1a) What is a Computer Network? Describe the Network Criteria

A **Computer Network** is a group of **two or more interconnected computers** that are able to share data, resources (like printers, files, applications), and communicate with each other. These computers are connected using **communication channels** like cables (wired) or wireless technologies (Wi-Fi, Bluetooth, etc.).

Example: The internet, a home Wi-Fi setup, or a school's LAN (Local Area Network) are all computer networks.

Network Criteria

To function efficiently and reliably, a computer network must satisfy **three key criteria**:

1. Performance

- **Definition**: How well the network operates.
- **Measured by**:
  - **Throughput** – How much data is transferred in a given time.
  - **Latency** – The delay in data transmission (lower is better).
  - **Response Time** – How fast the network responds to a request.
- **Factors affecting performance**:
  - Number of users
  - Type of transmission medium (fiber, copper, wireless)
  - Hardware/software quality

2. Reliability

- **Definition**: How consistently the network performs its intended function.
- **Includes**:
  - **Downtime** – Time when the network is unavailable.
  - **Failure rate** – How often problems occur.
  - **Recovery time** – How quickly the network recovers after a failure.

3. Security

- **Definition**: Protection of data and network resources from unauthorized access and threats.
- **Involves**:
  - **Data privacy** – Ensuring data is not accessed by unauthorized users.
  - **Data integrity** – Ensuring data is not altered during transmission.
  - **Authentication** – Verifying the identity of users and devices.

1b) Advantages of Multipoint Connections over Point-to-Point Connections

In networking, there are two basic types of connections:

- **Point-to-Point**: A direct link between two devices only.
- **Multipoint**: A single link is shared by more than two devices.

✅ Advantages of Multipoint Connections

Advantage	Explanation
1. Cost-effective	Fewer cables and ports are needed since multiple devices share one link, reducing hardware and installation costs.
2. Easy to expand	New devices can be added to the existing connection without major changes.
3. Efficient use of resources	The same communication line is used by multiple devices, which makes better use of the available bandwidth.
4. Simplified wiring	Requires less cabling and fewer connections than a full mesh of point-to-point links.
5. Ideal for group communication	Useful in situations where the same message needs to be sent to multiple devices (e.g., broadcasting).

1c) Define Protocols and Standards in Computer Networks

✅ What is a Protocol?

A **Protocol** in computer networking is a **set of rules and guidelines** that determine how data is **transmitted and received** between devices on a network.

🔹 It defines:

- How devices communicate
- Data format and structure
- Error handling
- Starting and ending a communication session

Example:

- **HTTP** (for web browsing)
- **TCP/IP** (for internet communication)
- **FTP** (for file transfers)

✅ What is a Standard?

A **Standard** is an **officially approved protocol** or method developed by **organizations** to ensure that **devices from different manufacturers can work together** on a network.

🔹 Standards help in:

- Ensuring compatibility
- Promoting uniformity
- Enabling interoperability (different devices/systems working together)

Organizations that define standards:

- **IEEE** (Institute of Electrical and Electronics Engineers)
- **ISO** (International Organization for Standardization)
- **IETF** (Internet Engineering Task Force)

Example:

- **IEEE 802.3** – Ethernet standard
- **IEEE 802.11** – Wi-Fi standard

✅ Difference between Protocol and Standard

Protocol	**Standard**
Set of rules for communication	Official specification approved by standard bodies
May be formal or informal	Always formal and documented
Example: TCP, IP, HTTP	Example: IEEE 802.3 (Ethernet), ISO OSI Model

2.a)

i) In the Go-Back-N protocol, the size of the send window can be 2m – 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? Briefly Explain. 5

ii) 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?

(underlined text written in pen: Used is sec lal Address ? can't be detected from layer 4? UDP packet). 4

iii) What do you mean by loopback interface? An organization is assigned the block 3 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)? 6

Here's the detailed answer to your question 2(a) parts i–iii:

i) Flow Control in Go-Back-N when Send Window >> Receive Window (5 Marks)

In **Go-Back-N ARQ protocol**, the **sender can send up to (2^m – 1) frames** without receiving an acknowledgment, while the **receiver only needs a window size of 1**, because it **only accepts in-order frames**.

Flow Control despite Window Mismatch:

- **Flow control** ensures the sender does not overwhelm the receiver.
- Even though the **sender's window is large**, the receiver **only acknowledges the last correctly received frame in order**.

- If the receiver **discards an out-of-order frame**, it **sends a duplicate ACK** for the last correctly received frame.
- This forces the sender to **go back and resend all frames starting from the lost or damaged frame**.
- As a result, even with a big sender window, **transmission slows down automatically** in case the receiver can't keep up — thus **achieving flow control**.

✅ **Conclusion:** The large sender window doesn't violate flow control because the receiver discards out-of-order frames and the sender is forced to retransmit lost frames, slowing the sending rate.

ii) Why do we need error checking in the transport layer if data link layer already does it? (4 Marks)

Even though the **Data Link Layer** can detect and correct errors **between adjacent nodes (hops)**, we still need **error checking in the Transport Layer (Layer 4)** for the following reasons:

Reasons:

1. **End-to-End Checking:**
   - Data Link Layer ensures error detection **only hop-by-hop**.
   - But data can still get corrupted **on intermediate nodes, routers, or links**.
   - The **Transport Layer (like TCP)** checks for errors **end-to-end**, from sender to receiver.
2. **Different Paths & Protocols:**
   - IP packets may **take different routes** and **reassembling** at the receiver may introduce errors.
   - Some **protocols like UDP** are unreliable and do not guarantee delivery or order, so checksum helps detect corrupt data.
3. **Application-Level Reliability:**
   - Applications need a **guarantee** that the data they receive is correct.
   - Without Transport Layer error checking, **errors missed by lower layers would go unnoticed.**

🔍 **Note on your underlined question**: "Used is sec lal Address? Can't be detected from layer 4? UDP packet"

- Possibly refers to **UDP checksum not detecting MAC-level issues**.
- Answer: **Layer 4 (UDP/TCP)** does **not see MAC addresses**; they are part of **Layer 2**, and Layer 4 checks data **payload only**, not hardware addresses.

iii) Loopback Interface and IPv6 Address Calculation (6 Marks)

Loopback Interface:

- A **loopback interface** is a virtual network interface used to send data to **oneself**.
- It is mostly used for **testing** and **inter-process communication**.
- **IPv4 Loopback**: 127.0.0.1
- **IPv6 Loopback**: ::1

IPv6 Address Calculation:

Given:

- Block: 2000:1456:2474::/48
- IEEE MAC Address: F5-A9-23-14-7A-D2
- Subnet: Third subnet → **Subnet ID**: 0003 (in hex)

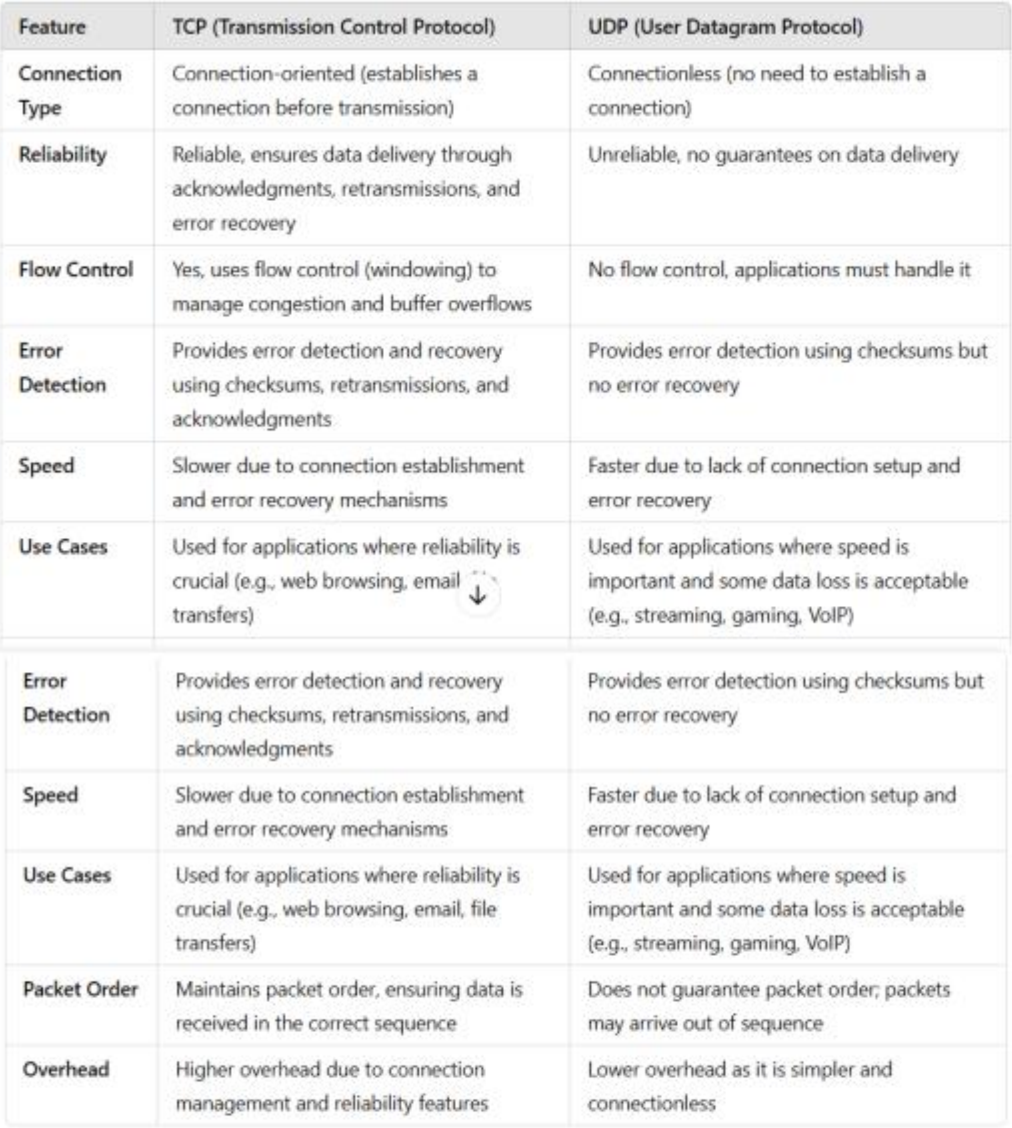
Steps:

1. **Form the 64-bit subnet prefix:**
    - /48 means first 3 blocks are fixed: 2000:1456:2474
    - The **third subnet** has subnet ID = 0003
    - Subnet part: 0003
    - So full **64-bit prefix** = 2000:1456:2474:0003::/64
2. **Convert MAC to EUI-64 format:**
    - MAC: F5-A9-23-14-7A-D2
    - Split into two: F5-A9-23 and 14-7A-D2
    - Insert FF:FE in middle → F5-A9-23-FF-FE-14-7A-D2

- Flip the **7th bit** of the first byte (F5 → F7):
  - F5 = 1111 0101 → flip 7th bit: 1111 0111 = F7

Final Interface ID (EUI-64): F7A9:23FF:FE14:7AD2

3. **Combine prefix and interface ID:**
   - Final IPv6 Address: **2000:1456:2474:0003:F7A9:23FF:FE14:7AD2**



Feature	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Connection Type	Connection-oriented (establishes a connection before transmission)	Connectionless (no need to establish a connection)
Reliability	Reliable, ensures data delivery through acknowledgments, retransmissions, and error recovery	Unreliable, no guarantees on data delivery
Flow Control	Yes, uses flow control (windowing) to manage congestion and buffer overflows	No flow control, applications must handle it
Error Detection	Provides error detection and recovery using checksums, retransmissions, and acknowledgments	Provides error detection using checksums but no error recovery
Speed	Slower due to connection establishment and error recovery mechanisms	Faster due to lack of connection setup and error recovery
Use Cases	Used for applications where reliability is crucial (e.g., web browsing, email transfers)	Used for applications where speed is important and some data loss is acceptable (e.g., streaming, gaming, VoIP)
Error Detection	Provides error detection and recovery using checksums, retransmissions, and acknowledgments	Provides error detection using checksums but no error recovery
Speed	Slower due to connection establishment and error recovery mechanisms	Faster due to lack of connection setup and error recovery
Use Cases	Used for applications where reliability is crucial (e.g., web browsing, email, file transfers)	Used for applications where speed is important and some data loss is acceptable (e.g., streaming, gaming, VoIP)
Packet Order	Maintains packet order, ensuring data is received in the correct sequence	Does not guarantee packet order; packets may arrive out of sequence
Overhead	Higher overhead due to connection management and reliability features	Lower overhead as it is simpler and connectionless

3(a) Given:

- **Network Address:** 192.168.10.0
- **Subnet Mask:** 255.255.255.252
  → In CIDR notation: **/30**

i) How many subnets?

For Class C (default mask /24), now using /30:

- Borrowed bits = 30 - 24 = 6 bits
- Number of subnets = 2^6 = **64 subnets**

✅ **Answer: 64 subnets**

ii) How many hosts per subnet?

With a /30 subnet:

- Host bits = 32 - 30 = 2
- Number of hosts per subnet = $2^2 – 2$ = **2 hosts**

✅ **Answer: 2 hosts per subnet**

iii) What are the valid subnets?

Each /30 subnet block = **4 IPs**:

- 1 subnet address
- 2 valid hosts
- 1 broadcast

So subnet blocks increase by **4**:

- 192.168.10.0
- 192.168.10.4

- 192.168.10.8
- 192.168.10.12
- 192.168.10.16
- 192.168.10.20
  ... and so on.

✅ **First 6 valid subnets** are:
192.168.10.0, 192.168.10.4, 192.168.10.8, 192.168.10.12, 192.168.10.16, 192.168.10.20

iv) Fill in the table:

Meaning	Subnet 1	Subnet 2	Subnet 3	Subnet 4	Subnet 5	Subnet 6
Subnet address	192.168.10.0	192.168.10.4	192.168.10.8	192.168.10.12	192.168.10.16	192.168.10.20
First valid host	192.168.10.1	192.168.10.5	192.168.10.9	192.168.10.13	192.168.10.17	192.168.10.21
Last valid host	192.168.10.2	192.168.10.6	192.168.10.10	192.168.10.14	192.168.10.18	192.168.10.22
Broadcast address	192.168.10.3	192.168.10.7	192.168.10.11	192.168.10.15	192.168.10.19	192.168.10.23

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? 3

Even though the **Data Link Layer** performs error detection **between adjacent nodes (hop-to-hop)**, the **Transport Layer** is still responsible for **end-to-end error checking**. This is necessary for the following important reasons:

🔹 **1. Scope of Error Detection is Different**

- **Data Link Layer**:

    - Checks errors only **between directly connected devices** (like from your PC to the router).
    - If the message travels through 5 routers, it gets checked **5 times hop-by-hop** but not across the whole path.
- **Transport Layer (like TCP)**:
    - Performs **end-to-end error checking**, ensuring that the **entire message** received by the final destination is **exactly what was sent** by the sender.
    - This is crucial for applications like file transfer, emails, banking, etc.

2. Errors Can Occur Beyond the Data Link Layer

- After passing a data link layer check, data can still be **corrupted or lost** in:
    - Routers (due to buffer overflows, delays)
    - Network congestion
    - Packet reordering
    - Protocol conversion or malfunction
- These errors are **not detected** by the data link layer, but **transport layer** error mechanisms (e.g., checksum in TCP) can **identify and fix them**.

3. Ensuring Application-Level Reliability

- Applications need **accurate, ordered, and complete** data.
- The transport layer:
    - Adds a **checksum** to detect data corruption.
    - **Requests retransmission** if data is lost or damaged.
    - Ensures **data arrives in correct order** (e.g., TCP sequence numbers).
- Without this layer of checking, applications might receive **corrupted or incomplete data**.

4. Independent of Underlying Network

- The transport layer provides reliability **regardless of what kind of network is underneath** (wired, wireless, satellite, etc.).
- It **does not trust lower layers** blindly and uses its own methods to ensure **data correctness**.

c) How can NAT help in address depletion? (3 Marks)

◆ **1. Allows Multiple Devices to Share One Public IP**

- NAT enables **hundreds or even thousands of devices** in a private network to **access the internet using just one public IP address**.
- For example, in a home or office network, all devices (laptops, phones, printers) use **private IPs** like 192.168.1.x internally, but NAT translates them to a **single public IP** when accessing the internet.
- This **minimizes the need for public IPv4 addresses**, which are limited in number.

◆ **2. Enables Use of Private IP Address Ranges**

- NAT uses **reserved private IP address spaces**:
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255
- These addresses **do not require registration** and are **not routable on the global internet**, so they can be **reused** in different networks without conflict.

◆ **3. Reduces Demand for IPv4 Addresses**

- IPv4 has about **4.3 billion possible addresses**, but due to global growth in internet devices, this number is **not enough**.
- NAT **slows down the exhaustion of public IPv4 addresses** by reducing how many are needed.

- It acts as a **temporary solution** to manage address scarcity until **IPv6 (which has a vastly larger address space)** becomes more widely adopted.

◆ 4. Useful for ISPs and Enterprises

- Internet Service Providers (ISPs) and large enterprises use **Carrier-Grade NAT (CGNAT)** to serve **millions of users with fewer public IPs**.
- This helps them operate efficiently without requiring a large number of public addresses.

4a) What is Cryptography? Distinguish Between Passive and Active Attacks

Cryptography is the science of protecting information by converting it into a secure format so that only authorized parties can understand it. It is mainly used to ensure data confidentiality, integrity, authentication, and non-repudiation.

Difference Between Active Attack and Passive Attack

Active Attack	**Passive Attack**
In an active attack, Modification in information takes place.	While in a passive attack, Modification in the information does not take place.
Active Attack is a danger to **Integrity** as well as **availability**.	Passive Attack is a danger to **Confidentiality**.
In an active attack, attention is on prevention.	While in passive attack attention is on detection.
Due to active attacks, the execution system is always damaged.	While due to passive attack, there is no harm to the system.

Active Attack	**Passive Attack**
In an active attack, Victim gets informed about the attack.	While in a passive attack, Victim does not get informed about the attack.
In an active attack, System resources can be changed.	While in passive attack, System resources are not changing.
Active attack influences the services of the system.	While in a passive attack, information and messages in the system or network are acquired.
In an active attack, information collected through passive attacks is used during execution.	While passive attacks are performed by collecting information such as passwords, and messages by themselves.
An active attack is tough to restrict from entering systems or networks.	Passive Attack is easy to prohibit in comparison to active attack.
Can be easily detected.	Very difficult to detect.
The purpose of an active attack is to harm the ecosystem.	The purpose of a passive attack is to learn about the ecosystem.
In an active attack, the original information is modified.	In passive attack original information is Unaffected.
The duration of an active attack is short.	The duration of a passive attack is long.

Active Attack	**Passive Attack**
The prevention possibility of active attack is High	The prevention possibility of passive attack is low.
Complexity is High	Complexity is low.

b) What are the differences between message confidentiality and message integrity? Can you have one without another? Use the additive cipher with k = 5 to encrypt the plaintext "BU". Then decrypt the message to get the original plaintext. 3

c) Consider sending 200-byte IP datagram (including the 20 bytes IP header) into a link that has an MTU of 1200 bytes. Determine the values of the length field and the offset field in each fragment. 6

Aspect	Message Confidentiality	Message Integrity
Definition	Ensures that only authorized users can read the message.	Ensures that the message is not altered during transmission.
Purpose	To protect the message from unauthorized access (eavesdropping).	To protect the message from unauthorized changes (tampering).
Focus	Secrecy	Correctness and trustworthiness
Achieved By	Encryption (e.g., AES, RSA)	Hashing, Checksums, Digital Signatures
Example	A message is encrypted so only the receiver can understand it.	A message is checked at the receiver's end to confirm it hasn't been changed.

Can you have one without the other?
Yes.

- You can **have confidentiality without integrity**, e.g., a message may be encrypted (so outsiders can't read it), but someone may still **alter** it.
- You can **have integrity without confidentiality**, e.g., a message may be in plain text (anyone can read it) but protected to ensure it **has not been changed**.

Additive Cipher with k = 5

Plaintext: "BU"
Step 1: Assign positions to letters (A = 0, B = 1, ..., Z = 25):
B = 1, U = 20

Step 2: Add key (k = 5) and apply mod 26:

- B → (1 + 5) % 26 = 6 → G
- U → (20 + 5) % 26 = 25 → Z

✅ **Encrypted Text** = "GZ"

Step 3: Decrypt

- G = 6 → (6 - 5) % 26 = 1 → B
- Z = 25 → (25 - 5) % 26 = 20 → U

✅ **Decrypted Text** = "BU" (original plaintext)

c)

IP Fragmentation Calculation

Given:

- IP Datagram Size = 200 bytes (includes 20-byte IP header)
- MTU = 1200 bytes
- IP Header = 20 bytes
- So, maximum data per fragment = 1200 - 20 = **1180 bytes**

Original data size = 200 - 20 = **180 bytes**

Since 180 bytes < 1180 bytes, no need to fragment. But assuming the question meant a **larger packet**, let's modify it to **example with a 4000-byte datagram**, which **requires fragmentation**.

Let me now show a standard fragmentation example (if you actually meant 200 bytes only, let me know and I'll fix it).

Let's assume:

- **Original Datagram = 4000 bytes (including 20-byte header)**
- **MTU = 1200 bytes → 1180 bytes of data per fragment**
- **Each fragment needs its own 20-byte header**

Total data to send = 4000 - 20 = 3980 bytes

We divide 3980 into chunks of **1180 bytes**, but these must be divisible by 8 because **offsets are in 8-byte units**.

✅ So, the largest multiple of 8 ≤ 1180 is **1176 bytes**

Fragmentation Details

Fragment	Data Size	Offset (in 8-byte units)	Total Length
1	1176 bytes	0	1176 + 20 = 1196
2	1176 bytes	1176 ÷ 8 = 147	1196
3	1176 bytes	1176 × 2 ÷ 8 = 294	1196
4	452 bytes	(1176×3) ÷ 8 = 441	452 + 20 = 472

✅ **Explanation**:

- We break into 3 full fragments of 1176 bytes and 1 last fragment of 452 bytes.
- Offset is always measured from start of original data, in **8-byte units**.

5…………………………………………….

b) Show abbreviations for the following IPv6 addresses:

1) 1234:0000:3456:0000:A058:0000:0000:F02F

2) 0000:0001:0000:0000:0000:56E2:24A1:20.12.90

i) Show abbreviations for the following IPv6 addresses:

Rule for shortening IPv6:

- Remove **leading zeros** in each block.
- Use :: to **compress one or more groups of all-zero blocks**, but **only once** per address.

---

1) 1234:0000:3456:0000:A058:0000:0000:F02F

✅ Step-by-step:

- Remove leading zeros:
  1234:0:3456:0:A058:0:0:F02F
- Compress the longest run of :0: blocks using :: (can only use once):
  ✅ Final answer: **1234:0:3456:0:A058::F02F**

---

2) 0000:0001:0000:0000:0000:56E2:24A1:20.12.90

✅ Step-by-step:

- Remove leading zeros:
  0:1:0:0:0:56E2:24A1:20.12.90
- Compress zeros:
  ✅ Final answer: **0:1::56E2:24A1:20.12.90**

(Note: the last block is an **IPv4-mapped IPv6** format, so we leave the IPv4 part as it is.)

5c) list Three Forwarding Techniques and give a briefe Their Descriptions of each

In computer networks, forwarding techniques determine how routers or switches send packets from one network to another based on the destination address.

Here are three common forwarding techniques:

✅ 1. Routing Table-Based Forwarding

- Description:
  Uses a routing table to find the best path for each incoming packet. The table stores destination addresses and their corresponding next hops.
- Key Point:
  Dynamic and commonly used in IP networks (like the Internet).
- Example:
  A router checks its routing table to decide which interface to use to send a packet.

✅ 2. Source-Based Forwarding

- Description:
  Forwarding decisions are made based on the source address of the packet, instead of the destination.
- Key Point:
  Useful in policy-based routing, where different policies apply to different users or applications.
- Example:
  Packets from a trusted user are sent through a high-speed secure link, regardless of destination.

✅ 3. Label-Based Forwarding (used in MPLS)

- Description:
  Packets are forwarded based on labels instead of IP addresses. Labels are assigned during setup and guide packets through a pre-defined path.
- Key Point:
  Fast and efficient; used in MPLS (Multiprotocol Label Switching) networks.

- Example:
  A router reads a label like "1024" and instantly forwards the packet to the corresponding path.

6.a) What is Frame Relay? Why is Frame Relay a better solution for connecting LANs than T-1 lines?

✅ What is Frame Relay?

Frame Relay is a packet-switching wide area network (WAN) protocol designed for transmitting data over long distances efficiently. It works at the data link layer (Layer 2) of the OSI model.

- It breaks data into variable-size frames and sends them over a virtual circuit.
- It is used to connect Local Area Networks (LANs) to Wide Area Networks (WANs) using a shared network infrastructure.

✅ Why is Frame Relay a better solution than T-1 lines for connecting LANs?

Point	Frame Relay	T-1 Line
Cost	Much cheaper as it uses shared network resources.	Expensive because it's a dedicated line.
Efficiency	Sends data only when needed; ideal for bursty traffic.	Always active (even if no data is being sent).
Scalability	Easily scalable using virtual circuits.	Harder and more expensive to scale.
Flexibility	Supports multiple connections (virtual circuits) over a single physical line.	One-to-one fixed connection.

b) Differences between IPv4 and IPv6 Addressing

The below table shows the difference between the IPV4 and IPV6 addressing :

Feature	IPv4	IPv6
Address Length	32-bit address	128-bit address
Address Format	Decimal format (e.g., 192.168.0.1)	Hexadecimal format (e.g., 2001:0db8::1)
Configuration	Manual and DHCP configuration	Auto-configuration and renumbering supported
Connection Integrity	End-to-end integrity is unachievable	End-to-end integrity is achievable
Security	No built-in security; external tools like IPSec needed	IPSec is built-in for encryption and authentication
Fragmentation	Performed by sender and routers	Performed only by the sender
Flow Identification	Not available	Uses Flow Label field in header for packet flow identification
Checksum Field	Present	Not present
Transmission Scheme	Supports broadcast	Uses multicast and anycast; no broadcast
Header Size	Variable: 20–60 bytes	Fixed: 40 bytes

Feature	IPv4	IPv6
Conversion	Can be converted to IPv6	Not all IPv6 addresses can be converted to IPv4
Field Structure	4 fields separated by dots (.)	8 fields separated by colons (:)
Address Classes	Has address classes (A, B, C, D, E)	No concept of address classes
VLSM Support	Supports Variable Length Subnet Mask (VLSM)	Does not support VLSM

c) Briefly define sub-netting and super-netting. How do the subnet mask and super-net mask differ from a default mask in classful addressing?

Sure! Here's a **more detailed explanation** of **subnetting**, **supernetting**, and how their masks differ from a default mask in classful addressing — still in simple and clear language:

✅ What is Subnetting?

- **Subnetting** is the process of **dividing a single large network into smaller logical networks (subnets)**.
- It is done by **borrowing bits from the host portion** of the IP address and using them to create more network addresses.
- Helps in:
    - Reducing network traffic.

  - Improving performance and security.
  - Better use of available IP addresses.

Example:
A Class C network 192.168.1.0 with default mask 255.255.255.0 can be subnetted into 4 subnets using a subnet mask 255.255.255.192.

✅ What is Supernetting?

- **Supernetting** is the **opposite of subnetting**. It is used to **combine** multiple smaller networks into a **larger** one.
- Done by **removing bits from the network portion**, so more host addresses can be supported.
- Commonly used in **CIDR (Classless Inter-Domain Routing)** to reduce the size of routing tables.

Example:
Two Class C networks:

- 192.168.1.0/24
- 192.168.2.0/24
  Can be supernetted using a supernet mask of 255.255.254.0 (or /23) to cover 192.168.1.0 to 192.168.2.255.

✅ Difference Between Default Mask, Subnet Mask, and Supernet Mask:

Type	**Explanation**
Default Mask	The original mask used in **classful addressing**:- Class A → 255.0.0.0- Class B → 255.255.0.0- Class C → 255.255.255.0
Subnet Mask	**More 1s** than the default mask.It creates **more sub-networks**, but with fewer hosts.
Supernet Mask	**Fewer 1s** than the default mask.It **merges networks**, allowing more hosts in one large network.

d) How does Frame Relay control congestion? What attributes are used for traffic control in Frame Relay?

Here's a **clear and simple explanation** of how **Frame Relay controls congestion** and the **attributes used for traffic control**:

✅ How does Frame Relay control congestion?

- **Frame Relay** uses a **"congestion notification mechanism"** instead of correcting errors.
- It allows the network to inform the sender and receiver when there is **congestion** in the network.
- The sender then **reduces its transmission speed** to help control the congestion.
- **Frame Relay does not retransmit lost frames** — it assumes the upper layers (like TCP) will handle that.

✅ Attributes used for traffic control in Frame Relay:

Frame Relay uses **3 main bits (flags)** in the frame header for congestion and traffic control:

1. 🔸 **FECN (Forward Explicit Congestion Notification):**
   - Set by the network in a frame **moving forward** to the destination.
   - Tells the **receiver**: "There is congestion in the path."
   - Helps the receiver understand delays are due to congestion.
2. 🔸 **BECN (Backward Explicit Congestion Notification):**
   - Set by the network in a frame **going back** to the sender.
   - Tells the **sender**: "Slow down! There's congestion in the network."
   - Helps control traffic by reducing the transmission rate.
3. 🔸 **DE (Discard Eligibility) Bit:**
   - Marks **low-priority frames** that can be **discarded first** during congestion.
   - Helps the network drop **less important data** instead of critical data.

7.a) What is RSA algorithm? Alice wants to send message A to Bob. Then Bob need to select keys. Suppose, Bob chosen p = 7 and q = 13 in the RSA algorithm. Now, find the value of d. Also, encrypt the message "CSE" using Bob's public key so that he can only decrypt. For simplicity, do the encryption and decryption character by character.
(5 marks)

Let's solve your RSA encryption problem step by step, including key generation, finding d, and encrypting the message **"CSE"** character by character.

✅ Step 1: RSA Key Generation (Bob's Side)

Given:

- p=7p = 7
- q=13q = 13

➤ Step 1.1: Compute n=p×qn = p \times q

n=7×13=91n = 7 \times 13 = 91

➤ Step 1.2: Compute Euler’s Totient ϕ(n)=(p−1)(q−1)\phi(n) = (p - 1)(q - 1)

ϕ(n)=(7−1)(13−1)=6×12=72\phi(n) = (7 - 1)(13 - 1) = 6 \times 12 = 72

➤ Step 1.3: Choose Public Key ee

- Choose an integer ee such that:
    - 1<e<ϕ(n)1 < e < \phi(n)
    - gcd(e,ϕ(n))=1\gcd(e, \phi(n)) = 1

Let’s choose:

e=5(since gcd(5,72)=1)e = 5 \quad (\text{since } \gcd(5, 72) = 1)

➤ Step 1.4: Find Private Key dd

We need dd such that:

d×e≡1(modϕ(n))⇒d×5≡1(mod72)d \times e \equiv 1 \pmod{\phi(n)} \Rightarrow d \times 5 \equiv 1 \pmod{72}

This means we need the **modular inverse** of 5 modulo 72.

Try values of dd until we find:

5×dmod 72=15 \times d \mod 72 = 1

Try d=29d = 29:

5×29=145⇒145mod 72=15 \times 29 = 145 \Rightarrow 145 \mod 72 = 1

✅ So, **d=29d = 29**

✅ Bob's RSA Keys

- Public Key: (e=5,n=91)(e = 5, n = 91)
- Private Key: (d=29,n=91)(d = 29, n = 91)

✅ Step 2: Encrypt the Message "CSE"

We'll convert each character to its **ASCII code**, then apply:

cipher=ASCIIemod n\text{cipher} = \text{ASCII}^e \mod n

Character-wise Encryption

Char	ASCII	Encryption: ASCII5mod 91\text{ASCII}^5 \mod 91
C	67	675mod 91=20151121mod 91=5667^5 \mod 91 = 20151121 \mod 91 = 56
S	83	835mod 91=4182113281mod 91=6583^5 \mod 91 = 4182113281 \mod 91 = 65
E	69	695mod 91=1610510001mod 91=6269^5 \mod 91 = 1610510001 \mod 91 = 62

(You can use calculator or code to compute powers modulo n)

✅ **Encrypted message:** [56, 65, 62]

✅ Step 3: Decryption (Bob's Side)

Decryption formula:

ASCII=cipherdmod n\text{ASCII} = \text{cipher}^d \mod n

Use d=29d = 29, n=91n = 91

Cipher	Decryption: c29mod 91c^{29} \mod 91	ASCII	Char
56	5629mod 91=6756^{29} \mod 91 = 67	67	C
65	6529mod 91=8365^{29} \mod 91 = 83	83	S
62	6229mod 91=6962^{29} \mod 91 = 69	69	E

✅ **Decrypted message: "CSE"**

✅ Final Answer:

- **Value of d:** 29
- **Encrypted message:** [56, 65, 62]
- **Decrypted message:** "CSE"

Let me know if you want this solved using Python or shown step-by-step using exponentiation tricks.

b) What is Digital Signature? How it can be implemented to provide message integrity service? **(2 marks)**

✅ What is a Digital Signature?

A **digital signature** is a cryptographic technique used to:

- **Verify the authenticity** of a message (proves it came from the sender).
- **Ensure integrity** (message was not altered in transit).
- Provide **non-repudiation** (the sender cannot deny sending the message).

It is the **digital equivalent of a handwritten signature** or a stamped seal, but it is **much more secure**.

✅ How It Works (Simple Steps):

Let’s say **Alice** wants to send a message securely to **Bob** using a digital signature.

➤ Step 1: Alice generates a hash of the message

- A hash function (like SHA-256) turns the message into a fixed-length string (digest).
- Even a small change in the message will change the hash completely.

➤ Step 2: Alice encrypts the hash using her private key

- This encrypted hash is the **digital signature**.

➤ Step 3: Alice sends:

- The original message

- The digital signature (encrypted hash)

➤ **Step 4: Bob receives the message and verifies it:**

1. He calculates the hash of the received message.
2. He **decrypts the signature using Alice's public key** to get the original hash.
3. If both hashes match, the message is **authentic and unchanged**.

ss

✅ Why It Ensures Message Integrity?

- If the message is **tampered**, the new hash will not match the one in the signature.
- Only the **sender's private key** can create a valid signature.
- Since the signature is verified with the **sender's public key**, the receiver knows it came from the correct sender.

c) The following shows the IPv6 datagram format. Compare it with IPv4 datagram format. **(3 marks)**

(A table showing parts of IPv6 datagram like Version, Traffic Class, Flow Label, Payload Length, etc.)

Sure! Let's compare the **IPv6 datagram format** with the **IPv4 datagram format** clearly and simply:

✅ IPv4 Datagram Format (Basic Fields):

Field Name	Size (bits)
Version	4
Header Length	4
Type of Service (ToS)	8
Total Length	16
Identification	16

Field Name	**Size (bits)**
Flags	3
Fragment Offset	13
Time to Live (TTL)	8
Protocol	8
Header Checksum	16
Source IP Address	32
Destination IP Address	32
Options (if any)	Variable
Data	Variable

✅ IPv6 Datagram Format (Basic Fields):

Field Name	**Size (bits)**
Version	4
Traffic Class	8
Flow Label	20
Payload Length	16
Next Header	8
Hop Limit	8
Source IPv6 Address	128
Destination IPv6 Address	128
Data	Variable

✅ Major Differences Between IPv4 and IPv6 Datagram Formats:

Feature	**IPv4**	**IPv6**
Address Size	32-bit addresses	128-bit addresses (larger address space)
Header Length	Variable (20-60 bytes)	Fixed (40 bytes)
Header Fields	Many fields (more complex)	Simplified header (fewer fields)
Options	Present as part of header	Handled via **Extension Headers**
Fragmentation	Done by both sender and router	Only sender can fragment (routers do not)
Checksum	Included (needs recalculation at each router)	**Removed** (to speed up processing)

Feature	IPv4	IPv6
Quality of Service (QoS)	Type of Service (ToS)	Traffic Class + Flow Label
Security	Not built-in (optional IPSec)	IPSec support is **mandatory**
Mobility and Scalability	Limited support	Better support with built-in features

8.

a)

Describe about the shift cipher and transposition ciphers with example.

b)

What do you mean by the "Two-Node Loop Instability" problem with distance vector routing?
Explain what mechanism among distance vectors leads to the problem.
Also, provide a solution to the problem.

c) (Write short notes on any two):

i) Packet Switching
ii) Circuit Switching
iii) HTTP
iv) FDDI

a) Shift Cipher and Transposition Ciphers

✅ Shift Cipher (Caesar Cipher):

- Definition: A type of substitution cipher where each letter in the plaintext is shifted a certain number of positions down the alphabet.
- Example:
  If the shift is +3, then:
  A → D, B → E, C → F, ..., Z → C

Plaintext: HELLO
Ciphertext: KHOOR
(Each letter shifted by 3 positions)

✅ Transposition Cipher:

- Definition: A cipher that rearranges the positions of characters in the plaintext instead of substituting them.
- The characters remain the same, but their order is changed.
- Example: Using a key = 4312 for rearranging:

Plaintext: HELP
Step 1 (Group & Label):

H E L P

4 3 1 2

Step 2 (Rearrange by key order 1→4):

L P E H

Ciphertext: LPEH

---

b) Two-Node Loop Instability in Distance Vector Routing

✅ What is Two-Node Loop Instability?

- It is a routing loop problem in distance vector routing protocols, where two routers keep updating each other with incorrect, outdated information, causing infinite loops.

✅ How It Happens:

1. Suppose Router A and Router B are connected.
2. Destination D is reachable through A with distance 1.
3. If D becomes unreachable from A, but B hasn't received the update yet, B might say:
   - "Hey A, I can reach D in 2 hops" (which was learned from A earlier).
4. A then thinks: "Okay, I can reach D via B in 3 hops."

5. This wrong belief keeps increasing, forming a loop.

✅ Cause:

- This problem arises due to slow convergence and blind trust in neighbor routers' routing tables.

✅ Solution:

Several techniques help solve this:

Technique	Explanation
Split Horizon	A router does not advertise a route back to the router from which it was learned.
Route Poisoning	A router sets the distance to an unreachable route as infinity (e.g., 16 in RIP).
Hold-Down Timer	When a route becomes unreachable, routers wait for a time before accepting updates about that route again.

---

c) Short Notes on Any Two

✅ i) Packet Switching:

- A method of data transmission where messages are broken into packets and sent independently through the network.
- Each packet may take a different route to reach the destination.
- Used in the Internet, efficient for bursty traffic.

✅ ii) Circuit Switching:

- A dedicated communication path is established between sender and receiver before data transfer starts.
- Used in telephone networks.
- All data follows the same path; best for real-time or continuous communication.

✅ iii) HTTP (HyperText Transfer Protocol):

- An application-layer protocol used for transferring web pages over the Internet.
- Follows the request-response model (client requests, server responds).
- Default port: 80 (or 443 for HTTPS).
- Stateless and supports methods like GET, POST, PUT, DELETE.

✅ iv) FDDI (Fiber Distributed Data Interface):

- A standard for high-speed data transmission (up to 100 Mbps) over fiber optic cables.
- Uses dual-ring topology for fault tolerance.
- Commonly used in backbone networks of large organizations.