COMPUTER NETWORKS

Unit-1

1. Write a short note on LAN, MAN and WAN.

(i)LAN (Local Area Network)

Definition:
A LAN connects computers and devices in a **small area** like a single building, office, school, or home.

Features:

- Covers **few meters to few kilometers**.
- Usually **owned and managed by one organization**.
- High speed (up to **1 Gbps or more**).
- Cost of setup is **low**.
- Can be **wired (Ethernet cables)** or **wireless (Wi-Fi)**.

Examples:

- Computers connected in a **school computer lab**.
- All employees' PCs connected in an **office**.
- Wi-Fi network in your **home**.

(ii) MAN (Metropolitan Area Network)

Definition:
A MAN connects multiple LANs across a **city or large campus**. It's larger than a LAN but smaller than a WAN.

Features:

- Covers **several kilometers to an entire city**.
- Usually maintained by **Internet Service Providers (ISP)** or government.
- Medium speed (higher than WAN, lower than LAN).
- Cost is **medium** (higher than LAN, lower than WAN).

Examples:

- A **city-wide broadband network**.
- Cable TV networks in a city.
- A university connecting all campuses across the city.

(iii) WAN (Wide Area Network)

Definition:
A WAN covers a **very large area**, like countries or the whole world. It connects multiple LANs and MANs together.

Features:

- Covers **thousands of kilometers**.
- Can use **satellites, fiber optic cables, telephone lines**.
- Slower compared to LAN (because of long distance).
- Very **expensive to set up and maintain**.
- Not owned by a single organization, but by **multiple organizations and providers**.

Examples:

- **The Internet** (biggest WAN).
- A multinational company connecting offices in different countries.
- Banking networks that connect ATMs across the world.

Comparison / Difference:

Feature	LAN	MAN	WAN
Full Form	Local Area Network	Metropolitan Area Network	Wide Area Network
Area Size	Small (1 building, office, school)	Medium (city, large campus)	Very large (country, world)
Ownership	One person/organization	ISP or government	Multiple organizations

COMPUTER NETWORKS

Feature	LAN	MAN	WAN
Speed	Very High	Medium	Low (compared to LAN)
Cost	Low	Medium	High
Example	School lab, office Wi-Fi	City broadband, cable TV	Internet, global banking

2. Compare OSI reference model and TCP/IP protocol suite.

Feature	OSI Model	TCP/IP Model
Full Form	Open Systems Interconnection	Transmission Control Protocol / Internet Protocol
Layers	7 layers	4 layers
Layers Name	Application, Presentation, Session, Transport, Network, Data Link, Physical	Application, Transport, Internet, Network Access
Developed by	ISO (International Standards Organization)	DoD (U.S. Department of Defense)
Approach	Theoretical → standard model for teaching and reference	Practical → protocols used in real communication

COMPUTER NETWORKS

Feature	OSI Model	TCP/IP Model
Protocol Dependency	Independent of protocols	Defines actual protocols (HTTP, FTP, TCP, IP, etc.)
Usage	Mainly for understanding and design	Used in real networking (Internet)
Transport Layer	Supports both connection-oriented (TCP) and connectionless (UDP) concepts theoretically	Defines actual TCP (reliable) and UDP (fast, unreliable)
Flexibility	Strictly layered, clear separation of functions	More flexible, some functions combined in one layer

3. Define following terms:
Computer Network, Processing Delay, Queuing Delay.

- 1. Computer Network

  A computer network is a group of computers and devices that are connected together to share information, files, resources (like printers), and the internet.
  👉 Example: Wi-Fi in your home that connects laptops, mobiles, and smart TVs.

- 2. Processing Delay

Processing delay is the time taken by a computer or router to process the data packet.

👉 Example: Checking the packet's address, error checking, or deciding where to send it next.

- 3. Queuing Delay

Queuing delay is the time a packet spends waiting in a queue (line) inside a router or switch before it is forwarded.

👉 Example: Like waiting in a line at a ticket counter when many people are ahead of you.

4. Sketch the diagram of OSI reference model and discuss functionalities of all the layers.

- **Layer 7 — Application**

- What it is: The layer closest to the user. It's where programs (web browser, email client) use the network.
- Main job: Provide network services to applications (web, email, file transfer).
- Examples / protocols: HTTP (web), SMTP (email), FTP (file transfer), DNS (name lookup).
- Simple line: Where applications talk to the network.

- **Layer 6 — Presentation**

- What it is: Prepares data so the application on the other side can understand it.
- Main job: Translate data formats (like from one character set to another), encrypt/decrypt, and compress/decompress data.
- Examples: Converting text formats (ASCII/Unicode), image/video formats (JPEG, MPEG), encryption like TLS (conceptually here).
- Simple line: Makes sure data is in the right format and secure.

COMPUTER NETWORKS

Layer 5 — Session

- What it is: Manages the conversation between two applications.
- Main job: Start, maintain, and end communication sessions; provide checkpoints or synchronization (so long transfers can resume).
- Examples: Managing a login session, controlling dialogs between database and client.
- Simple line: Opens and closes conversations and keeps them organized.

Layer 4 — Transport

- What it is: Ensures reliable (or fast) transfer of data between end systems (computers).
- Main job: Break large messages into smaller pieces (segmentation), provide error checking, flow control, and reassembly.
- Protocols: TCP (reliable, ordered delivery), UDP (fast, no guarantee).
- Simple line: Makes sure data gets from one computer to another correctly.

Layer 3 — Network

- What it is: Handles moving data between different networks (routing).
- Main job: Logical addressing (like IP addresses), finding the best path, and forwarding packets across routers.
- Protocols / devices: IP, ICMP, OSPF, BGP; routers work here.
- Simple line: Finds the way across networks so data reaches the right place.

Layer 2 — Data Link

- What it is: Responsible for node-to-node transfer inside the same local network.
- Main job: Put raw bits into structured frames, use MAC addresses (hardware addresses), detect and sometimes correct errors, and control who can use the medium (e.g., Ethernet rules).
- Sub-layers: MAC (Media Access Control) and LLC (Logical Link Control).
- Examples / devices: Ethernet (IEEE 802.3), Wi-Fi (IEEE 802.11); switches and bridges operate here.
- Simple line: Delivers frames on the local link and handles physical addressing (MAC).

Layer 1 — Physical

- What it is: The lowest layer — sends the actual raw bits over the wire or air.
- Main job: Define electrical/optical/radio signals, cables, connectors, voltages, bit rate, and physical topology.
- Examples / devices: Copper cable, fiber, radio waves, hubs, repeaters, network interface card (NIC).
- Simple line: Sends and receives raw bits through cables or air.

5. Define Network Topology. List all types of topologies. Discuss the concepts of all network topologies.

What is Network Topology?

- **Definition**:
  A **network topology** is the **arrangement (layout or structure)** of how computers, cables, and devices are connected in a network.

COMPUTER NETWORKS

👉 In simple words: It shows **how devices are linked** and how data flows between them.

🔹 Types of Network Topologies

1. **Bus Topology**
2. **Star Topology**
3. **Ring Topology**
4. **Mesh Topology**
5. **Tree Topology**
6. **Hybrid Topology**

🔹 Explanation of Each Topology

1. Bus Topology

- All computers are connected to a **single main cable (backbone)**.
- Data travels in both directions on the cable.
- If the backbone fails → whole network fails.
- **Example:** Small LANs, old Ethernet networks.
- **Advantage:** Cheap and easy to set up.
- **Disadvantage:** Difficult to troubleshoot, backbone failure stops the network.

---

2. Star Topology

- All devices are connected to a **central device (hub or switch)**.
- Communication happens through this central hub.

- **Example:** Modern LANs in offices and homes.
- **Advantage:** Easy to manage; if one computer fails, others still work.
- **Disadvantage:** If the hub/switch fails, the whole network goes down.

---

3. Ring Topology

- Each device is connected to **two other devices**, forming a **closed loop (circle)**.
- Data passes around the ring in one direction until it reaches its destination.
- **Example:** Some old LANs, Token Ring networks.
- **Advantage:** Easy to install, equal access for all computers.
- **Disadvantage:** If one computer/cable fails, the entire ring can break.

---

4. Mesh Topology

- Every device is connected to **every other device** directly.
- Provides multiple paths for data transfer.
- **Example:** Used in military communication, critical systems.
- **Advantage:** Very reliable; if one link fails, data uses another path.
- **Disadvantage:** Very costly and requires a lot of cables.

---

5. Tree Topology

- A combination of **star and bus topologies**.
- Devices are arranged in a hierarchy (like a family tree).

- Used in large networks (like universities).
- **Advantage:** Scalable (easy to add more devices).
- **Disadvantage:** If backbone cable fails, entire network is affected.

---

6. Hybrid Topology

- A **mix of two or more topologies** (e.g., star + mesh, star + ring).
- Used in large organizations where one single topology is not enough.
- **Advantage:** Flexible and reliable.
- **Disadvantage:** Expensive and complex design.

6. Define Computer Network with example & List application used in real life.

🔹 **Definition of Computer Network**

A **computer network** is a system where **two or more computers/devices are connected together** so they can **share data, files, resources (like printers), and internet**.

👉 **Example:** Wi-Fi in your home that connects your **mobile, laptop, and smart TV** to the internet.

🔹 **Applications of Computer Networks in Real Life**

1. **Communication** – Emails, video calls (Zoom, WhatsApp, Teams).
2. **Resource Sharing** – Sharing printers, scanners, and files in offices.

3. **Online Banking** – Transferring money, checking account details.
4. **E-commerce** – Online shopping (Amazon, Flipkart).
5. **Social Networking** – Facebook, Instagram, Twitter (X).
6. **Education** – Online classes, e-learning platforms (Google Classroom, Coursera).
7. **Entertainment** – Watching movies or listening to music online (YouTube, Netflix, Spotify).
8. **Healthcare** – Telemedicine, sharing medical reports between hospitals.
9. **Government & Defense** – For communication, data sharing, and security systems.
10. **Cloud Computing** – Using Google Drive, Dropbox for storing and sharing data online.

7. Define Computer Network with example & List application used in real life.

- **Definition of Computer Network**

A **computer network** is a system where **two or more computers/devices are connected together** so they can **share data, files, resources (like printers), and internet**.

👉 **Example:** Wi-Fi in your home that connects your **mobile, laptop, and smart TV** to the internet.

- **Applications of Computer Networks in Real Life**

1. **Communication** – Emails, video calls (Zoom, WhatsApp, Teams).

2. **Resource Sharing** – Sharing printers, scanners, and files in offices.
3. **Online Banking** – Transferring money, checking account details.
4. **E-commerce** – Online shopping (Amazon, Flipkart).
5. **Social Networking** – Facebook, Instagram, Twitter (X).
6. **Education** – Online classes, e-learning platforms (Google Classroom, Coursera).
7. **Entertainment** – Watching movies or listening to music online (YouTube, Netflix, Spotify).
8. **Healthcare** – Telemedicine, sharing medical reports between hospitals.
9. **Government & Defense** – For communication, data sharing, and security systems.
10. **Cloud Computing** – Using Google Drive, Dropbox for storing and sharing data online.

8. Circuit Switched Network:

- **What is a Circuit Switched Network?**

A circuit-switched network is a type of network where a dedicated path (circuit) is created between two devices before they can communicate.

Once the circuit is set up, the devices can exchange data continuously until the communication ends.

After communication is finished, the circuit is released so others can use it.

- **How it Works (Step by Step)**

Connection Setup: Before communication, a path is established between sender and receiver.

Example: When you dial a phone number, the network sets up a line for your call.

Data Transfer: After the path is ready, data (voice or message) flows continuously.

No one else can use this path during the call.

Connection Termination: When the communication ends (you hang up), the path is closed and freed for others.

- **Example**

Traditional telephone system (landline calls) uses circuit switching.

When you call someone, the network creates a dedicated line between you and the other person, and only you two can use it until the call ends.

- **Features of Circuit Switching**

Dedicated Path: A fixed path is reserved for the whole session.

Continuous Data Flow: No delay once the path is set.

Inefficient Use: Even if you stay silent on a phone call, the path is still blocked and wasted.

Connection-Oriented: Communication only starts after a circuit is established.

9. Packet Switched Network:

- **What is a Packet Switched Network?**

  - In a **packet-switched network**, data is **broken into small packets** before being sent.
  - Each packet can **take different routes** through the network to reach the destination.

- At the receiver's side, all packets are **reassembled** to form the original message.

How it Works (Step by Step)

1. **Data Breakup**
   - The message (text, voice, video, etc.) is split into small **packets**.
   - Each packet contains:
     - Part of the data
     - Source and destination address
2. **Routing**
   - Packets travel **independently** through the network.
   - They may take **different paths** depending on availability.
3. **Reassembly**
   - At the destination, all packets are collected and arranged in the correct order.

Example

- **The Internet** is the best example.
- When you send an email or browse a website, your data is split into packets and sent across the network.
- Packets may travel through different routes but will arrive at the same destination.

Features of Packet Switching

- **No fixed path** → Packets can take any available route.
- **Efficient use** → Network resources are shared by many users.
- **Connectionless or connection-oriented** → Can work in both ways.
- **Fast and flexible** → Best for modern communication like browsing, video calls, and streaming.

COMPUTER NETWORKS

Comparison / Difference (Circuit vs Packet):

Feature	Circuit Switching	Packet Switching
Definition	A dedicated path (circuit) is reserved for the whole communication.	Data is split into packets, and packets are sent independently through the network.
Connection	Connection-oriented (needs setup before communication).	Can be connectionless (packets sent without setup) or connection-oriented.
Resource Use	Wastes resources → path stays busy even if no data is sent.	Efficient → network resources are shared among many users.
Data Transfer	Continuous, without interruption once path is set.	Packets may arrive at different times and need reassembly.
Delay	Small delay (after setup) because data flows directly.	Packets may face delay, disorder, or loss due to congestion.
Reliability	Reliable once circuit is established.	Reliability depends on error checking and retransmission.
Speed	Fixed speed (depends on circuit).	Faster for bursty data (Internet use).
Cost	More expensive (dedicated line needed).	Cheaper and more scalable.
Example	Traditional telephone calls (landline).	Internet (emails, browsing, video streaming).

COMPUTER NETWORKS

10. Transmission Media

What is Transmission Media?

- **Transmission Media** is the **path through which data travels** from one device to another in a network.
- It can be **wired (cables)** or **wireless (waves)**.

Types of Transmission Media

1. Twisted Pair Cable

- **Description:** Two insulated copper wires twisted together.
- **Use:** Telephone networks, LANs (Ethernet).
- **Advantages:** Cheap, easy to install.
- **Disadvantages:** Low speed, affected by noise.
- **Example:** LAN cables (Cat 5, Cat 6).

---

2. Coaxial Cable

- **Description:** Single copper wire in the center, surrounded by insulation, metallic shield, and outer cover.
- **Use:** Cable TV, broadband internet.
- **Advantages:** Better shielding, less noise than twisted pair.
- **Disadvantages:** More expensive than twisted pair.

---

3. Fiber Optic Cable

- **Description:** Thin glass or plastic fibers that transmit **data as light signals**.
- **Use:** High-speed internet, long-distance communication.

- **Advantages:** Very high speed, long distance, immune to electrical noise.
- **Disadvantages:** Expensive, requires special equipment.

---

4. Radio Waves

- **Description:** Electromagnetic waves that travel **through the air**.
- **Use:** Wi-Fi, Bluetooth, mobile communication.
- **Advantages:** Wireless, no cables needed.
- **Disadvantages:** Limited distance, affected by interference.

---

5. Microwave

- **Description:** High-frequency radio waves used for **line-of-sight communication**.
- **Use:** Satellite communication, TV signals, long-distance telephone.
- **Advantages:** Can travel long distances via satellites.
- **Disadvantages:** Needs clear line-of-sight, expensive equipment.

---

6. Infrared Waves

- **Description:** Light waves with wavelengths longer than visible light.
- **Use:** Remote controls, short-distance device communication.
- **Advantages:** Wireless, safe for short distances.
- **Disadvantages:** Short range, cannot pass through obstacles.

COMPUTER NETWORKS

11. TCP/IP Model

◆ What is TCP/IP Model?

- **TCP/IP = Transmission Control Protocol / Internet Protocol**
- It is the **basic protocol suite of the Internet**.
- It defines **how data is sent, received, and routed** across networks.
- It is a **practical model**, unlike OSI which is theoretical.

◆ Layers of TCP/IP Model

Layer	Function	Protocols / Examples
1. Application Layer	Provides services to **end users**. Programs use this layer to communicate.	HTTP (web), FTP (file transfer), SMTP (email), DNS (name resolution)
2. Transport Layer	Ensures **reliable or fast delivery** of data between devices.	TCP (reliable, connection-oriented), UDP (fast, connectionless)
3. Internet Layer	Handles **logical addressing and routing** of data across networks.	IP (addressing), ICMP (error messages), ARP (address resolution)
4. Network Access / Link Layer	Deals with **physical transmission of data** over cables or wireless.	Ethernet, Wi-Fi, switches, NICs

---

◆ How TCP/IP Works (Simple Example)

1. **Application Layer:** You open a browser and type www.google.com.
2. **Transport Layer:** TCP splits the web page data into segments.

3. **Internet Layer:** IP adds the **source and destination addresses**.
4. **Network Access Layer:** Data is converted to **bits** and sent over Wi-Fi or cable.

At the receiver side, all layers **reverse the process** to reconstruct the original data.

- **Key Features**
  - **Reliable communication** (TCP) and **fast communication** (UDP).
  - **Routing and addressing** handled by IP.
  - **Supports real-world Internet protocols**.
  - **Flexible and widely used** worldwide.

Unit-2

1. Write the full form of DNS. List all types of DNS Components. Explain any 2 components.

- **Full Form of DNS**

DNS = Domain Name System

👉 It is like the **phonebook of the Internet** – it converts human-readable names (like www.google.com) into IP addresses (like 142.250.190.68) so computers can understand.

- **Components of DNS**

The main components of DNS are:

1. **DNS Resolver (Stub Resolver)**
2. **DNS Root Server**
3. **Top-Level Domain (TLD) Server**

4. **Authoritative Name Server**

- **Explanation of Any 2 Components**

1. DNS Resolver (Stub Resolver)

- This is the **first step** in the DNS process.
- It is usually provided by your **Internet Service Provider (ISP)**.
- When you type a website name (e.g., www.google.com), the resolver sends the query to DNS servers to find the IP address.
- **Example:** Airtel/Jio/BSNL DNS resolvers that your computer or phone uses.

---

2. Authoritative Name Server

- This server has the **final answer** for a domain name.
- It stores the actual DNS records (like IP addresses, mail server info) of a website.
- Example: If you search www.openai.com, the authoritative server for openai.com gives the exact IP address back to your resolver.
- **Simple line:** It's like the **official record keeper** of the domain.

2. What is the need of FTP? Discuss working of FTP.

- **What is the Need of FTP?**

FTP (File Transfer Protocol) is needed to:

- Transfer files **between two computers** over the Internet or a network.
- Upload files from your computer to a **web server**.
- Download files from a server to your **computer**.
- Share large files (like software, documents, images) safely and easily.

👉 Example: When a web developer uploads website files (HTML, CSS, images) to a hosting server, they use FTP.

- **Working of FTP**

  FTP works on a **Client-Server model** using two connections:

  Control Connection (Port 21) – Used for commands (like login, upload, download).

  Data Connection (Port 20) – Used for transferring actual files.

  Steps in FTP Working:

1. **User Connects to FTP Server** – Client (your computer) connects to the FTP server using a username & password.
2. **Authentication** – Server checks login details and allows access.
3. **File Operations** – User can upload, download, rename, or delete files on the server.
4. **Transfer of Files** – Data is sent through the **data connection**.
5. **Session Close** – After work is done, the user disconnects.

3. Distinguish persistent and non-persistent http.

Point	Non-Persistent HTTP	Persistent HTTP
Connection	Creates a **new TCP connection** for every request (like HTML, image, CSS).	Uses **one single TCP connection** for multiple requests and responses.
Efficiency	Slow, because each file needs a separate connection.	Faster, because many files are transferred in the same connection.

Point	Non-Persistent HTTP	Persistent HTTP
Overhead	More overhead (extra time for opening & closing connections).	Less overhead (connection is reused).
Usage	Older method (HTTP/1.0).	Modern method (HTTP/1.1 and later).
Example	If a webpage has 5 images, it will open 6 separate connections (1 for HTML + 5 for images).	If a webpage has 5 images, all can be fetched in **one connection**.

4. Discuss the DORA process in DHCP.

- **DHCP and DORA**

DHCP (Dynamic Host Configuration Protocol) is used to automatically assign IP addresses to computers in a network.

DORA is the 4-step process DHCP uses.

👉 **DORA** = Discover, Offer, Request, Acknowledge

- **Steps of DORA Process**

D – Discover

When a new device (client) joins a network, it does not have an IP.

It sends a broadcast message saying: “Is there any DHCP server? I need an IP!”

O – Offer

The DHCP server replies with an Offer message.

It suggests an available IP address (like 192.168.1.10) and other details (subnet mask, gateway).

R – Request

The client receives the offer and sends back a Request message.

It says: "I would like to use this IP, please confirm."

A – Acknowledge

The DHCP server confirms by sending an Acknowledge message.

5. List the protocols which are used in email. Explain mail access/receiving protocols with diagram.

- **Protocols Used in Email**

1. **SMTP (Simple Mail Transfer Protocol)** – used to **send emails** from client to server or server to server.
2. **POP3 (Post Office Protocol version 3)** – used to **receive emails** and download them to your computer.
3. **IMAP (Internet Message Access Protocol)** – used to **receive emails** but keeps them on the server so you can access from multiple devices.

- **Mail Access / Receiving Protocols**

1. POP3 (Post Office Protocol 3)

- Downloads emails from the mail server to your computer.
- **Emails are removed from the server** after download (unless set to leave a copy).
- Works well if you **check emails from one device only**.
- **Port**: 110 (non-secure), 995 (secure/SSL).

2. IMAP (Internet Message Access Protocol)

- Accesses emails **directly on the mail server** without downloading them permanently.
- You can **read emails from multiple devices** (PC, phone, tablet).
- **Port**: 143 (non-secure), 993 (secure/SSL).

◆ **How Email Works (Simplified Diagram)**

[Sender Computer]

SMTP (send)

[Mail Server of Sender]

SMTP (transfer)

[Mail Server of Receiver]

POP3 / IMAP (receive)

[Receiver Computer / Device]

6. What is importance of application layer in computer network? Justify.

◆ **Importance of Application Layer**

- The **Application Layer** is the **topmost layer (Layer 7) of the OSI model**.
- It is **closest to the user**, meaning it directly interacts with software applications.
- Its main role is to **provide network services to end-users** so they can use the network easily.

◆ **Key Points / Importance:**

1. **User Interface** – Allows users to **send/receive data** without worrying about how the network works.

2. **Service Provision** – Supports services like **email, file transfer, web browsing, remote login**, etc.
3. **Data Formatting & Translation** – Ensures data is in a format that applications can understand.
4. **Communication Management** – Coordinates **end-to-end communication** between applications on different devices.

🔹 **Example:**

- When you **browse a website**, the web browser communicates through the **Application Layer** using **HTTP/HTTPS** protocols.
- When you **send an email**, it uses **SMTP** at the Application Layer.

✅ **Justification**

Without the Application Layer, users **cannot directly use network services**. It **connects software applications to the network**, making communication and data exchange simple and user-friendly.

7. State the port of below: 1) DNS, 2) SMTP, 3) HTTPS.

Protocol	Port Number	Purpose
DNS	53	Resolves domain names to IP addresses
SMTP	25	Sends emails from client to server or between servers

Protocol	Port Number	Purpose
HTTPS	443	Secure web browsing (HTTP over SSL/TLS)

8. Discuss how email works using SMTP protocol.

- **What is SMTP?**
  - **SMTP = Simple Mail Transfer Protocol**
  - Used to **send emails** from the sender to the receiver's mail server.
  - Works only for **sending, not receiving** emails.

- **How Email Works Using SMTP**

Step 1: Compose Email

- User writes an email on a **mail client** (like Gmail, Outlook).

Step 2: Send Email via SMTP

- The mail client uses **SMTP protocol** to send the email to the **sender's mail server**.
- SMTP checks the recipient's address to know where to deliver the email.

Step 3: Transfer Between Servers

- If the recipient's email is on a **different server**, SMTP forwards the email from the **sender's server to the recipient's server**.
- This may involve multiple servers until the email reaches the destination server.

Step 4: Email Delivery

- The recipient's mail server receives the email and stores it.

- The recipient can now access it using **POP3 or IMAP** (these are mail-receiving protocols).

9. Explain persistent HTTP with suitable example.

What is Persistent HTTP?

- **Persistent HTTP** is a type of HTTP connection where **one TCP connection** is used to send **multiple requests and responses** between client and server.
- Unlike **non-persistent HTTP**, it **does not open a new connection for every file**.
- It is **faster and efficient** because it reduces the overhead of opening and closing connections.

Key Features

1. Uses **one TCP connection** for multiple requests.
2. Reduces **network congestion** and **delay**.
3. Supported in **HTTP/1.1 and later versions**.

Example of Persistent HTTP

Suppose a webpage has:

- 1 HTML file
- 3 images
- 1 CSS file

Non-Persistent HTTP:

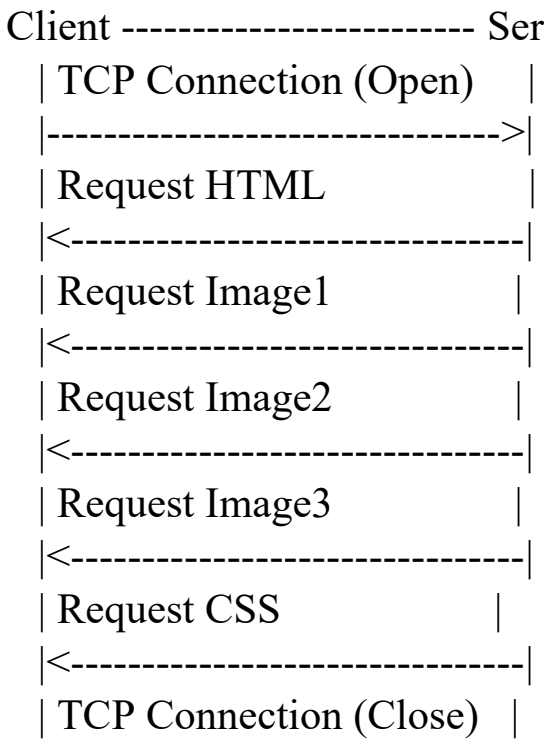
- Opens **5 separate TCP connections** (one for each file).
- Slower because of multiple connection setups.

Persistent HTTP:

- Opens **1 TCP connection**.
- Sends all 5 requests sequentially over the same connection.
- Faster and more efficient.

COMPUTER NETWORKS

- **Simple Diagram**



10. HTTP Message Format (Request-Response)

- **What is HTTP Message?**

  - HTTP (Hypertext Transfer Protocol) is used for **communication between a client (browser) and a server**.
  - Communication happens by **sending messages**:
    1. **HTTP Request** – From client to server
    2. **HTTP Response** – From server to client

- **1. HTTP Request Message**

  - **Purpose:** Client asks the server for a resource (like a web page or image).
  - **Format:**

    Request Line
    Headers
    (blank line)
    Optional Body

Components:

1. **Request Line** – Specifies:
    - **Method** (GET, POST, PUT, DELETE)
    - **URL** (resource path)
    - **HTTP version**
      Example:

      GET /index.html HTTP/1.1
2. **Headers** – Additional information about the request.
3. **Blank Line** – Marks the end of headers.
4. **Body (Optional)** – Contains data sent to the server (used in POST/PUT).

2. HTTP Response Message

Purpose: Server sends back the requested resource along with status info.
Format:

Status Line
Headers
(blank line)
Optional Body

Components:

1. **Status Line** – Contains:
    - HTTP version
    - **Status code** (e.g., 200, 404)
    - Reason phrase (like OK, Not Found)
      Example:

      HTTP/1.1 200 OK
2. **Headers** – Additional info about the response.
3. **Blank Line** – Marks the end of headers.

4. **Body (Optional)** – Contains the requested data (like HTML page, image, JSON).

11. Cookie

◆ What is a Cookie?

- A **cookie** is a **small piece of data** stored on your computer by a website.
- Websites use cookies to **remember information about you**.
- It helps improve your browsing experience.

◆ Key Points About Cookies

1. **Stored on Client Side** – Saved in your browser.
2. **Sent with Requests** – Whenever you visit the same website, the browser sends the cookie back to the server.
3. **Helps Websites Remember:**
    - Login details (so you don't have to log in every time)
    - Preferences (like theme, language)
    - Shopping cart items in online stores

◆ Example

- You log in to **Gmail**.
- Gmail stores a cookie on your browser to **remember that you are logged in**.
- Next time you open Gmail, you don't need to enter your password again.

◆ Types of Cookies

1. **Session Cookie** – Temporary, deleted when you close the browser.
2. **Persistent Cookie** – Remains on your device even after closing the browser (e.g., remember login).

3. **Secure Cookie** – Can only be sent over secure HTTPS connection.

12. Proxy Server (Web Caches)

- **What is a Proxy Server?**
  - A **proxy server** is an **intermediate computer/server** that sits **between a client (your computer) and the Internet**.
  - All requests from the client go **through the proxy** before reaching the Internet.
  - Similarly, the server's response passes through the proxy before reaching the client.

- **Functions / Uses of a Proxy Server**

1. **Security & Privacy** – Hides your **IP address** from the websites you visit.
2. **Control Internet Access** – Can **block certain websites** in schools or offices.
3. **Caching** – Stores copies of frequently accessed web pages to **load them faster**.
4. **Filter Content** – Can block harmful or unwanted content.
5. **Anonymity** – Makes your online activity **harder to trace**.

- **How It Works (Simple Flow)**

Client (You) --> Proxy Server --> Internet / Web Server

(Filters, Caches, Security)

Step by Step:

1. You request a website.

2. Request goes to **proxy server**.
3. Proxy checks cache / applies rules.
4. Proxy forwards the request to the **web server**.
5. Response comes back to **proxy**, which then sends it to you.

- **Example**
  - In a school, a **proxy server** can **block YouTube** but allow educational websites.
  - In offices, proxies can **cache pages** to save bandwidth and speed up browsing.

13. TCP vs UDP

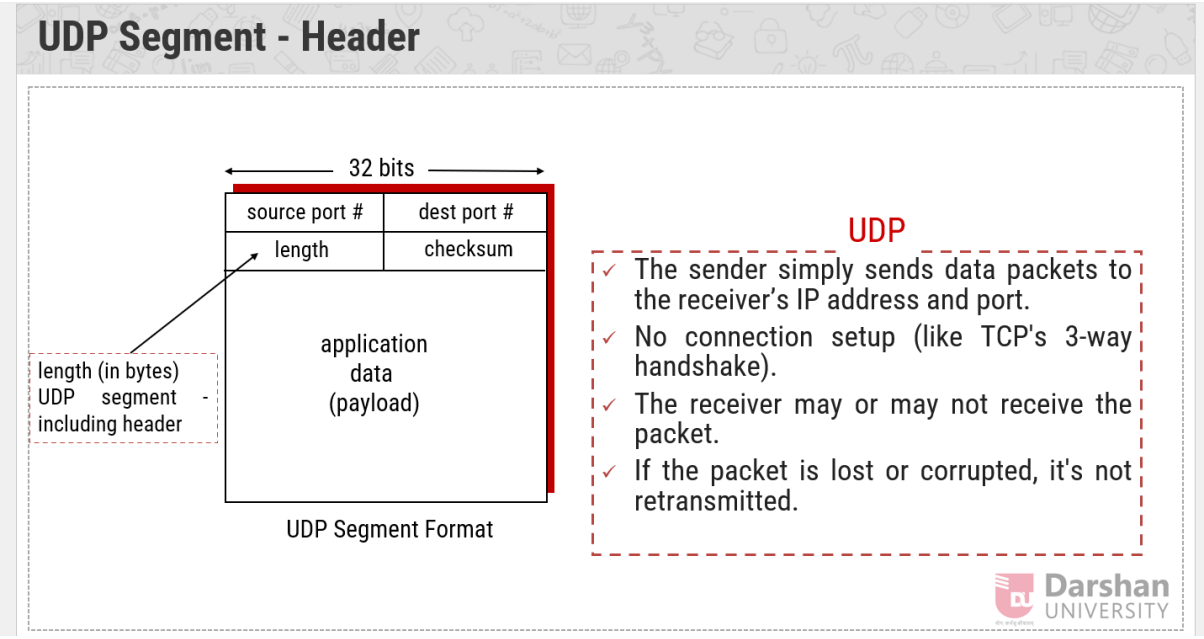
- **What are TCP and UDP?**
  - Both are **transport layer protocols** used to send data over the Internet.
  - **TCP = Transmission Control Protocol** → reliable, connection-oriented.
  - **UDP = User Datagram Protocol** → fast, connectionless, less reliable.

Feature	TCP	UDP
Full Form	Transmission Control Protocol	User Datagram Protocol
Connection	Connection-oriented (establishes a connection first)	Connectionless (no connection needed)
Reliability	Reliable, ensures data reaches safely	Unreliable, may lose data

Feature	TCP	UDP
Data Order	Data arrives in **correct order**	Data may arrive **out of order**
Speed	Slower due to error checking	Faster, less overhead
Error Checking	Yes, retransmits lost data	Yes, but no retransmission
Use Case / Example	Web browsing (HTTP/HTTPS), email, file transfer	Online gaming, video streaming, VoIP, live broadcast
Flow Control	Yes	No
Header Size	Larger (20 bytes)	Smaller (8 bytes)

Unit-3

1. UDP Header Format



- **Explanation of Each Field**

1. Source Port (16 bits)

Port number of the sending application.

Example: DNS uses port 53, a client may use a random port.

2. Destination Port (16 bits)

Port number of the receiving application.

Example: 53 for DNS server, 67 for DHCP server.

3. Length (16 bits)

Total length of UDP header + data (in bytes).

Minimum length = 8 bytes (header only).

4. Checksum (16 bits)

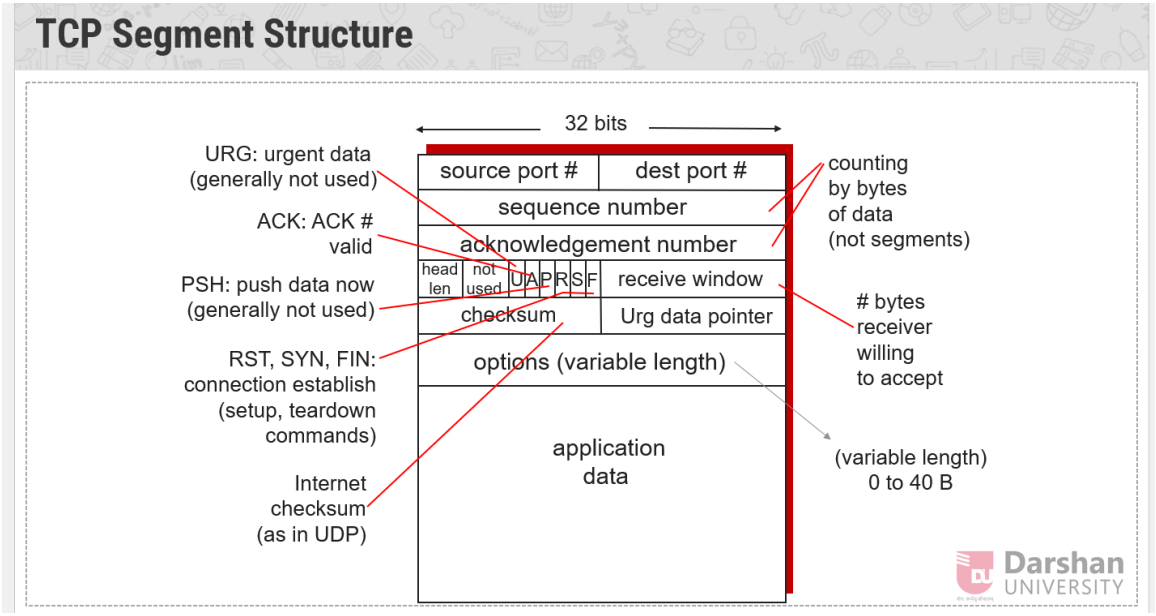
Used for error detection of header and data.

If data is corrupted, receiver can detect it.

5. Data (Payload)

Actual application data being sent (e.g., DNS query, video stream, gaming packets).

2. TCP Header Format



- **Explanation of Each Field**

1. **Source Port (16 bits)**
    - Port number of the **sending application**.
    - Example: 80 for HTTP, 25 for SMTP.
2. **Destination Port (16 bits)**
    - Port number of the **receiving application**.
    - Example: 443 for HTTPS.
3. **Sequence Number (32 bits)**
    - Identifies the **position of the first byte** in this segment.
    - Ensures data arrives in correct order.
4. **Acknowledgment Number (32 bits)**
    - If **ACK flag** is set, this field tells the sender the **next expected byte**.
    - Used for **reliability**.
5. **Data Offset (4 bits)**
    - Tells the **length of TCP header** (in 32-bit words).
    - Helps receiver know where data begins.
6. **Reserved (6 bits)**
    - Reserved for future use.
    - Always set to 0.
7. **Flags / Control Bits (6 bits)**
    - Control connection and data transfer.
        - **URG** → Urgent data
        - **ACK** → Acknowledgment valid

- **PSH** → Push data immediately
- **RST** → Reset connection
- **SYN** → Start connection
- **FIN** → Finish/close connection

8. **Window Size (16 bits)**
   - Tells how much data (in bytes) the receiver can accept.
   - Used for **flow control**.
9. **Checksum (16 bits)**
   - Error-checking for **header and data**.
   - Ensures data integrity.
10. **Urgent Pointer (16 bits)**
    - Points to urgent data in the segment.
    - Used when **URG flag** is set.
11. **Options (Variable)**
    - Extra settings like **Maximum Segment Size (MSS)**, window scaling, etc.
12. **Padding**
    - Ensures the header size is a multiple of **32 bits**.
13. **Data (Payload)**
    - The actual data being sent (e.g., part of a web page, email, file).

COMPUTER NETWORKS

3. Compare connection-oriented and connection less protocol.

Feature	Connection-Oriented Protocol	Connectionless Protocol
Definition	A connection is first **established** between sender & receiver before data transfer.	Data is sent **without establishing a connection**.
Reliability	Reliable – ensures data reaches safely and in correct order.	Not reliable – data may be lost, duplicated, or arrive out of order.
Acknowledgment	Acknowledgment is used (receiver confirms receipt).	No acknowledgment of received data.
Overhead	More overhead (extra steps like connection setup, sequencing, error-checking).	Less overhead (just sends data directly).
Speed	Slower due to reliability mechanisms.	Faster because no setup or acknowledgment.
Example Protocols	TCP (Transmission Control Protocol).	UDP (User Datagram Protocol).
Use Cases	Web browsing, file transfer, emails.	Video streaming, gaming, voice calls (VoIP).

4. What is the main reason to use sliding window protocol? Draw and discuss sliding window protocol.

- **What is Sliding Window Protocol?**

COMPUTER NETWORKS

- Sliding Window Protocol is a **flow control method** in computer networks.
- It ensures that the **sender does not overwhelm the receiver** with too much data.
- It also makes transmission **efficient** by allowing multiple packets to be sent before waiting for acknowledgment.

🔹 **Main Reason to Use Sliding Window Protocol**

👉 The main reason is to **improve efficiency and reliability** in data transmission by:

1. **Avoiding congestion** (sender does not send more data than receiver can handle).
2. **Sending multiple frames without waiting** for each acknowledgment (better performance).
3. **Ensuring reliable delivery** (lost packets are detected and retransmitted).

🔹 Sliding Window Protocol Diagram

Sender Side (Window of 4 frames) Receiver Side

```
-------------------------------------------------------------
| [1] [2] [3] [4] | 5  6  7  8 ...        <-- Frames
-------------------------------------------------------------
```

↑ Window moves (slides) forward as ACKs are received

5. Explain the process of connection-establishment and connection release in terms of TCP.

🔹 **1. TCP Connection Establishment (3-Way Handshake)**

TCP uses a **3-way handshake** to establish a reliable connection between client and server.

Steps:

1. **SYN (synchronize):**
    - Client sends a **SYN** packet to the server (request to start connection).
2. **SYN + ACK (synchronize + acknowledgment):**
    - Server replies with **SYN + ACK** (acknowledges client request and sends its own request).
3. **ACK (acknowledgment):**
    - Client sends an **ACK** back to the server.
    - Connection is now established → both can start sending data.

Diagram:

Client Server

| -------- SYN ----------> |

| <----- SYN + ACK ---------- |

| -------- ACK ----------> |

Connection Established

- **2. TCP Connection Release (4-Way Handshake)**

To close a TCP connection, a **4-step process** (called **4-way handshake**) is used.

Steps:

1. **FIN (finish):**

- Client sends **FIN** to server → "I have no more data to send."

2. **ACK:**
   - Server acknowledges the FIN with **ACK**.
   - (Server may still send remaining data if needed.)
3. **FIN:**
   - Server sends its own **FIN** → "I am done too."
4. **ACK:**
   - Client sends final **ACK**.
   - Connection is now closed.

Diagram:

```
Client                    Server
  | -------- FIN ----------> |
  | <------- ACK ---------- |
  | <------- FIN ---------- |
  | -------- ACK ----------> |
     Connection Closed
```

6. Pipelined Protocol

◆ What is a Pipelined Protocol?

- Instead of sending **one packet at a time**, the sender sends **multiple packets (frames)** before waiting for acknowledgment.
- This improves **efficiency** and keeps the channel busy.
- Two main types: **Go-Back-N** and **Selective Repeat**.

1️⃣ Go-Back-N ARQ

Working:

- Sender can send up to **N frames** before needing an acknowledgment.
- If one frame is lost/damaged, the receiver **discards it and all following frames**.
- Sender must **go back** and retransmit that frame and all after it.

Example:

- Sender sends frames 1, 2, 3, 4.
- Suppose frame 2 is lost.
- Receiver discards 3 and 4 (since 2 was missing).
- Sender retransmits **2, 3, 4** again.

✅ **Advantage:** Simple to implement.
❌ **Disadvantage:** Wastes bandwidth (retransmits many frames even if only one was lost).

2️⃣ Selective Repeat ARQ

Working:

- Sender still sends multiple frames.
- If a frame is lost/damaged, **only that frame is retransmitted** (not all after it).
- Receiver stores out-of-order frames in a buffer until the missing one arrives.

Example:

- Sender sends frames 1, 2, 3, 4.
- Suppose frame 2 is lost.
- Receiver accepts 1, 3, 4 and buffers them.
- Sender retransmits only frame 2.
- Receiver reorders them as 1, 2, 3, 4.

✅ **Advantage:** Saves bandwidth (retransmits only the lost frame).
❌ **Disadvantage:** More complex (needs buffer and reordering).

COMPUTER NETWORKS

7. Discuss the concepts of Multiplexing and demultiplexing in transport layer with appropriate diagram.

1️⃣ Multiplexing

- **Meaning:** Combining data from many applications into one stream for transmission over the network.
- At the **sender side**, the transport layer collects messages from different applications (like browser, email, chat app) and sends them over the same network connection.
- It uses **port numbers** to identify which application the data belongs to.

👉 **Example:**
Your laptop can use **port 80 for web browsing**, **port 25 for email**, and **port 21 for FTP** → all share the same network connection.

2️⃣ Demultiplexing

- **Meaning:** Delivering the received data to the correct application process.
- At the **receiver side**, the transport layer uses **port numbers** to separate (demultiplex) the incoming data and pass it to the correct application.

👉 **Example:**
If a packet arrives with **port 25**, it is delivered to the **email application**; if it has **port 80**, it goes to the **browser**.

8. Explain checksum with any example.

🔹 What is a Checksum?

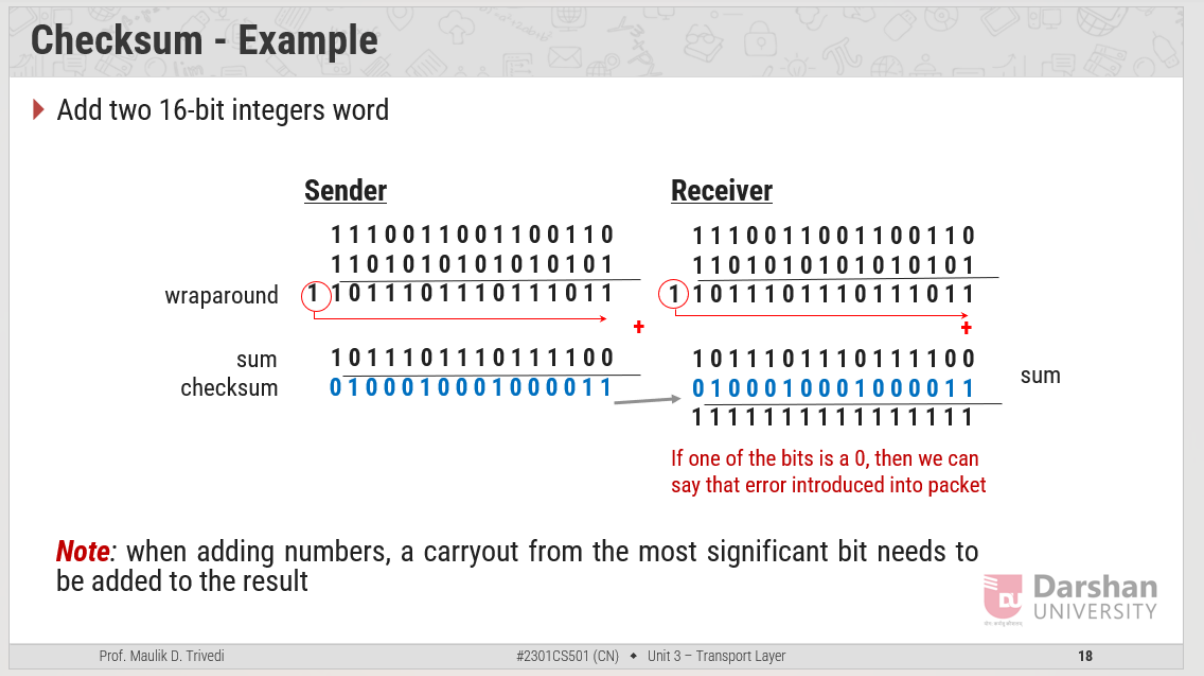
- A **checksum** is an error-detection method used in computer networks.

- It helps to check whether the data received is the **same as the data sent** (no corruption).
- Both **sender and receiver** calculate the checksum → if they match, the data is correct.

How It Works (Steps)

1. The sender adds up all the data segments (binary numbers).
2. The **sum is inverted** (1's complement) → this is the **checksum**.
3. The sender sends both **data + checksum**.
4. The receiver adds all received data (including checksum).
    - If the result is all **1's**, no error.
    - If not, an error occurred.

Example



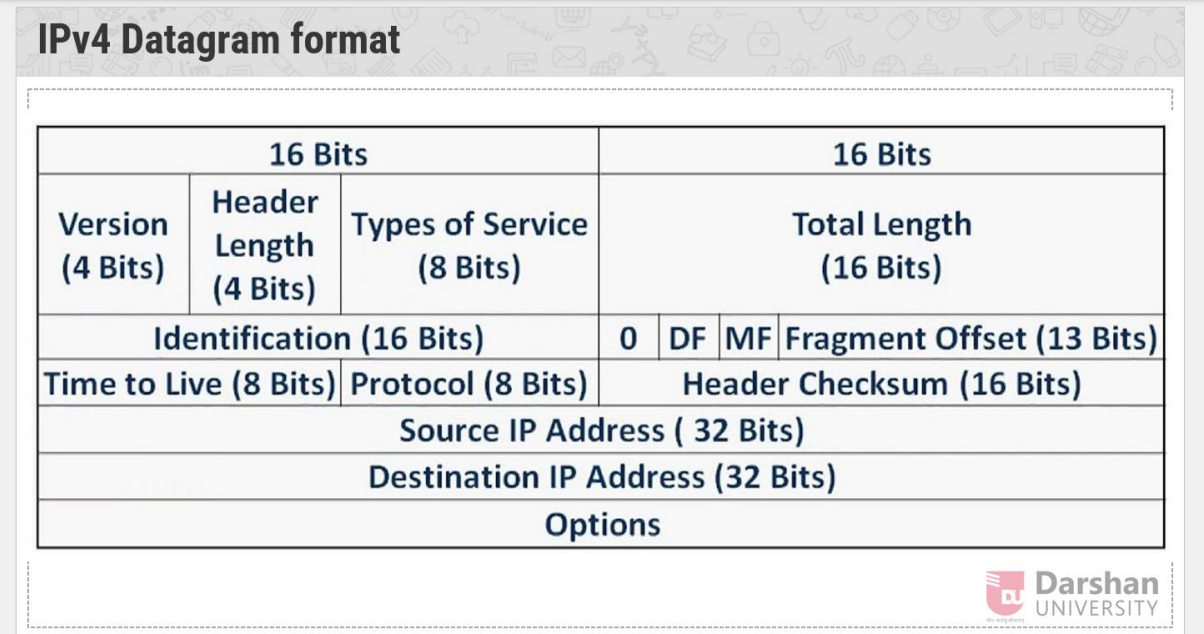
9. rdt 1.0, 2.0, 2.1, 2.2, 3.0

COMPUTER NETWORKS

Unit-4

1. Draw IPV4 header format and explain the functionality of each field of IPV4 header.

- **IPv4 Header Format (Diagram)**



- **Explanation of Each Field**

Version (4 Bits)

- Specifies the **IP version** being used.
- For IPv4, this value is **4**.

Header Length / IHL (4 Bits)

- Length of the **IP header** in 32-bit words.
- Minimum = 20 bytes, maximum = 60 bytes.

Type of Service (8 Bits)

- Indicates **priority and quality of service** for the packet.
- Can prioritize low delay, high throughput, or high reliability.

Total Length (16 Bits)

- Total length of the IP packet **including header + data**.

- Maximum size = 65,535 bytes.

Identification (16 Bits)

- Unique number for **each packet** to identify fragments of the same packet.

Flags (3 Bits)

- Control packet fragmentation:
  - **0** → Reserved
  - **DF** → Don't Fragment
  - **MF** → More Fragments

Fragment Offset (13 Bits)

- Position of this fragment in the **original packet**.
- Helps receiver **reassemble fragmented packets**.

Time to Live / TTL (8 Bits)

- Limits the lifetime of the packet.
- Decremented by 1 at each router.
- If TTL = 0 → packet is discarded (prevents looping).

Protocol (8 Bits)

- Indicates which **transport layer protocol** is used.
- Example: TCP (6), UDP (17), ICMP (1).

Header Checksum (16 Bits)

- Error-checking for **IP header only**.
- Ensures header integrity.

Source IP Address (32 Bits)

- IP address of the **sender** of the packet.

COMPUTER NETWORKS

Destination IP Address (32 Bits)

- IP address of the **receiver** of the packet.

Options (Variable)

- Optional field for **special features**, like security, record route, timestamp.

2. Make a list of IP address class with its range. What are the default subnet mask of class A, B & C. Draw and explain network id and host id in class A, B & C.

📌 IP Address Classes

An **IP address** is 32 bits (4 bytes) divided into **Network ID** + **Host ID**. It is classified into 5 classes: **A, B, C, D, E**.

🔹 1. Classes and Ranges

Class	Range	First Octet Range	Usage
A	0.0.0.0 to 127.255.255.255	0 – 127	Very large networks
B	128.0.0.0 to 191.255.255.255	128 – 191	Medium networks
C	192.0.0.0 to 223.255.255.255	192 – 223	Small networks
D	224.0.0.0 to 239.255.255.255	224 – 239	Multicasting
E	240.0.0.0 to 255.255.255.255	240 – 255	Research/Experimental

🔹 **2. Default Subnet Masks**

- Class A → 255.0.0.0

- Class B → 255.255.0.0
- Class C → 255.255.255.0

🔹 **3. Network ID & Host ID**

👉 Each IP address is divided into **Network ID** (identifies the network) and **Host ID** (identifies device within that network).

(i) Class A

- **1st Octet** = Network ID
- **Remaining 3 Octets** = Host ID
- Example: **10.0.0.1** → Network = 10, Host = 0.0.1

| Network ID (8 bits) | Host ID (24 bits) |

(ii) Class B

- **First 2 Octets** = Network ID
- **Last 2 Octets** = Host ID
- Example: **172.16.5.1** → Network = 172.16, Host = 5.1

| Network ID (16 bits) | Host ID (16 bits) |

(iii) Class C

- **First 3 Octets** = Network ID
- **Last Octet** = Host ID
- Example: **192.168.1.5** → Network = 192.168.1, Host = 5

| Network ID (24 bits) | Host ID (8 bits) |

3.Distinguish between IPV4 address and IPV6 address.

Feature	IPv4	IPv6
Address Size	32-bit (4 bytes)	128-bit (16 bytes)

COMPUTER NETWORKS

Feature	IPv4	IPv6
Format	Written as **4 decimal numbers** separated by dots (e.g., 192.168.1.1)	Written as **8 groups of hexadecimal numbers** separated by colons (e.g., 2001:0db8:85a3::8a2e:0370:7334)
Number of Addresses	About **4.3 billion**	About **3.4×10^{38}** (almost unlimited)
Header Size	20–60 bytes (variable)	40 bytes (fixed)
Security	Security depends on applications (no built-in)	Built-in security with **IPSec**
Configuration	Manual or with DHCP	Supports **Auto-configuration**
Broadcast	Supports **broadcast**	No broadcast, only **multicast & anycast**
Routing	Larger routing tables	Simplified and faster routing

Feature	IPv4 (less efficient)	IPv6
Address Example	192.168.1.1	2001:0db8:0000:0000:0000:ff00:0042:8329

4.Explain Routing Information Protocol with appropriate diagram.

- **What is RIP?**
  - **RIP** is one of the oldest **distance vector routing protocols**.
  - It is used by routers to **exchange routing information** inside a small or medium-sized network.
  - It uses the **hop count** (number of routers a packet passes through) as the metric to find the best path.

- **Key Points of RIP**
  1. **Type:** Distance Vector Routing Protocol.
  2. **Metric Used:** Hop Count (max = 15 hops, more than 15 = unreachable).
  3. **Updates:** Routers send their **entire routing table** to neighboring routers every **30 seconds**.
  4. **Version:**
     - RIP v1 → Classful (does not support subnet info).
     - RIP v2 → Classless (supports subnetting, authentication, multicast).
  5. **Protocol Used:** UDP, Port Number **520**.

How RIP Works (Steps)

1. Each router maintains a **routing table** with destination networks and hop counts.
2. Routers **exchange tables** with neighbors every 30 seconds.
3. A router updates its own table if it finds a **shorter path** from a neighbor.
4. If no update is received within 180 seconds, the route is considered **down**.

RIP Diagram (Easy Example)

[Router A] ---- [Router B] ---- [Router C]

| | |

Net 1 Net 2 Net 3

Router A knows only **Net 1**.

Router B knows **Net 2**.

Router C knows **Net 3**.

Using RIP, they exchange routing tables → soon all routers know paths to **Net 1, Net 2, and Net 3**.

Advantages of RIP

- Simple and easy to configure.
- Works well in small networks.

Disadvantages of RIP

- Slow convergence (takes time to update).
- Not suitable for large networks (hop count limited to 15).

COMPUTER NETWORKS

5. Discuss Link state routing protocol with proper diagram.

◇ What is it?

- A **dynamic routing protocol** where each router knows the **complete topology** of the network.
- Instead of sending the whole routing table (like RIP), it sends **Link State Advertisements (LSAs)** to tell neighbors about its directly connected links.
- Then each router uses **Dijkstra's Shortest Path Algorithm** to calculate the **best route** to every destination.

◇ Steps in Link State Routing

1. **Neighbor Discovery**
    - Each router discovers its **directly connected neighbors**.
2. **Link State Advertisement (LSA)**
    - Each router creates an LSA containing information about its neighbors and link costs.
3. **Flooding of LSAs**
    - LSAs are sent (flooded) to **all routers in the network**.
4. **Building Link State Database**
    - Every router stores received LSAs in a database → this gives a **full map of the network**.
5. **Shortest Path Calculation**
    - Each router runs **Dijkstra's algorithm** on the database → builds its routing table with shortest paths.

◇ Features

- **Protocol Examples:** OSPF (Open Shortest Path First), IS-IS.
- **Metric Used:** Cost (can be bandwidth, delay, etc.).
- **Faster convergence** than RIP.
- **Scalable** for large networks.

◇ **Diagram (Simple Example)**

```
[R1] ------ [R2]
 |    \     /   |
 |     \   /    |
[R3]---[R4]---[R5]
```

- Each router (R1–R5) sends LSAs about its connected links.
- Every router receives the same network map.
- Each runs Dijkstra's algorithm → finds the shortest paths.

◇ **Advantages**

- Faster convergence.
- Works well for **large and complex networks**.
- Provides **loop-free** routing.

◇ **Disadvantages**

- More **complex** than RIP.
- Requires **more memory and CPU power**.

6. Define subnetting and find total subnet of IP address 192.168.0.1/26.

◆ **Definition**

COMPUTER NETWORKS

Subnetting is the process of **dividing a large network (IP address range)** into **smaller networks (subnets)**.

- 192.168.0.1 → This is a **Class C** IP address.
- Default subnet mask for Class C = **255.255.255.0 (/24)**.
- Default = /24 → 24 network bits.
- Given = /26 → 26 network bits.
- Extra borrowed bits = 26 − 24 = 2 bits.

Number of Subnets=2^Borrowed Bits

Here , Subnets=2^2=4

Hosts per Subnet=2^Host Bits−2

Total bits = 32

Network bits = 26

Host bits = 32 − 26 = 6

$$\text{Hosts} = 2^6 - 2 = 64 - 2 = 62$$

7. Find the total number of host IP address 192.168.0.1/26.

Total bits in IPv4 = 32

Prefix length = /26 → means **26 bits are for network**

Remaining = 32 − 26 = **6 bits for host**

Total Host IPs=2^Host Bits

Here, Total Host IPs=2^6=64

Usable Host Addresses = 64−2=62

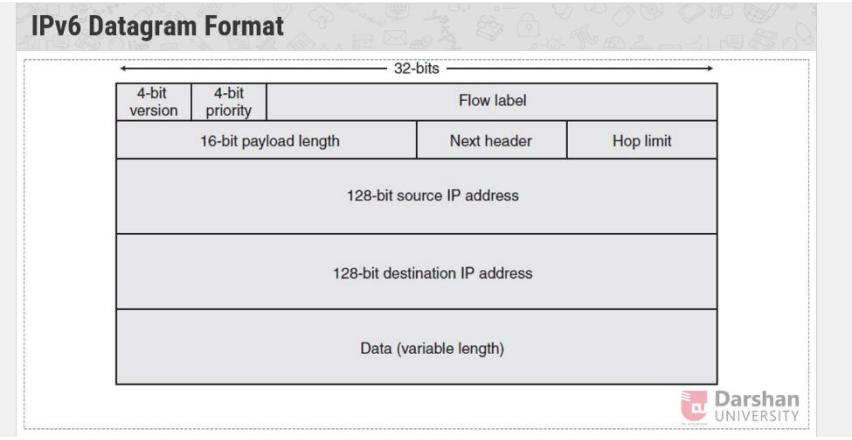
8.Compare datagram network and virtual circuit network.

Feature	Datagram Network	Virtual Circuit Network
Connection	**Connectionless** (no setup before sending data)	**Connection-oriented** (a path is set up before data transfer)

COMPUTER NETWORKS

Feature	Datagram Network	Virtual Circuit Network
Path	Each packet may take a **different path**	All packets follow the **same path**
Addressing	Each packet carries the **full destination address**	Only the **VCI (Virtual Circuit Identifier)** is carried (short address)
Reliability	No guarantee of delivery, order, or reliability (like postal system)	Packets arrive in order and more reliable (like a telephone call)
Delay	Less setup delay, but each router must make an independent decision	Initial setup delay, but forwarding is faster after setup
Resource Reservation	No resource reservation → best-effort delivery	Resources (like bandwidth) may be reserved
Example Protocols	IP (Internet Protocol) → used in the Internet	ATM, Frame Relay, MPLS
Analogy	Like sending letters (each letter can take a different route)	Like a phone call (path is set up first, then communication happens)

9.IPV6 datagram format



COMPUTER NETWORKS

10. DHCP

◇ What is DHCP (Dynamic Host Configuration Protocol)?

- **DHCP** is a **network protocol** that automatically provides IP addresses and other network settings (like subnet mask, gateway, DNS) to computers.
- Without DHCP, you would have to **manually assign IP addresses** to each device.

◇ Why do we need DHCP?

1. Avoids manual IP configuration.
2. Prevents IP conflicts.
3. Makes network management easier.
4. Useful in large networks (like offices, universities, Wi-Fi routers).

◇ How DHCP Works (DORA Process)

DHCP uses a **4-step process** (called **DORA**):

1. **Discover** – The client broadcasts a message: *"I need an IP!"*
2. **Offer** – DHCP server replies with an available IP address and settings.
3. **Request** – Client requests to use that offered IP.
4. **Acknowledge** – Server confirms and assigns the IP to the client.

   👉 After this, the client can use that IP address to communicate on the network.

◇ Example in Real Life

- When you connect your phone/laptop to **Wi-Fi**, DHCP automatically assigns an IP address, gateway, and DNS.
- That's why you don't need to manually set network settings.

COMPUTER NETWORKS

◇ Diagram (Easy Concept)

[Client] --Discover--> [DHCP Server]

[Client] <--Offer----- [DHCP Server]

[Client] --Request---> [DHCP Server]

[Client] <--Acknowledge [DHCP Server]

11. NAT

◇ What is NAT (Network Address Translation)?

- **NAT** is a method used in networks to **convert private IP addresses into public IP addresses** (and vice versa).
- It is mostly used in **routers** to allow multiple devices in a private network (like your home Wi-Fi) to access the Internet using **one public IP address**.

◇ Why do we need NAT?

1. **IP Address Saving** – Public IPv4 addresses are limited, NAT allows many devices to share one.
2. **Security** – Hides internal IP addresses from the outside world.
3. **Flexibility** – Devices can use private IP ranges (like 192.168.x.x, 10.x.x.x) without conflict.

◇ Types of NAT

1. **Static NAT**
    - One private IP ↔ One public IP (1-to-1 mapping).
    - Example: Server hosting.
2. **Dynamic NAT**

- Many private IPs ↔ Pool of public IPs (mapping changes).

3. **PAT (Port Address Translation) / NAT Overloading**
   - Many private IPs ↔ **One public IP** (different port numbers used).
   - Example: Home Wi-Fi, office LAN.

◇ **Diagram**

Private Network (192.168.1.x) Internet (Public IP)

[Laptop: 192.168.1.2] \

[Mobile: 192.168.1.3] ----> [Router with NAT] ----> [Public IP: 117.200.45.10]

[Printer:192.168.1.4] /

Unit-5

1.Discuss the concept of variable size framing in terms of character oriented and Bit oriented with example.

📌 **Framing in Data Link Layer**

- **Framing** means dividing data into smaller units called **frames** for transmission.
- In **variable-size framing**, frames do not have a fixed length → instead, **special markers/flags** are used to indicate where a frame starts and ends.
- Two main methods:
  1. **Character-Oriented Framing(Byte)**
  2. **Bit-Oriented Framing(Bit)**

◇ **1. Character-Oriented Framing(Byte stuffing)**

- Uses **ASCII characters (bytes)** as special markers.
- **Start of Text (STX)** and **End of Text (ETX)** are added to mark beginning and end of a frame.

- If data itself contains these special characters, an **escape (ESC)** character is inserted (called *byte stuffing*).

👉 **Example:**

[STX] DATA1 DATA2 DATA3 [ETX]

If data contains ETX, then:

[STX] DATA1 ESC+ETX DATA3 [ETX]

Used in: Older protocols like **BISYNC**.

◇ **2. Bit-Oriented Framing (Bit stuffing)**

- Uses a **bit pattern** (not characters) as start and end markers.
- Commonly uses the **flag 01111110** to indicate frame boundaries.
- If the same flag pattern appears inside data, **bit stuffing** is used → after every 5 consecutive 1s, a **0 is inserted**.

👉 **Example:**

FLAG (01111110) + Data Bits + FLAG (01111110)

If data contains 01111110, it becomes:

0111110 0 (stuffed extra 0 after five 1s)

Used in: Modern protocols like **HDLC, PPP**.

2.Draw and discuss ethernet frame structure.

- Ethernet is a widely used **data link layer protocol**.
  When sending data, Ethernet wraps it in a **frame** that contains **control information + actual data**.

◇ Ethernet Frame Format (Standard IEEE 802.3)\

Preamble	SFD	Dest. MAC	Source MAC	Type/Length	Data	CRC/FCS
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46-1500 B	4 Bytes

◇ Explanation of Each Field

1. **Preamble (7 Bytes)**
    - A sequence of alternating 1s and 0s.
    - Purpose → Helps the receiver **synchronize the clock** before receiving actual data.
2. **SFD (Start Frame Delimiter) – 1 Byte**
    - Special pattern 10101011.
    - Marks the **end of preamble** and the **start of the actual frame**.
3. **Destination MAC Address (6 Bytes)**
    - The **physical address** of the receiver's network card.
    - Ensures data reaches the correct device.
4. **Source MAC Address (6 Bytes)**
    - The **physical address** of the sender's device.
    - Used by receiver to know who sent the data.
5. **Type/Length (2 Bytes)**
    - If value ≥ 1536 → Indicates the **protocol type** (e.g., IPv4 = 0x0800, ARP = 0x0806).
    - If value < 1500 → Indicates the **length of data** field.
6. **Data / Payload (46 to 1500 Bytes)**
    - Actual message (e.g., IP packet).
    - Minimum size = 46 bytes (padding is added if smaller).
    - Maximum size = 1500 bytes.
7. **FCS (Frame Check Sequence) – 4 Bytes**
    - Uses **CRC (Cyclic Redundancy Check)**.

- Helps receiver detect **errors** in transmission.

3.Write short note on random access collision sense protocol for collision detection and collision avoidance.

📌 **Random Access Collision Sense Protocols**

In computer networks (especially in LAN like Ethernet or Wi-Fi), many devices share the same channel to send data.
If two devices send data at the same time → **Collision** occurs.

To handle this, two important **Random Access Collision Sense Protocols** are used:

---

1. CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

- Mostly used in **wired LAN (Ethernet)**.
- **Working:**
  1. Before sending, device **listens** to the channel (Carrier Sense).
  2. If channel is **free**, device starts sending data.
  3. If another device also sends at the same time → **Collision happens**.
  4. Devices **detect collision** by monitoring the signal.
  5. On collision → both devices **stop transmission**, wait for a **random backoff time**, and then try again.

👉 This avoids wasting bandwidth after a collision by **detecting it early**.

---

2. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

- Mostly used in **wireless LAN (Wi-Fi)**.
- **Working:**
    1. Device **listens** to check if channel is free.
    2. If channel is free → device sends a **signal (RTS – Request to Send)** to inform others.
    3. Receiver replies with **CTS – Clear to Send**.
    4. Now device sends data safely.
    5. This way, **collision is avoided before transmission**.

👉 Since wireless cannot detect collisions easily (signals interfere), CSMA/CA focuses on **avoiding collisions**.

4.CRC sums

5. Explain parity check technique with any example.

Parity check is an **error detection method** used in computer networks and digital communication.
It helps to know if **data got corrupted** during transmission.

◇ **How it works?**

- A **parity bit** (extra bit) is added to the original data.
- This parity bit makes the number of 1s in the whole message **either even or odd**.

👉 Two types:

1. **Even Parity** → Total number of 1s should be even.
2. **Odd Parity** → Total number of 1s should be odd.

◇ Example 1 (Even Parity)

Suppose we want to send **data = 1010001**

- Count number of 1s → There are 3 ones (odd).
- For **even parity**, we need total 1s = even.
- So we add **parity bit = 1**.
- Final data sent = **10100011** (now total 4 ones → even).

At receiver:

- Receiver also counts number of 1s.
- If even → no error.
- If odd → error detected.

◇ Example 2 (Odd Parity)

Data = **110010**

- Count ones = 3 (odd).
- For **odd parity**, total should remain odd.
- So we add **parity bit = 0**.
- Final data sent = **1100100**.

◇ Limitation

- Parity check can only **detect single-bit errors**.
- If 2 bits change, error may go undetected.

6.Write a short note on FDMA (Frequency Division Multiple Access).

- **Definition:**
  FDMA is a channel access method where the **available frequency band** is divided into **multiple smaller frequency channels**, and each user is assigned a **separate frequency** to communicate.

◇ How it works?

- Total bandwidth → divided into **frequency slots**.
- Each user gets a **fixed frequency slot** for communication.
- Guard bands are kept between slots to avoid interference.

◇ Example

If we have **100 MHz bandwidth** and 10 users → each user may get **10 MHz** channel for communication.

◇ Advantages

- Simple and easy to implement.
- No collision since each user has a dedicated frequency.

◇ Disadvantages

- Bandwidth is wasted if a user is idle (cannot share unused slot).
- Needs guard bands → reduces efficiency.

7.Write a short note on CDMA (Code Division Multiple Access).

- **Definition:**
  CDMA is a multiple access method where **all users share the**

same frequency band at the same time, but each user is assigned a **unique code** to separate their data.

◇ How it works?

- Every user's signal is multiplied by a **unique spreading code** (a sequence of bits).
- At the receiver side, the same code is used to extract the intended signal.
- Other signals (with different codes) appear as noise and are ignored.

◇ Example

- Imagine a classroom where many people speak at once, but each pair uses a **different language**.
- You only understand the person speaking in your language (code).

◇ Advantages

- High bandwidth efficiency (many users can share same frequency).
- Provides better security and privacy (hard to intercept).
- Resistant to interference and noise.

◇ Disadvantages

- Complex to implement.

- Requires strict power control (strong signals can overpower weak ones → near–far problem).

8. Explain CRC technique with any example.

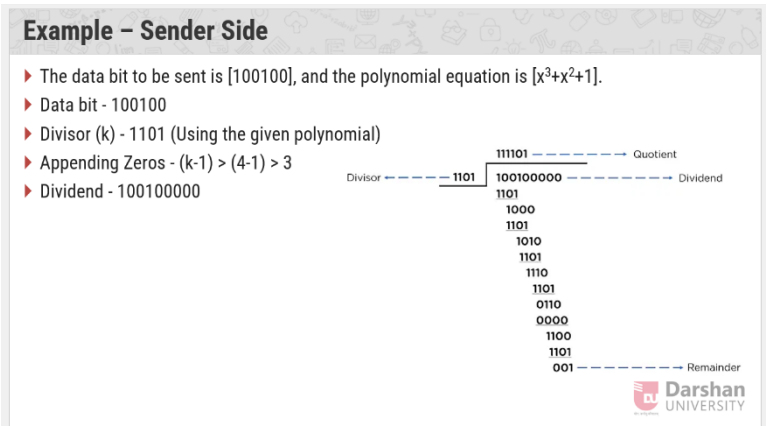
📌 **CRC (Cyclic Redundancy Check)**

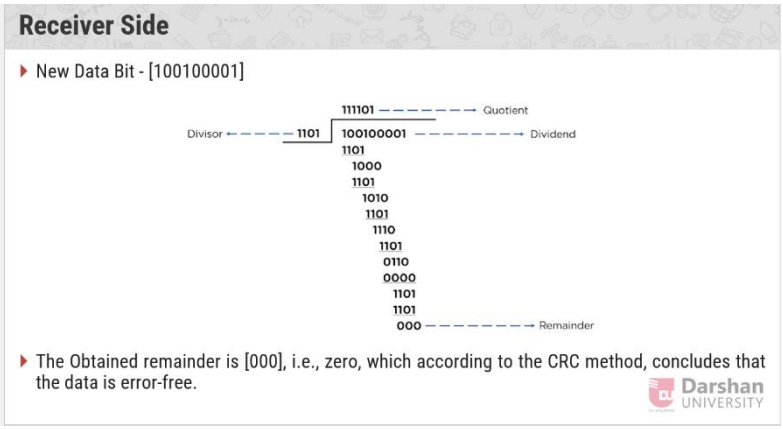
- CRC is an **error detection technique** used in computer networks and digital communication.
- It detects if the data got corrupted during transmission.
- It uses **binary division** (modulo-2 division) between data and a generator polynomial.

◇ **Steps in CRC**

1. **Sender Side:**
    - Take the **data bits** (message).
    - Append **r zeros** at the end (where r = degree of generator polynomial).
    - Divide this extended data by the **generator polynomial (key)** using modulo-2 division.
    - The remainder = **CRC bits**.
    - Append CRC bits to the original data → This full frame is sent.
2. **Receiver Side:**
    - Receiver divides the received frame (data + CRC) by the same generator polynomial.
    - If remainder = 0 → No error.
    - If remainder ≠ 0 → Error detected.

Example:





9. Describe how ARP works with suitable example.

📌 ARP (Address Resolution Protocol)

- **Purpose:** ARP is used to **find the MAC (physical) address** of a device when we already know its **IP address**.
- It works in a **local network (LAN)**.

◇ How ARP Works (Step by Step)

1. **Host wants to send data** → Suppose **Computer A** wants to send data to **Computer B**.
    - A knows **B's IP address**, but not its MAC address.
2. **Broadcast ARP Request:**
    - A sends an **ARP Request** message:
    - "Who has IP = 192.168.1.5? Tell me your MAC address."
    - This request is broadcast to all devices in the LAN.
3. **ARP Reply:**
    - Only the device with IP = 192.168.1.5 (Computer B) replies.
    - It sends back an **ARP Reply** directly to A:
    - "My MAC address is 00:1A:2B:3C:4D:5E."

4. **Store in ARP Table:**
    - Computer A stores this mapping (IP ↔ MAC) in its **ARP Cache/Table** for future use.
5. **Communication Begins:**
    - Now A can send data directly to B using B's MAC address.

◇ **Example**

- **Computer A (Sender):**
    - IP = 192.168.1.2, MAC = AA-AA-AA-AA-AA-AA
- **Computer B (Receiver):**
    - IP = 192.168.1.5, MAC = BB-BB-BB-BB-BB-BB

👉 A wants to send to B.

- A sends: *"Who has 192.168.1.5?"*
- B replies: *"I am 192.168.1.5, my MAC = BB-BB-BB-BB-BB-BB"*
- A stores this info in ARP cache.

Now A sends the data frame using **MAC = BB-BB-BB-BB-BB-BB**.

10. Pure, Slotted ALOHA

📌 **ALOHA Protocol**

ALOHA is a **random access protocol** used for sharing a communication channel among multiple users.
It was first used in early radio networks.

There are **two versions**: **Pure ALOHA** and **Slotted ALOHA**.

◇ **1. Pure ALOHA**

- **How it works:**

  - Whenever a station has data → it **sends immediately** (without waiting).
  - If two or more stations send at the same time → **collision** occurs.
  - Sender waits for acknowledgment (ACK).
  - If no ACK → sender assumes collision → waits for a random time → retransmits.
- **Efficiency:**
  - Very low, because collisions can happen at **any time**.
  - Maximum efficiency = **18.4%**.
- **Example:**
  - Computer A sends a frame at 2:01:05.
  - Computer B also sends a frame at 2:01:06.
  - Frames overlap → collision → both must retransmit.

◇ 2. Slotted ALOHA

- **Improvement over Pure ALOHA.**
- **How it works:**
  - Time is divided into **equal slots**.
  - A station can **only send at the beginning of a time slot**.
  - If two stations send in the same slot → collision happens.
  - But chance of collision is less than Pure ALOHA.
- **Efficiency:**
  - Much better than Pure ALOHA.
  - Maximum efficiency = **36.8%**.

- **Example:**
  - Computer A waits for slot 3 and sends.
  - Computer B also chooses slot 3 → collision.
  - But if B waits for slot 4, no collision.

11. Define TDMA with proper example.

📌 **TDMA (Time Division Multiple Access)**

- **Definition:**
  TDMA is a channel access method where the **available channel time** is divided into **time slots**, and each user is assigned a **specific time slot** to send or receive data.
- It is used in **digital cellular systems (2G GSM)**, satellite communication, etc.

◇ **How TDMA Works?**

1. The channel (frequency) is shared by multiple users.
2. Each user gets a **time slot** in a repeating cycle (frame).
3. Users transmit only during their slot; in other slots they stay silent.
4. Since slots are non-overlapping, **no collision** occurs.

◇ **Example**

- Suppose there are **3 users (A, B, C)** sharing one channel.
- The channel is divided into **3 time slots** per frame:
  Frame 1: | Slot 1 (A) | Slot 2 (B) | Slot 3 (C) |
  Frame 2: | Slot 1 (A) | Slot 2 (B) | Slot 3 (C) |
  Frame 3: | Slot 1 (A) | Slot 2 (B) | Slot 3 (C) |

👉 User A sends only in Slot 1, User B in Slot 2, and User C in Slot 3.
This repeats in every frame, so all get fair access.

◇ Advantages

- No collisions (each slot is fixed).
- Efficient use of channel.
- Works well for digital systems like GSM.

◇ Disadvantages

- Requires synchronization (users must know when their slot starts).
- If a user has no data, its slot is wasted.