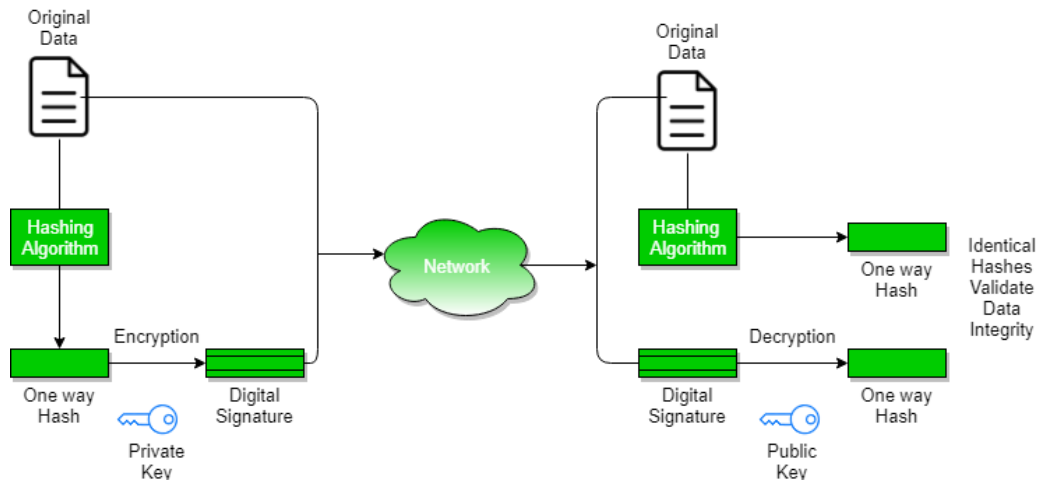
How Digital Signature Works (to Ensure Message Integrity):

Steps to Implement a Digital Signature:

1. **Hashing the Message:**
 - A **hash function** (e.g., SHA-256) is applied to the message.
 - Output: A fixed-length **message digest** (hash).
 - This digest uniquely represents the content.
2. **Encrypting the Hash (Signing):**
 - The sender encrypts the **hash** with their **private key**.
 - The result is the **digital signature**.
 - This signature is then attached to the message.
3. **Sending the Message + Signature:**
 - The original message and the digital signature are sent to the receiver.

At the Receiver's Side:

1. **Hash the Received Message:**
 - Receiver applies the same hash function to the received message.
 - Gets a new message digest.
2. **Decrypt the Digital Signature:**
 - Receiver decrypts the digital signature using the sender's **public key**.
 - This gives the original message digest (from the sender).
3. **Compare Both Digests:**
 - If both digests match →  Message is **authentic and unmodified**.
 - If not →  Message was **altered** or **tampered with**.



Why It Ensures Message Integrity:

- Any tiny change in the message causes a **different hash**.
- So if the message is altered, the comparison will **fail**.
- Only the **sender's private key** can create the signature, and only the **public key** can verify it.

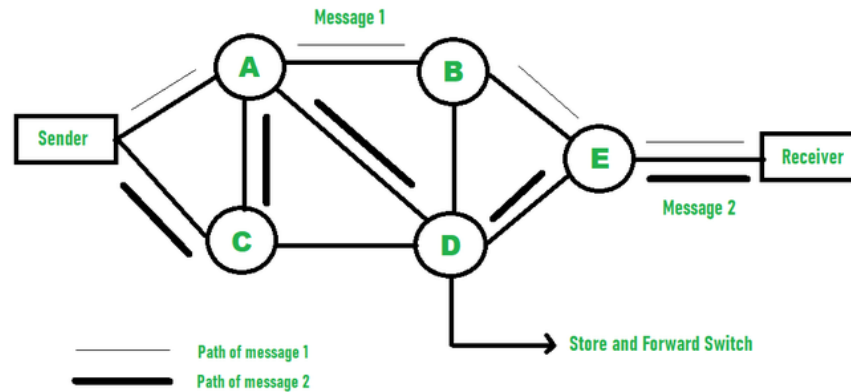
Real-world Use Cases:

- Secure emails (e.g., PGP, S/MIME)
- Software distribution and updates
- Digital certificates (SSL/TLS)
- E-commerce and e-contracts

8[c]

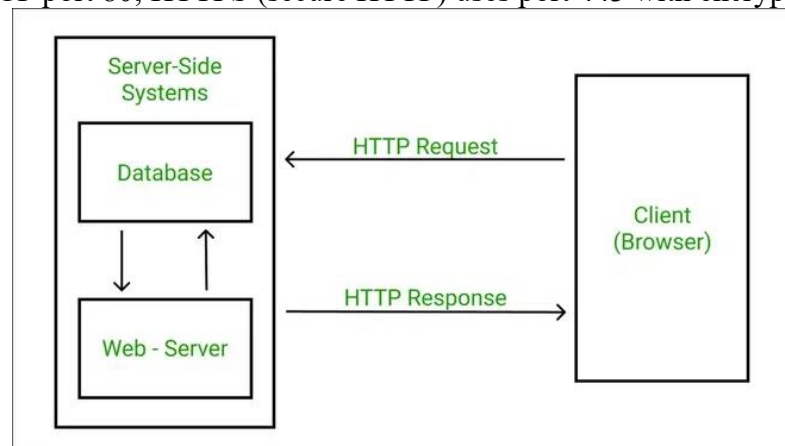
1. Packet Switching

- **Concept:** Data is divided into small packets, each containing part of the message plus control information like source, destination, and sequence number.
- **Routing:** Each packet can take a different path through the network, based on current traffic conditions and routing algorithms.
- **Advantages:**
 - Efficient utilization of network resources since bandwidth is shared.
 - Robust and fault-tolerant; if a path fails, packets are rerouted.
 - Supports bursty data traffic typical in computer communications.
- **Disadvantages:**
 - Variable delay because packets may take different routes.
 - Possible packet loss or out-of-order delivery, requiring error handling and reassembly.
- **Example:** The Internet uses packet switching with protocols like IP.



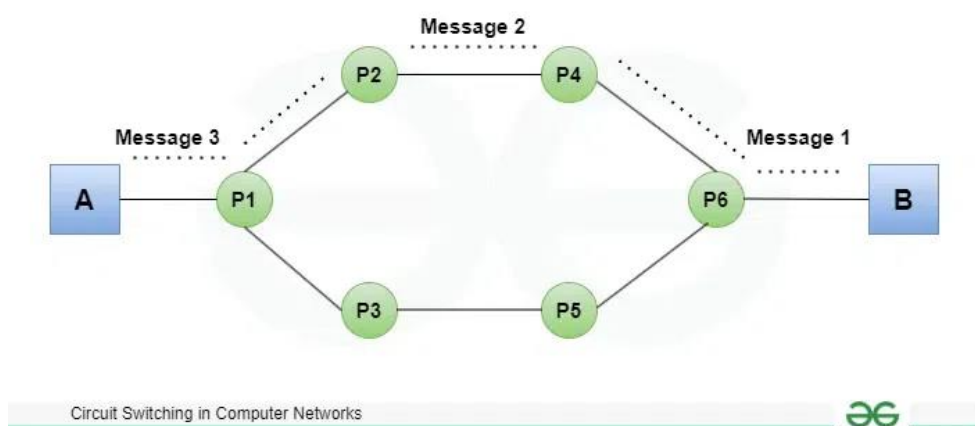
2. HTTP (HyperText Transfer Protocol)

- **Purpose:** Protocol for transferring hypertext documents (web pages) between clients (web browsers) and servers.
- **How it Works:**
 - Client sends an HTTP request (e.g., GET or POST).
 - Server processes the request and sends back a response (e.g., HTML content, status codes).
- **HTTP Versions:**
 - HTTP/1.1 supports persistent connections.
 - HTTP/2 improves performance with multiplexing and header compression.
- **Stateless Nature:** Each HTTP request is independent, so servers don't retain client state between requests unless sessions or cookies are used.
- **Port:** Default uses TCP port 80; HTTPS (secure HTTP) uses port 443 with encryption via TLS.



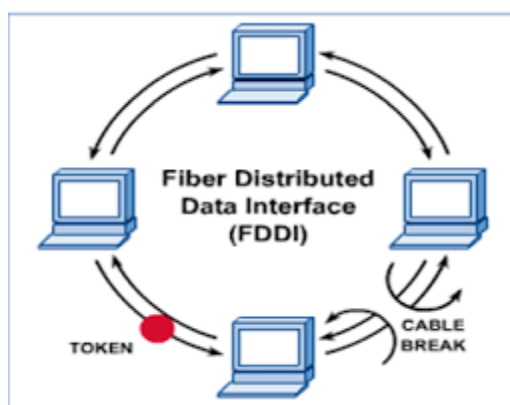
3. Circuit Switching

- **Operation:**
 - A dedicated communication path (circuit) is established before data transfer.
 - The circuit remains reserved for the entire session.
- **Characteristics:**
 - Guarantees a fixed bandwidth and constant transmission delay.
 - Connection-oriented: setup, data transfer, and teardown phases.
- **Drawbacks:**
 - Inefficient for bursty or intermittent data, as the reserved path is idle when no data is sent.
 - Not scalable for many users.
- **Example:** Traditional telephone networks.



4. FDDI (Fiber Distributed Data Interface)

- **Technology:** A LAN technology standardized by ANSI that uses fiber optic cable.
- **Speed and Distance:**
 - Operates at 100 Mbps.
 - Can cover up to 200 kilometers using dual rings.
- **Topology:**
 - Dual ring (primary and secondary) for redundancy and fault tolerance.
 - If one ring fails, the other ring maintains network operation.
- **Access Method:**
 - Token-passing protocol for managing access to the network medium.
- **Applications:**
 - Used as a backbone network connecting different LAN segments.
 - Suitable for environments needing high-speed, long-distance, and reliable data transmission.



Feature	Packet Switching	Circuit Switching
Data Handling	Data is divided into packets and sent independently.	Data follows a dedicated path in a continuous stream.

Feature	Packet Switching	Circuit Switching
Connection Setup	No call setup is required.	Requires call setup before transmission.
Path	No fixed path; packets may take different routes.	A fixed physical path is established for the entire session.
Intermediate Node Processing	Data is processed at all intermediate nodes.	Data is processed only at the source.
Delay Between Data Units	Delay is not uniform; may vary.	Delay is uniform and consistent.
Reliability	Less reliable; packets may be lost or arrive out of order.	More reliable; dedicated path ensures ordered delivery.
Data Transmission Responsibility	Transmission is handled by both source and intermediate routers.	Transmission is handled mainly by the source.
Resource Utilization	Efficient; less resource wastage.	Inefficient; resources are reserved even when not in use.
Protocol Complexity	Requires complex protocols for reordering and error checking (e.g., TCP/IP).	Requires simpler protocols due to dedicated path.
Latency	Higher latency due to dynamic routing and reassembly.	Lower latency due to dedicated and continuous path.
Overhead	More overhead due to routing and addressing in each packet.	Less overhead since the path and addressing are fixed.

8[b]

Why do you mean by 'two-node-loop-instability' with distance vector? explain with necessary diagram. Also provide a solution for this problem

In **distance vector routing protocols**, two routers (nodes) connected to each other can sometimes get stuck in a **routing loop**, where they continuously update each other with incorrect route information, causing instability. This problem is often called "**count-to-infinity**" or **routing loop**, but specifically in the case of **two nodes**, it's known as **two-node loop instability**.

Explanation:

- Suppose two routers, **Router A** and **Router B**, are connected.
- Each router has a route to a destination network through the other.

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career
The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

- If the destination network becomes unreachable, both routers might mistakenly keep updating their distance vectors indicating a longer and longer path to that network.
- Because each router depends on the other's routing table, they keep increasing the distance metric indefinitely (hence the term **count-to-infinity**).
- This causes **routing instability** and delays convergence.

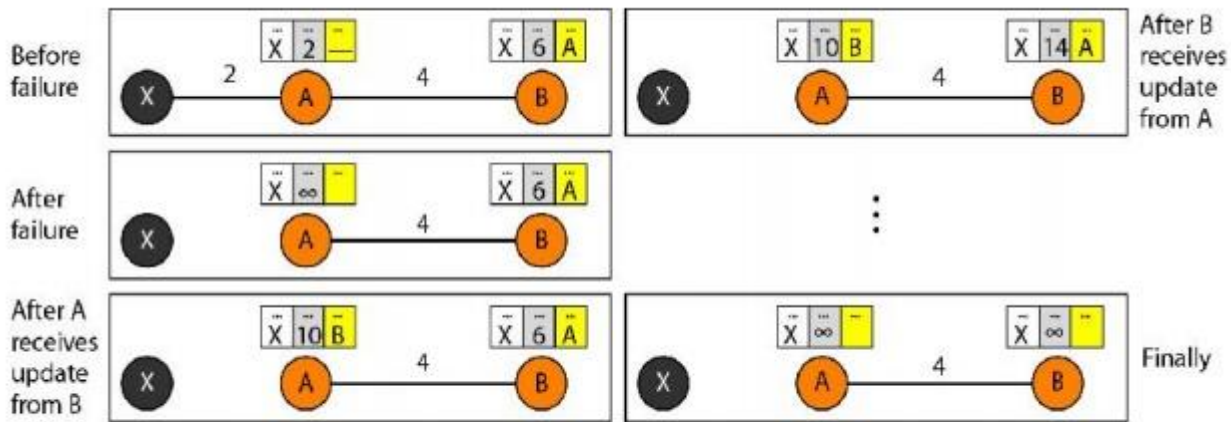


Figure 3.48 Two-node instability

Based on the (Figure 3.48), here's a step-by-step explanation of how two-node-loop instability occurs in what appears to be a routing protocol scenario:

Initial State (Before Failure)

- Node A has a path to destination X with:
 - Next hop: B
 - Cost: 10
- Node B has a path to destination X with:
 - Next hop: A
 - Cost: 4

This creates an immediate routing loop between A and B for destination X.

Step-by-Step Instability Process:

- 1. After Failure:**
 - The direct path to X fails for both nodes
 - Both nodes now only know about X through each other
- 2. Initial Routing Tables:**
 - Node A thinks: "I can reach X through B with cost 10"
 - Node B thinks: "I can reach X through A with cost 4"
- 3. After A receives update from B:**
 - Node A receives B's advertisement that it can reach X with cost 4
 - Node A updates its path: cost = B's cost (4) + link cost (6) = 10
 - This doesn't change A's existing path (still cost 10 through B)
- 4. After B receives update from A:**
 - Node B receives A's advertisement that it can reach X with cost 10
 - Node B updates its path: cost = A's cost (10) + link cost (6) = 16
 - Now B's path to X is through A with cost 16
- 5. Next Exchange:**
 - Node A receives B's new advertisement (cost 16)
 - Node A updates: cost = 16 (from B) + 6 = 22
 - Node B receives A's advertisement (cost 22)
 - Node B updates: cost = 22 + 6 = 28

- This cycle continues indefinitely with costs increasing without bound

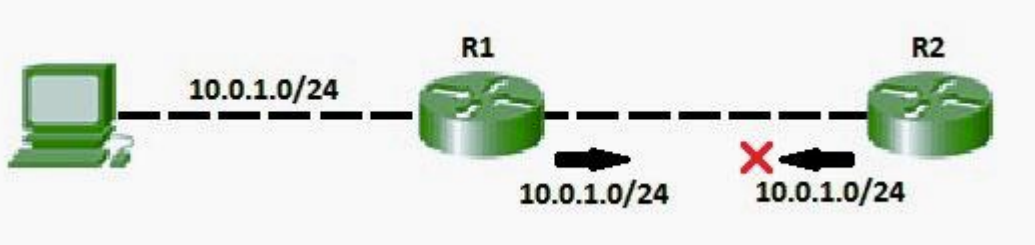
Why This Happens:

- **Routing Loop:** Each node is depending on the other to reach X
- **Counting to Infinity:** The costs keep increasing in each exchange
- **No Loop Prevention:** There's no mechanism to break this cycle

Solution to Two-Node Loop Instability:

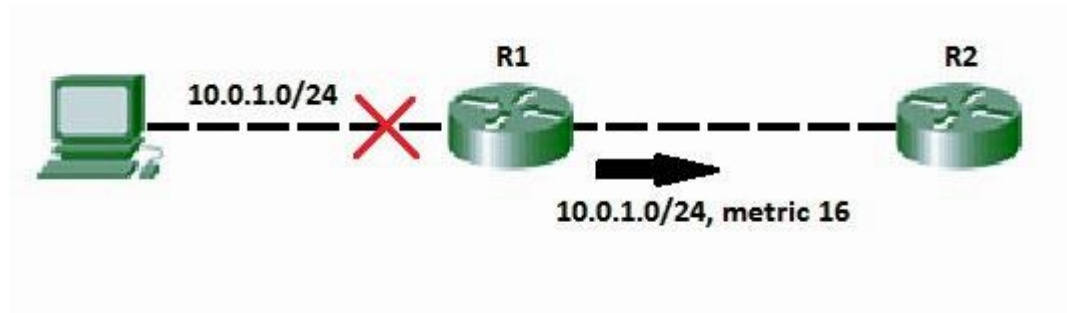
1. Split Horizon:

- Routers do not advertise a route back to the neighbor from which they learned it.
- This prevents Router A from telling Router B that it can reach a destination via Router B.



2. Poisoned Reverse:

- A variation of split horizon where routers explicitly advertise an infinite metric (unreachable) back to the neighbor for routes learned from that neighbor.
- This quickly informs the neighbor that the route is invalid.



2(b)

A **socket address** is a unique identifier used in networking to represent an endpoint for communication. It combines two key pieces of information:

- **IP Address:** Identifies the device on the network.
- **Port Number:** Identifies the specific process or service on that device.

Format:

Socket Address = IP Address + Port Number

Example: 192.168.1.10:80

This tells the network to deliver data to port 80 (usually HTTP) on the device with IP 192.168.1.10.

Comparison between TCP and UDP Protocols

Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection Type	Connection-oriented	Connectionless
Reliability	Reliable – ensures delivery, order, and no duplication	Unreliable – no guarantee of delivery or order
Error Checking	Yes (with acknowledgment and retransmission)	Yes (with checksum, but no retransmission)

Data Sequencing	Maintains order of packets	No sequencing – packets may arrive out of order
Speed	Slower (due to overhead)	Faster (less overhead)
Overhead	High (due to acknowledgments, handshakes)	Low
Use Cases	Web browsing, email, file transfer (FTP), etc.	Streaming, gaming, DNS, VoIP
Header Size	20–60 bytes	8 bytes
Congestion Control	Yes	No
Handshake	3-way handshake before data transmission	No handshake – data sent directly

Difference between Circuit Switching and Packet Switching

Circuit Switching	Packet Switching
<p>Circuit switching has 3 phases:</p> <ul style="list-style-type: none"> i) Connection Establishment. ii) Data Transfer. iii) Connection Released. 	<p>In Packet switching directly data transfer takes place.</p>
<p>In circuit switching, each data unit knows the entire path address which is provided by the source.</p>	<p>In Packet switching, each data unit just knows the final destination address intermediate path is decided by the routers.</p>
<p>In Circuit switching, data is processed at the source system only</p>	<p>In Packet switching, data is processed at all intermediate nodes including the source system.</p>
<p>The delay between data units in circuit switching is uniform.</p>	<p>The delay between data units in packet switching is not uniform.</p>
<p>Resource reservation is the feature of circuit switching because the path is fixed for data transmission.</p>	<p>There is no resource reservation because bandwidth is shared among users.</p>
<p>Circuit switching is more reliable.</p>	<p>Packet switching is less reliable.</p>
<p>Wastage of resources is more in Circuit Switching</p>	<p>Less wastage of resources as compared to Circuit Switching</p>
<p>It is not a store and forward technique.</p>	<p>It is a store and forward technique.</p>
<p>Transmission of the data is done by the source.</p>	<p>Transmission of the data is done not only by the source but also by the intermediate routers.</p>

Circuit Switching	Packet Switching
Congestion can occur during the connection establishment phase because there might be a case where a request is being made for a channel but the channel is already occupied.	Congestion can occur during the data transfer phase, a large number of packets comes in no time.
Circuit switching is not convenient for handling bilateral traffic.	Packet switching is suitable for handling bilateral traffic.
In Circuit switching, the charge depends on time and distance and not on traffic in the network.	In Packet switching, the charge is based on the number of bytes and connection time.
Recording of packets is never possible in circuit switching.	Recording of packets is possible in packet switching.
In Circuit Switching there is a physical path between the source and the destination	In Packet Switching there is no physical path between the source and the destination
Circuit Switching does not support store and forward transmission	Packet Switching supports store and forward transmission
Call setup is required in circuit switching.	No call setup is required in packet switching.
In circuit switching each packet follows the same route.	In packet switching packets can follow any route.
The circuit switching network is implemented at the physical layer.	Packet switching is implemented at the datalink layer and network layer
Circuit switching requires simple protocols for delivery.	Packet switching requires complex protocols for delivery.

Department of Computer Science and Engineering, University of Barishal Course Code: CSE-3105; Course Title: Computer Networks Admission Session: 2019-20; Exam: Mid Term-II; Time: 01 Hour; Marks: 10	
1.	An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows: i) The first group has 64 customers; each need 256 addresses. ii) The second group has 128 customers; each need 128 addresses. iii) The third group has 128 customers; each need 64 addresses. Design the subblocks and find out how many addresses are still available after these allocations.
2.	Find errors, if any, in the following IPv4 address. i) 111.56.045.78 ii) 110101.23.14.69 iii) 75.35.325.12 iv) EF6.23.00011.6
3.	In an IPv4 datagram, the value of total-length field is (00A0) ₁₆ and the value of the header-length (HLEN) is (5) ₁₆ . How many bytes of payload are being carried by the datagram? What is the efficiency (ratio of the payload length to the total length) of this datagram?
4.	What is NAT? How can NAT help in address depletion? Explain with necessary diagram.
5.	Draw the IPv4 datagram format. What do you know about circuit switching and packet switching.
6.	Switch is a 'self-learning' device. What does it learn when time progress? How does it learn?

1.Subnetting Problem

Given:

- Starting block: 190.100.0.0/16 (Total = $2^{16} = 65,536$ addresses)

i) **64 customers × 256 addresses = $64 \times 2^8 = 16,384$**

Each customer gets a /24 block

→ Total used = $64 \times 256 = 16,384$

ii) **128 customers × 128 addresses = $128 \times 2^7 = 16,384$**

Each customer gets a /25 block

→ Total used = $128 \times 128 = 16,384$

iii) **128 customers × 64 addresses = $128 \times 2^6 = 8,192$**

Each customer gets a /26 block

→ Total used = $128 \times 64 = 8,192$

Total allocated = $16,384 + 16,384 + 8,192 = 40,960$

Remaining = $65,536 - 40,960 = 24,576$ addresses

2. Find Errors in IPv4 Addresses

i) 111.56.045.78 → **✗** Error: 045 is invalid (decimal must be 0–255 and no leading zeros)

ii) 110101.23.14.69 → **✗** Error: 110101 is invalid (max is 255)

iii) 75.35.325.12 → **✗** Error: 325 is out of range

iv) EF6.23.00011.6 → **✗** Error: EF6 and 00011 are not valid decimal octets

3. IPv4 Datagram Payload and Efficiency

Given:

Total Length = $00A0_{16} = 160$ bytes

HLEN = 5 (means $5 \times 4 = 20$ bytes of header)

Step-by-step:

Header size = 20 bytes

Payload = Total - Header = $160 - 20 = 140$ bytes

Efficiency = Payload / Total = $140 / 160 = 0.875 = 87.5\%$

Answer:

Payload size = 140 bytes

Efficiency = 87.5%

5.

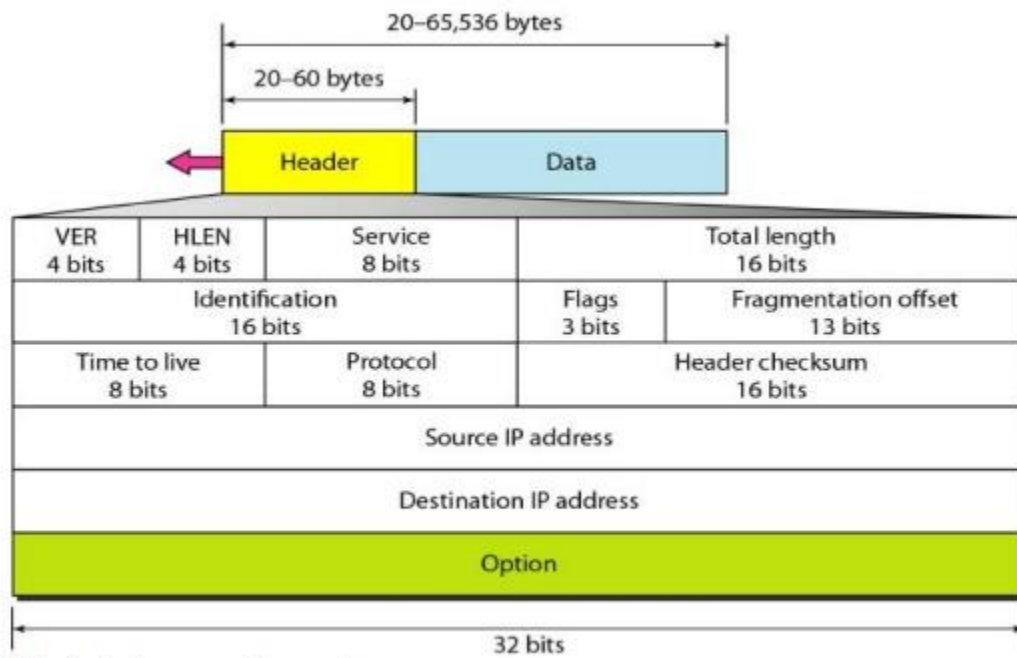


Figure 3.19 IPv4 datagram format

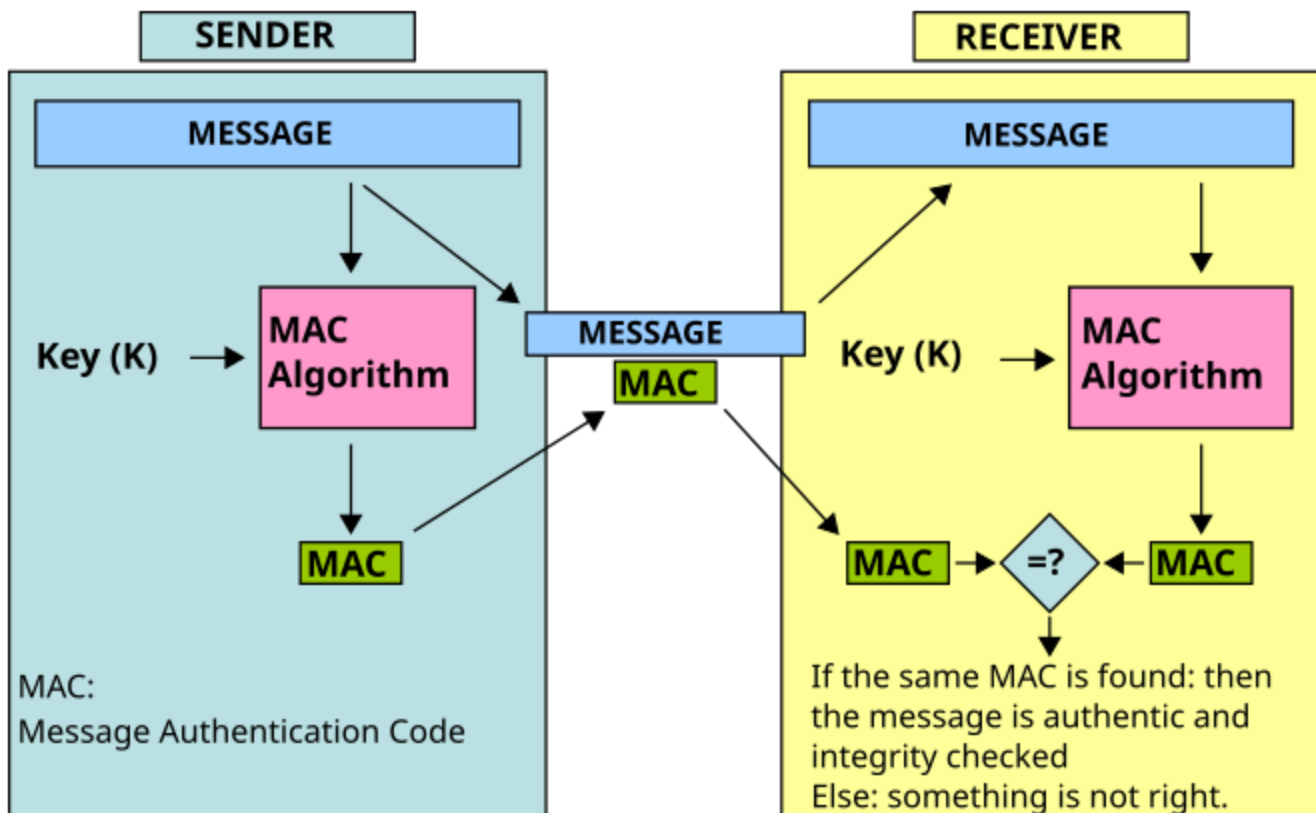
The IPv4 datagram is the basic unit of data transmission in the Internet Protocol version 4. It consists of two parts: the **header** and the **data (payload)**. The header carries information necessary for routing and delivery, while the data is the actual content being transmitted. The size of an IPv4 datagram can vary from **20 bytes to 65,535 bytes**, depending on the header options and the size of the data.

Header Fields (Bit-wise Description and Function)

1. **Version (4 bits)**
 - This field specifies the IP version being used.
 - For IPv4, the value is always 4 (binary 0100).
2. **Header Length (HLEN) (4 bits)**
 - Indicates the length of the header in 32-bit words.
 - Minimum value is 5 (which means $5 \times 4 = 20$ bytes).
 - Maximum value is 15 (i.e., 60 bytes).
3. **Type of Service (ToS) / DSCP (8 bits)**
 - Specifies the priority and quality of service desired.
 - Includes fields like delay, throughput, and reliability.
4. **Total Length (16 bits)**
 - Represents the total size of the datagram (header + data).
 - Maximum possible value is 65,535 bytes.
5. **Identification (16 bits)**
 - Used for uniquely identifying each datagram.
 - Important when a large datagram is fragmented into smaller parts.
6. **Flags (3 bits)**
 - Control fragmentation.
 - First bit: Reserved
 - Second bit: DF (Don't Fragment)
 - Third bit: MF (More Fragments)
7. **Fragment Offset (13 bits)**
 - Shows the position of a fragment in the original datagram.
 - Used during reassembly.
8. **Time To Live (TTL) (8 bits)**

- Limits the number of hops a datagram can take before being discarded.
 - Prevents infinite looping.
 - Decreased by one at each router.
9. **Protocol (8 bits)**
- Indicates the higher-level protocol using this datagram.
 - For example, TCP = 6, UDP = 17, ICMP = 1.
10. **Header Checksum (16 bits)**
- Used to detect errors in the header only.
 - Calculated at the sender side and verified by the receiver.
11. **Source IP Address (32 bits)**
- The IP address of the sender (e.g., 192.168.1.1).
12. **Destination IP Address (32 bits)**
- The IP address of the intended recipient.
13. **Options (0–40 bytes, variable length)**
- Optional field used for debugging, security, or timestamping.
 - Rarely used in modern networks.
14. **Padding (variable)**
- Ensures that the header ends on a 32-bit boundary.
 - Only added if Options field is used.

How does Message Authentication Code (MAC) work? Does it provide confidentiality? Justify your answer.



A Message Authentication Code (MAC) is a cryptographic technique used to ensure the **integrity** and **authenticity** of a message. It allows the receiver to verify that the message has not been altered during transmission and that it was sent by an authenticated sender.

Working Principle:

1. At the sender's side, a MAC is generated by applying a cryptographic function on the message using a **shared secret key**. This can be represented as:

AVAILABLE AT:

$$\text{MAC} = F(K, M)$$

where K is the key, M is the message, and F is the MAC algorithm (such as HMAC or CMAC).

2. The sender transmits both the original message and the generated MAC to the receiver.
3. At the receiver's end, the same MAC function is applied to the received message using the same secret key to generate a new MAC value.
4. The receiver compares the newly generated MAC with the MAC received.
 - If both values match, the message is considered **authentic** and **unchanged**.
 - If they differ, the message is either **tampered with** or **not from the expected sender**.

Confidentiality:

MAC does **not** provide confidentiality. It only ensures **integrity** and **authenticity**. The content of the message remains in plain text and can be read by any third party who intercepts it, even though they cannot modify it without detection. Therefore, while MAC protects against unauthorized modification and forgery, it does **not** protect the message content from being disclosed.

Justification:

Confidentiality means keeping the contents of the message secret from unauthorized parties. Since MAC does not encrypt the message, it does not prevent others from reading it. To achieve confidentiality, encryption techniques such as symmetric or asymmetric encryption must be used along with MAC.

1(a). Determine one or more layers of OSI model to perform the following tasks:

i) Format and code conversion services

Layer Involved: Presentation Layer (Layer 6)

This layer is responsible for ensuring that the data sent by the application layer of one system is readable by the application layer of another system. It performs translation, encryption/decryption, and compression. For example, if one system uses ASCII and the other uses EBCDIC, the presentation layer handles the conversion.

ii) Establishes, manages, and terminates sessions

Layer Involved: Session Layer (Layer 5)

The session layer is responsible for creating, maintaining, and terminating connections (sessions) between applications. It controls the dialog between systems, including who can transmit data and when. This layer manages sessions during file transfers, remote logins, or video calls.

iii) Ensures reliable transmission of data

Layer Involved: Transport Layer (Layer 4)

This layer ensures complete and error-free data delivery. It provides flow control, error correction, and retransmission in case of loss. For example, TCP (Transmission Control Protocol) operates at this layer to guarantee data reaches its destination in the correct order.

iv) Log-in and log-out procedures

Layer Involved: Session Layer (Layer 5)**

This layer also handles user authentication and authorization procedures. When a user logs in to a remote system, the session layer establishes the connection and ensures that the user has permission to access the system. Logging out ends the session.

v) Provides independence from differences in data representation

Layer Involved: Presentation Layer (Layer 6)

This layer removes the dependency on how data is represented internally. It allows different systems with different data formats to communicate effectively. For instance, it allows a Windows system to communicate with a UNIX system by translating between their different data representations.

1(b). What is a port address? Show the contents of packets and frames at network, data link, and transport layer for each hop (based on Fig. 1):

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

Definition of Port Address:

A **port address** is a 16-bit number used by the **transport layer** to identify specific processes running on a host. It allows multiple applications on the same computer to communicate over the network simultaneously. The combination of an IP address and a port number is known as a **socket** (e.g., A:m and D:n).

Scenario from the Figure:

- Communication is between:
 - **Source:** Process at Computer A with IP A and port number **m**
 - **Destination:** Process at Computer D with IP D and port number **n**
- Intermediate Devices: B, Router R1, C
- Two LANs: LAN 1 (A → B → R1) and LAN 2 (R1 → C → D)

Contents of the Packet and Frame at Each Hop:

At each hop, the **transport layer** and **network layer** headers stay the same, but the **data link layer** headers (MAC addresses) change based on the link.

Hop 1: From A to B

- **Transport Layer:**
 - Source Port = m
 - Destination Port = n
- **Network Layer:**
 - Source IP = A
 - Destination IP = D
- **Data Link Layer:**
 - Source MAC = A
 - Destination MAC = B

Hop 2: From B to R1

- **Transport Layer:**
 - Source Port = m
 - Destination Port = n
- **Network Layer:**
 - Source IP = A
 - Destination IP = D
- **Data Link Layer:**
 - Source MAC = B
 - Destination MAC = R1

Hop 3: From R1 to C

- **Transport Layer:**
 - Source Port = m
 - Destination Port = n
- **Network Layer:**
 - Source IP = A
 - Destination IP = D
- **Data Link Layer:**
 - Source MAC = R1
 - Destination MAC = C

Hop 4: From C to D

- **Transport Layer:**
 - Source Port = m

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

- Destination Port = n
- **Network Layer:**
 - Source IP = A
 - Destination IP = D
- **Data Link Layer:**
 - Source MAC = C
 - Destination MAC = D

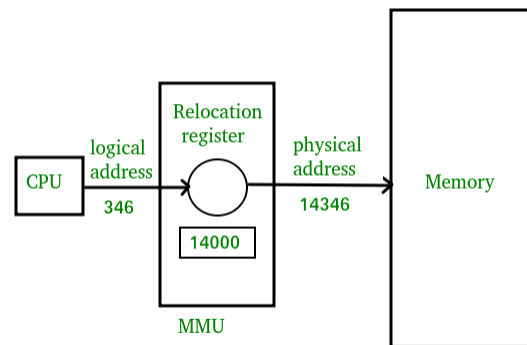
define physical logical and socket address

1. Physical Address

- Also called **MAC Address** (Media Access Control address).
- It is a unique hardware identifier assigned to a network interface card (NIC).
- Used for communication within the same local network (LAN).
- Format: Usually 48 bits (6 bytes), represented in hexadecimal (e.g., 00:1A:2B:3C:4D:5E).
- **Purpose:** Identifies devices on the physical network layer.

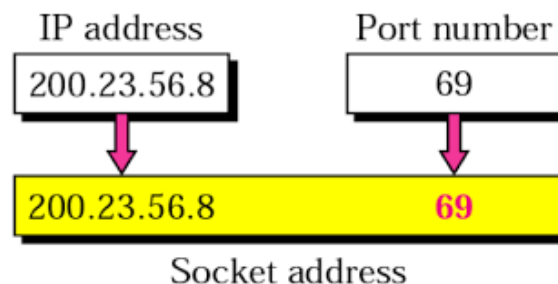
2. Logical Address

- Also called **IP Address** (Internet Protocol address).
- Assigned to devices to identify them on a network logically.
- Used to route data between different networks (across the internet).
- Format: IPv4 (32 bits, e.g., 192.168.1.1) or IPv6 (128 bits).
- **Purpose:** Used for routing packets across interconnected networks.



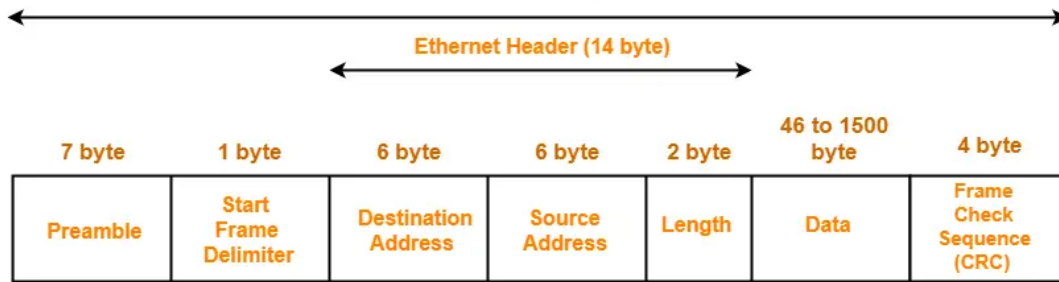
3. Socket Address

- Combination of **IP address** + **Port number**.
- Used to uniquely identify a network endpoint for communication between applications.
- Format: IP_address:Port (e.g., 192.168.1.1:8080).
- **Purpose:** Specifies the sending/receiving application on a device.



depict the frame format of standard ethernet

64 - 1518 byte



IEEE 802.3 Ethernet Frame Format

Field	Size (Bytes)	Description
Preamble	7	Synchronization pattern to allow receiver to lock timing
Start Frame Delimiter (SFD)	1	Marks the start of the frame (10101011)
Destination MAC Address	6	MAC address of the receiving device
Source MAC Address	6	MAC address of the sending device
Type / Length	2	Indicates either the protocol type (EtherType) or the length of the payload
Data / Payload	46–1500	Actual data being transmitted (minimum 46 bytes, maximum 1500 bytes)
Frame Check Sequence (FCS)	4	CRC error-checking code

- **Preamble + SFD** together: 8 bytes, used for synchronization.
- Minimum frame size (excluding preamble and SFD) is 64 bytes.
- If the data is less than 46 bytes, padding bytes are added to meet minimum size.
- The **Type** field indicates the protocol (e.g., 0x0800 for IPv4).

a) When is the Diffie-Hellman algorithm more effective than RSA in public-key cryptography?

Diffie-Hellman is more effective than **RSA** in the following scenarios:

- Key Exchange Only:**
 - Diffie-Hellman is **specifically designed for secure key exchange**, not for encryption or digital signatures.
 - It's faster and more efficient when the goal is just to establish a **shared secret key** over an insecure channel.
- Forward Secrecy:**
 - Diffie-Hellman supports **ephemeral key exchange** (e.g., DHE or ECDHE), which provides **forward secrecy**—even if private keys are compromised later, past sessions remain secure.
 - RSA typically does **not** provide forward secrecy unless combined with additional mechanisms.
- Lower Computational Overhead (in some implementations):**
 - For **ephemeral, short-term communications**, Diffie-Hellman (especially **Elliptic Curve DH**) is faster than RSA with large keys.

b) What are the policies of congestion control in TCP? Explain any of them with necessary diagram [5]

TCP Congestion Control

When many computers try to send data over the internet, the network can become **congested**, just like a traffic jam on a busy road. To handle this, TCP (Transmission Control Protocol) uses **congestion control policies** to avoid sending too much data too quickly.

These policies help maintain a balance between speed and safety, ensuring the network doesn't get overloaded.

Main Congestion Control Policies in TCP:

1. Slow Start
2. Congestion Avoidance
3. Fast Retransmit
4. Fast Recovery
5. AIMD (Additive Increase/Multiplicative Decrease)

Explanation of Slow Start

Why Slow Start?

At the beginning of a TCP connection, the sender doesn't know how much data the network can handle. If it sends too much at once, packets may get dropped, causing delays. **Slow Start** solves this by starting with a small amount of data and increasing it gradually.

How Slow Start Works:

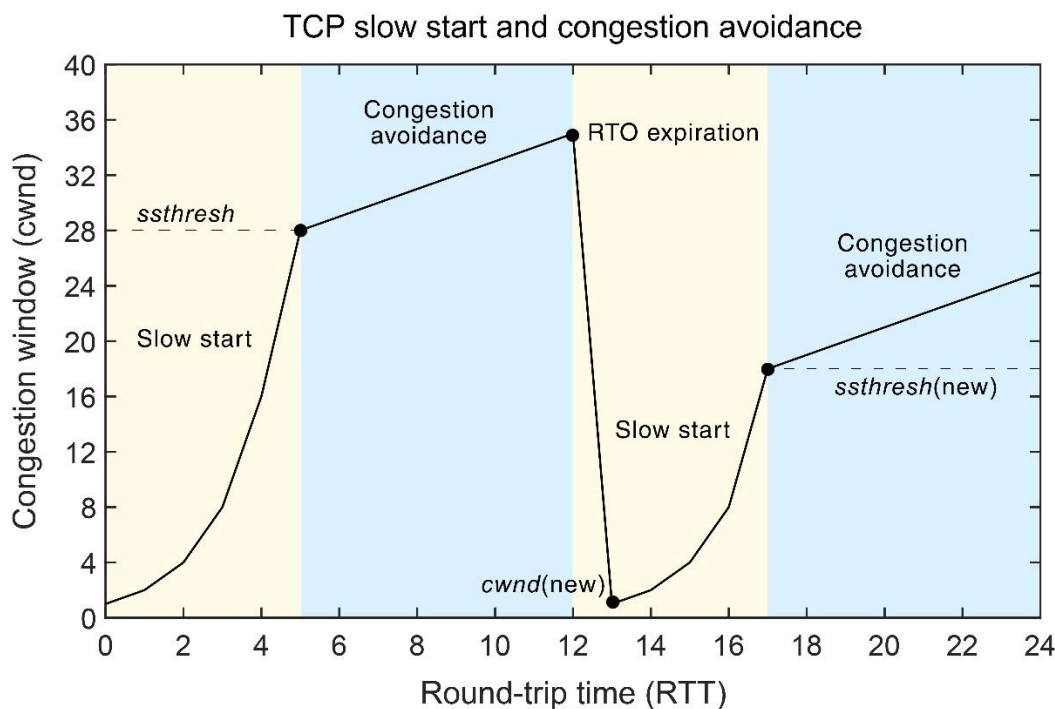
- TCP starts with a small **congestion window (cwnd)**, usually equal to 1 MSS (Maximum Segment Size).
- For each **ACK (Acknowledgment)** received, the congestion window increases.
- This causes **exponential growth**:
 $cwnd = 1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16$, and so on (per Round Trip Time or RTT).
- This continues until either:
 - The **cwnd reaches a threshold value** called **ssthresh**, or
 - A **packet is lost**, which indicates congestion.

What Happens on Packet Loss?

When packet loss occurs:

- TCP assumes there is congestion in the network.
- It reduces the **ssthresh** to half of the current cwnd.
- Then it resets **cwnd to 1** and restarts slow start.

Diagram: Congestion Window Growth in Slow Start



AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

Real-Life Analogy:

Imagine you're filling a glass with water:

- At first, you pour slowly to avoid spilling.
- As you see the glass is not overflowing, you pour faster.
- If water spills, you slow down again.

This is how slow start works — careful at first, then faster, and cautious again if there's trouble.

EXTRA

Slow Start

Purpose: Gradually probe the network to avoid overwhelming it.

Mechanism:

- Starts with a **congestion window (cwnd) = 1 MSS** (Maximum Segment Size).
- **Exponentially increases** cwnd (doubles per RTT) until:
 - A packet loss occurs (timeout/duplicate ACKs), or
 - **Threshold (sssthresh)** is reached (switches to Congestion Avoidance).

2. Congestion Avoidance

Purpose: Stabilize the network after Slow Start.

Mechanism:

- **Linear growth:** Increments cwnd by **1 MSS per RTT** (instead of doubling).
- Continues until congestion is detected (packet loss).
- On loss:
 - **sssthresh = cwnd/2** (new threshold).
 - **cwnd = 1 MSS** (re-enters Slow Start).

Example:

If cwnd = 16 MSS when loss occurs:

- New **sssthresh = 8 MSS**.
- Restart with **cwnd = 1 MSS**.

3. Fast Retransmit

Purpose: Reduce recovery time from packet loss.

Mechanism:

- If **3 duplicate ACKs** are received, TCP retransmits the lost packet **immediately** (without waiting for timeout).
- Avoids unnecessary delays.

4. Fast Recovery

Purpose: Maintain throughput after Fast Retransmit.

Mechanism:

- After retransmission, **cwnd = sssthresh + 3 MSS** (accounts for delivered packets).
- Continues in **Congestion Avoidance** mode (linear growth).

A switch uses a filtering table; a router uses a routing table. Can you explain the difference?

1. Switch – Filtering Table

Purpose:

A **switch** operates at **Layer 2 (Data Link Layer)** of the OSI model. It uses a **filtering table** (also called a MAC address table or forwarding table) to decide **which port** to forward a frame to.

How it works:

AVAILABLE AT:

Onebyzero Edu - Organized Learning, Smooth Career

The Comprehensive Academic Study Platform for University Students in Bangladesh (www.onebyzeroedu.com)

- The table maps **MAC addresses** to specific **switch ports**.
- When a frame arrives:
 - The switch **checks the destination MAC address**.
 - If it finds a match in its table, it **forwards the frame only to that port**.
 - If no match is found, it **floods** the frame to all ports except the source.

Example:

If MAC address AA:BB:CC:DD:EE:FF is connected to **Port 3**, the switch will send frames for that MAC only to Port 3.

2. Router – Routing Table

Purpose:

A **router** operates at **Layer 3 (Network Layer)**. It uses a **routing table** to decide **which network** to forward a packet to and **which interface** to use.

How it works:

- The table contains mappings of **IP address prefixes** to **next-hop addresses or interfaces**.
- When a packet arrives:
 - The router **checks the destination IP address**.
 - It finds the **best matching route** in the table.
 - Then it forwards the packet to the **next hop** or correct network.

Example:

If the routing table says 192.168.1.0/24 goes out **Interface 1**, any packet destined for 192.168.1.25 will be sent through that interface.