Computer Networks

(LECTURE NOTES)

Prepared by:

Er. AKSHAY KUMAR PATRA

(Assistant Professor)

DEPT. OF COMPUTER SCIENCE. & ENGINEERING

Modern Engineering and Management Studies

Banaparia, Kuruda, Balasore, Odisha.

Module-1

Computer Networks

---

1. Introduction to Networks

What is a Network ?

A **network** is a group of two or more computing devices (like computers, servers, routers) connected to share resources, exchange data, and communicate. The internet is the largest network.

TYPES OF NETWORK:-

1.Personal Area Network (PAN)

A PAN connected device for personal use to sharing data only .Its range 0-10 m only

2.Local Area Network (LAN)

A LAN connect devices within a limited geographical area, such as a home, office building, or campus. Its range less then 150 m

3.Metropolitatan Area Network (MAN)

A MAN connects devices within a metropolitan area,such as city or town.Its range <50 m

4.Wide Area Network (WAN)

A WAN connects devices over a large geographical area ,such as a city or country. Its range not fix.

5.Wireless Area Network (WLAN)

Its connects devices without wire.Its connect process using radio wave/infrared signal.

6.Virtual Private Network (VPN)

It connects devices over the internet using encryption and tunneling protocols to secure data.

NETWORK TOPOLOGY:-

Network topology is the process in which multiple device are connected to each other over network. There are different types of Network topology.

1.Bus Topology

2.Star Topology

3.Ring Topology

4.Mesh Topology

5.Hybrid Topology

NETWORK DEVICES:-

1.Router

2.Switch

3.Hub

4.Modem

5.Network Interface Card (NIC)

NETWORK PROTOCALS:-

1.Transmission Control Protocal/Internet Protocol (TCP/IP)

2.Hypertext Transfer Protocal (HTTP)

3.File Transfer Protocal (FTP)

4.Domain Name System (DNS)

Network Hardware

These are physical devices required to build a network. Key components include:

- **Routers**: Direct data packets between networks.
- **Switches**: Connect multiple devices in a LAN and manage data flow.
- **Hubs**: Simple devices that connect multiple Ethernet devices; they broadcast data to all ports.
- **Modems**: Convert digital data to analog (and vice versa) for internet access.
- **Network Interface Cards (NICs)**: Allow computers to connect to a network.

Network Software

Network software enables devices to communicate and manage data flow. It includes:

- **Protocols**: Rules governing data communication (e.g., TCP/IP, HTTP).
- **Operating Systems**: Provide network services (e.g., Windows Server, Linux).
- **Network Management Software**: Tools to monitor and manage network traffic.

OSI Model

The **OSI Model** (Open Systems Interconnection) standardizes network functions into **7 layers**:

1. **Physical layer** – Transmission of raw bits over a medium.
2. **Data Link layer**– Frames, error control, MAC addressing.
3. **Network layer** – Routing, IP addressing.
4. **Transport layer** – Reliable data transfer (TCP/UDP).
5. **Session layer** – Manages sessions between applications.
6. **Presentation layer** – Data format translation, encryption.
7. **Application layer** – End-user services like email, FTP.

Physical Layer in Computer Network

The **Physical Layer** is the **first and lowest layer** in the **OSI (Open Systems Interconnection) model** of computer networks. It is responsible for the **transmission and reception of raw bit streams** over a physical medium.

Key Functions of the Physical Layer:

1. **Bit Transmission:**
   - Converts data into **electrical, optical, or radio signals**.
   - Transmits **0s and 1s** (binary data) between devices.
2. **Data Encoding:**
   - Defines how bits are represented on the physical medium (e.g., voltage levels, light pulses).
3. **Physical Media:**
   - Specifies the physical means used for transmission, such as:
     - Copper wires (Ethernet cables)
     - Fiber optics
     - Wireless (radio frequencies)
4. **Topology and Physical Design:**
   - Determines how devices are physically connected (e.g., star, bus, ring topologies).
5. **Data Rate (Bandwidth):**
   - Defines the rate at which data is transmitted, usually in **bits per second (bps)**.
6. **Synchronization:**
   - Ensures sender and receiver are synchronized in timing.
7. **Transmission Mode:**
   - **Simplex**: One-way communication
   - **Half-duplex**: Two-way communication, one at a time
   - **Full-duplex**: Two-way communication simultaneously

Devices Operating at Physical Layer:

- **Cables (Coaxial, Twisted Pair, Fiber)**
- **Hubs**
- **Repeaters**
- **Network Interface Cards (NICs)**

Example:

When you send a file over the internet, the physical layer is responsible for transmitting the binary bits of that file through wires or wirelessly from one device to another.

Summary:

Feature	Description
Layer in OSI Model	1st (Lowest)
Role	Transmission of raw bits
Medium	Cables, fiber optics, wireless
Devices	Hubs, repeaters, NICs
Concerned with	Hardware and signal transmission

Data Link Layer in Computer Networks

The **Data Link Layer** is the **second layer** in the **OSI (Open Systems Interconnection)** model, just above the **Physical Layer** and below the **Network Layer**.

Purpose of the Data Link Layer

The main job of the data link layer is to provide **reliable data transfer** across the **physical link** (wires, fiber, etc.) between two directly connected nodes.

It ensures that the **bits** received are the **same as the bits sent**, by handling errors, flow control, and framing.

Key Functions of the Data Link Layer

1. Framing

- Divides the stream of bits from the network layer into **manageable data units** called **frames**.
- A frame contains:
  - Source & Destination addresses (MAC addresses)
  - Control information
  - Payload (actual data)
  - Error-checking bits

2. Error Detection and Correction

- Uses techniques like **CRC (Cyclic Redundancy Check)** or **Parity Bits** to detect errors.
- If errors are found, frames can be **retransmitted** (based on protocol).

3. Flow Control

- Prevents the **sender** from overwhelming the **receiver**.
- Uses techniques like **stop-and-wait** and **sliding window protocols**.

4. Media Access Control (MAC)

- Determines **who can access the communication channel** and **when** (especially in shared media like Ethernet).
- Example protocols: **CSMA/CD (used in Ethernet)**, **CSMA/CA (used in Wi-Fi)**

5. Addressing

- Uses **MAC (Media Access Control)** addresses to identify devices on the same physical network.

Types of Data Link Layer Protocols

Protocol	Used In	Features
Ethernet	LANs	Fast, uses CSMA/CD
PPP (Point-to-Point Protocol)	Direct links like dial-up	Simple framing and authentication
HDLC (High-Level Data Link Control)	WAN links	Reliable communication
Wi-Fi (IEEE 802.11)	Wireless networks	Uses CSMA/CA for collision avoidance

Sub-Layers of the Data Link Layer

The data link layer is often divided into two sub-layers:

1. **LLC (Logical Link Control)**
   - Handles **error checking**, **flow control**, and **framing**.
   - Provides an interface to the network layer.
2. **MAC (Media Access Control)**
   - Controls how devices on the network **gain access** to the medium.
   - Deals with **hardware addressing** (MAC addresses).

Analogy for Better Understanding

Think of the data link layer as a **postal service**:

- **Framing** = Putting the message into an envelope.
- **MAC Address** = The recipient's house address.
- **Error Checking** = Making sure the envelope isn't damaged.
- **Flow Control** = Making sure the recipient can read one letter before the next arrives.
- **Access Control** = Making sure two postmen don’t deliver at the same time.

Why It's Important for B.Tech Students

- Forms the **foundation** for understanding **networking**.
- Helps in designing **protocols**, **switches**, and **error-resilient** systems.
- Crucial for subjects like **Computer Networks**, **Data Communication**, and **Network Security**.

Network Layer

The **Network Layer** is the **third layer** in the **OSI (Open Systems Interconnection) model**. It is responsible for **delivering data packets from the source to the destination**, even if they are on different networks.

Key Functions of the Network Layer

1. Logical Addressing

- Uses **IP addresses** to identify devices across different networks.
- Unlike MAC addresses (used in the Data Link Layer), IP addresses can change based on location.

2. Routing

- Determines the **best path** for data to travel across networks.
- Uses devices called **routers**.
- Common routing protocols: RIP, OSPF, BGP.

3. Packet Forwarding

- Moves packets from one router to the next toward the destination.
- Each router makes forwarding decisions based on the **routing table**.

4. Fragmentation and Reassembly

- Splits large packets into smaller ones to fit into the network's frame size.
- Reassembles them at the destination.

Key Protocols in the Network Layer

1. IP (Internet Protocol)

- Main protocol used.
- Two versions: IPv4 (32-bit) and IPv6 (128-bit).

2. ICMP (Internet Control Message Protocol)

- Used for diagnostics (e.g., `ping`, `traceroute`).

3. ARP (Address Resolution Protocol)

- Resolves IP addresses to MAC addresses (though it's more of a link-layer support protocol).

Devices Used at Network Layer

- **Router**: Main device that works at the Network Layer.
- **Layer 3 Switch**: Performs routing functions but faster and in hardware.

Real-Life Analogy

Imagine you're sending a letter:

- The **Network Layer** is like the **postal service**.
- It figures out **how** to get the letter to the recipient, even if they are in a different city or country.
- It doesn't care **what's inside** the envelope—just where it needs to go.

Summary

Feature	Description
Layer in OSI Model	3rd Layer
Key Responsibility	Routing, addressing, packet delivery
Main Protocol	IP (IPv4, IPv6)
Main Device	Router
Functionality	Path selection, packet forwarding, addressing

Transport Layer in Computer Networks

Transport Layer

The **Transport Layer** is the **4th layer** of the **OSI (Open Systems Interconnection)** model. Its main job is to **deliver data from one application to another**, across a network.

Think of it as a **delivery service**: the transport layer ensures that data from your computer reaches the correct application on another device **accurately**, **in order**, and **without duplication or loss**.

Key Responsibilities

1. **Process-to-Process Communication**
   - Identifies applications using **port numbers** (e.g., web browser uses port 80 for HTTP).
   - Ensures the right message goes to the right app.
2. **Segmentation and Reassembly**
   - Breaks large messages into smaller pieces (called **segments**).
   - Reassembles them at the receiver's end.
3. **Error Detection and Recovery**
   - Uses checksums to detect errors.
   - Can request retransmission if something goes wrong.
4. **Flow Control**
   - Ensures the sender doesn’t overwhelm the receiver.
   - Uses methods like **sliding window**.
5. **Congestion Control**
   - Prevents too much data from being sent through the network too fast, which can cause network congestion.
6. **Reliable vs. Unreliable Transmission**
   - Can provide **reliable** communication (with error checking, retransmissions) or **unreliable** communication (faster, but no guarantees).

Protocols at Transport Layer

Protocol	Reliability	Description
TCP (Transmission Control Protocol)	✅ Reliable	Connection-oriented. Guarantees delivery, order, and error-free transmission.
UDP (User Datagram Protocol)	❌ Unreliable	Connectionless. Faster, but doesn't guarantee delivery or order. Used in real-time apps like video calls, games.

□ Example

Suppose you are watching a YouTube video:

- **UDP** is used to stream the video because speed is more important than perfect accuracy. Now suppose you're logging into your bank:
- **TCP** is used to ensure no data is lost or delivered out of order.

□ Summary

The **Transport Layer** is crucial for:

- Making sure data gets to the right application
- Ensuring it’s delivered correctly and efficiently
- Providing both reliable (TCP) and fast (UDP) transmission options

Understanding this layer helps in building efficient, secure, and robust communication systems in networks

Session Layer

The **Session Layer** is the **5th layer** in the **OSI (Open Systems Interconnection)** model. It sits **above the Transport Layer** and **below the Presentation Layer**.

Its main role is to **establish, manage, and terminate sessions** between two communicating systems (like two computers on a network).

Key Functions of the Session Layer

1. **Session Establishment, Maintenance, and Termination**
    - It sets up a communication session between devices.
    - Keeps the session alive during data transfer.
    - Ends the session once the communication is complete.
2. **Synchronization**
    - Adds **checkpoints** (sync points) in data streams.
    - Useful for long transfers — if the transfer is interrupted, it can resume from the last checkpoint.
3. **Dialog Control**
    - Manages **two-way communication**.
    - Controls **who can send data** and **when**, preventing collisions (like turn-taking in a conversation).
    - Can operate in:
        - **Half-duplex mode** (one at a time),
        - **Full-duplex mode** (both at the same time).

Real-World Examples

- **Video conferencing**: Maintains a continuous session while users are connected.
- **Remote login (e.g., SSH, Telnet)**: Session layer helps keep the connection active.
- **Online banking**: Ensures your session is private and secure.

Protocols Used in Session Layer (Some overlap with other layers)

- **RPC (Remote Procedure Call)**
- **NetBIOS**
- **PPTP (Point-to-Point Tunneling Protocol)**

Note: In modern networks, the OSI model is mostly theoretical. Real-world networks often use the **TCP/IP model**, which doesn't have a separate session layer—its functions are handled by other layers (mainly the transport layer).

Summary

Feature	Description
Layer Number	5th Layer of OSI Model
Main Role	Manage sessions between applications
Key Functions	Session control, synchronization, dialog
Common Protocols	NetBIOS, RPC, PPTP

Presentation Layer in Computer Networks

The **Presentation Layer** is the **6th layer** of the **OSI (Open Systems Interconnection) model**. It acts as a **translator** between the **application layer (Layer 7)** and the **lower layers** of the network.

Main Function

The main job of the Presentation Layer is to ensure that the **data sent by the application layer of one system** is **readable by the application layer of another system**.

Think of it as the "**data translator**" of the OSI model.

Key Responsibilities of the Presentation Layer

Function	Description
Translation	Converts data between the formats used by the application and the network (e.g., from EBCDIC to ASCII).
Encryption & Decryption	Provides security by encrypting data before transmission and decrypting it at the receiver's end.
Compression & Decompression	Reduces the size of data to improve transmission speed and decompresses it upon reception.
Data Formatting	Ensures that data is in a format the receiving system can understand (e.g., JPEG, MP4, DOC).

Analogy

Imagine you're sending a letter from India to France. The **Presentation Layer** is like a translator who converts the letter from Hindi to French so that the recipient can understand it.

Real-World Examples

- **SSL/TLS (Secure Socket Layer / Transport Layer Security):** These use the Presentation Layer for **encryption and security** in HTTPS.
- **JPEG, MP3, GIF, MPEG:** These are file formats that the Presentation Layer may handle during **data formatting and compression**.

Summary

- The Presentation Layer ensures **data is presented correctly** between devices.
- It handles **data translation, encryption, decryption, and compression**.
- It makes communication between different systems **smooth and understandable**.

Application Layer in Computer Networks

What is the Application Layer?

The **Application Layer** is the **topmost layer** in the **OSI (Open Systems Interconnection)** model. It is responsible for **interacting with software applications** to provide **network services to end users**.

It is the layer **closest to the user** and **enables communication between software applications and lower layers of the network**.

Main Functions

1. **User Interface**: Allows users to interact with the network through applications like web browsers or email clients.
2. **Data Generation**: Collects data from the user to be sent over the network.
3. **Data Formatting and Encoding**: Ensures data is in a proper format for communication.
4. **Resource Sharing**: Supports services like file sharing and remote access.
5. **Network Virtual Terminal**: Allows users to log on to a remote host.
6. **Directory Services**: Provides distributed database sources and access for global information.

Common Protocols in the Application Layer

Protocol	Purpose
HTTP (Hypertext Transfer Protocol)	Used for accessing web pages.
FTP (File Transfer Protocol)	Used to transfer files between client and server.
SMTP (Simple Mail Transfer Protocol)	Sends emails.
POP3 / IMAP	Retrieves emails from servers.
DNS (Domain Name System)	Converts domain names to IP addresses.
Telnet	Remote login to another computer.

Real-life Examples

- **Web Browsing** → Uses **HTTP/HTTPS**
- **Email** → Uses **SMTP, POP3, IMAP**
- **File Transfer** → Uses **FTP**
- **Domain Name Lookup** → Uses **DNS**

Key Points to Remember

- It's the **7th layer** in the OSI model.
- It provides services **directly to users and applications**.
- It does **not** perform lower-level tasks like routing or encryption.
- It **relies on layers below** (like transport and network) to send and receive data.

Why is it Important?

Without the Application Layer, users wouldn't be able to **access network services** like the web, email, or file transfers in a structured and user-friendly way. It **bridges the gap** between human interaction and data transmission.

1.4 TCP/IP Reference Model

The **TCP/IP model**, which powers the Internet, has **4 layers**:

1. **Application** – High-level protocols like HTTP, FTP.
2. **Transport** – TCP/UDP, reliable data delivery.
3. **Internet** – IP addressing and routing.
4. **Network Interface** – Physical network communication.

OSI Layer	TCP/IP Equivalent
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data Link	Network Interface
Physical	

1.5 Example Networks

ARPANET

- Developed in the late 1960s by DARPA.
- First packet-switching network.
- Predecessor to the Internet.

Internet

- A global network connecting millions of devices.
- Uses TCP/IP protocols.
- Decentralized and scalable.

2.1 Data and Signals

Analog and Digital Signals

- **Analog Signals**: Continuous waves (e.g., sine wave).
- **Digital Signals**: Discrete levels (0s and 1s).

Periodic Analog Signals

- Characterized by:
  - **Amplitude** – Height of the wave.
  - **Frequency** – Cycles per second (Hz).
  - **Phase** – Shift of the wave.

Digital Signals

- Represent data using binary format (0s and 1s).
- Easier to process and less prone to noise.

2.2 Transmission Impairments

These are problems that affect signal transmission:

- **Attenuation** – Signal strength decreases over distance.
- **Noise** – Unwanted signals that interfere.
- **Distortion** – Signal changes shape due to different speeds of components.

2.3 Data Rate Limit

- The **maximum transmission speed** of a medium.
- Affected by:
  - Bandwidth
  - Noise level
  - Signal-to-noise ratio (SNR)

2.4 Guided Transmission Media

These media use a **physical path**:

Twisted Pair Cable

- Two insulated wires twisted together.
- Common in LANs.
- Types:
  - UTP (Unshielded)
  - STP (Shielded)

Coaxial Cable

- Single copper wire with shielding.
- Used in cable TV and broadband.

Fiber Optics

- Uses light to transmit data.
- High speed and long distance.
- Immune to electromagnetic interference.

2.5 Wireless Transmission (Unguided Media)

These media use **air or space**:

Radio Waves

- Omni-directional.
- Used in broadcasting and Wi-Fi.

Microwaves

- Unidirectional.
- Used in satellite and mobile communication.

Infrared

- Line-of-sight communication.
- Used in TV remotes.

Conclusion

Understanding network fundamentals is key for any B.Tech student in Computer Science or IT. The OSI and TCP/IP models provide a framework for understanding how data moves through networks, while transmission media and signal types form the basis of physical communication.

MODULE-2

Data Link Layer

The Data Link Layer is the second layer in the OSI model. It provides node-to-node data transfer—a link between two directly connected nodes. It also detects and possibly corrects errors that may occur in the Physical layer.

---

1. Design Issues in Data Link Layer

- **Framing**: Identifying the beginning and end of the data packet (frame).
- **Error Control**: Detecting and correcting errors.
- **Flow Control**: Preventing the sender from overwhelming the receiver.
- **Addressing**: MAC addresses are used for frame delivery between devices on the same network.
- **Link Management**: Establishing and terminating logical links between nodes.

---

2. Framing

Framing refers to the division of data into manageable units called frames. Techniques include:

- **Character Count**: Uses a field in the header to specify the number of characters.
- **Flag Bytes with Byte Stuffing**: Uses special flag bytes to denote frame boundaries. If the flag byte appears in data, it is "stuffed" (escaped).
- **Bit Stuffing**: Similar to byte stuffing, but uses bit-level escaping.

---

3. Error Detection and Correction

- **Error Detection**: Identifies errors in transmitted frames.
  - **Parity Bit**: Simple error detection.
  - **Checksum**: Adds up data segments.
  - **Cyclic Redundancy Check (CRC)**: Polynomial-based checking.
- **Error Correction**:
  - **Automatic Repeat Request (ARQ)**: Requests retransmission.
  - **Forward Error Correction (FEC)**: Adds redundancy for error correction.

---

4. CRC Codes (Cyclic Redundancy Check)

- Based on binary division.
- Both sender and receiver agree on a **generator polynomial**.
- The sender divides the data by this polynomial, appends the remainder.
- Receiver re-divides; if remainder is 0, data is correct.

Elementary Data Link Protocols

1. Simplex Protocol

- Unidirectional communication.
- Assumes error-free and flow-controlled transmission.

2. Simplex Stop-and-Wait Protocol (Error-Free Channel)

- Sender transmits one frame and waits for acknowledgment (ACK) before sending the next.
- Simple but inefficient for high-latency links.

3. Simplex Stop-and-Wait Protocol (Noisy Channel)

- Adds timeout and retransmission mechanisms.
- Receiver sends ACK; if sender does not get it in time, it resends the frame.
- Uses sequence numbers to detect duplicate frames.

Sliding Window Protocols

Efficient for bidirectional, full-duplex channels with noise. Allows multiple frames to be in transit.

1. One-Bit Sliding Window Protocol

- Each frame has a 1-bit sequence number (0 or 1).
- Useful for Stop-and-Wait ARQ.

2. Go-Back-N ARQ

- Sender can send multiple frames (window size N).
- Receiver only accepts frames in order.
- On error, receiver discards all subsequent frames.
- Sender goes back and resends from the erroneous frame.

3. Selective Repeat ARQ

- More efficient than Go-Back-N.
- Receiver stores and acknowledges correct frames, even if previous ones are missing.
- Only the incorrect frame is resent.

Example Data Link Protocols

- **HDLC (High-Level Data Link Control)**: Widely used, supports both point-to-point and multipoint configurations.
- **PPP (Point-to-Point Protocol)**: Used in dial-up and WAN links.
- **Ethernet (IEEE 802.3)**: Most common LAN protocol.

Medium Access Sub-layer

Responsible for controlling how devices on a shared medium access the channel.

1. The Channel Allocation Problem

- Involves deciding how to allocate communication resources (bandwidth, time) among users.
- **Static** vs **Dynamic** allocation.

---

2. Multiple Access Protocols

ALOHA (Pure and Slotted)

- **Pure ALOHA**: Transmits anytime, retransmits after random time if collision.
- **Slotted ALOHA**: Time is divided into slots; better efficiency than pure ALOHA.

Carrier Sense Multiple Access (CSMA)

- Listen before transmitting.
- Types:
    - **1-persistent CSMA**: Transmit immediately if channel is idle.
    - **Non-persistent CSMA**: Wait and retry.
    - **p-persistent CSMA**: Probabilistic transmission on idle channel.

---

3. Collision-Free Protocols

- Ensure no collisions occur:
  - **Bit-map Protocol**: Each station gets a turn.
  - **Binary Countdown**: Uses binary addressing to resolve contention.

---

4. Wireless LANs (IEEE 802.11)

- Use **CSMA/CA** (Collision Avoidance), not CSMA/CD.
- Employ **ACK** and **RTS/CTS** mechanisms to reduce collisions.
- Operate in unlicensed spectrum (e.g., 2.4 GHz, 5 GHz).

---

5. Data Link Layer Switching

- **Switches** operate at the Data Link Layer.
- Use MAC addresses to forward frames within a LAN.
- **Switching Types**:
  - **Store-and-forward**: Entire frame is received before forwarding.
  - **Cut-through**: Starts forwarding as soon as destination MAC is read.

MODULE-3

Connecting Devices

1. Learning Bridges

- **Definition**: A **bridge** connects two or more network segments. A **learning bridge** uses the MAC addresses to decide whether to forward or filter a frame.
- **Function**: It "learns" the MAC addresses on each port and builds a forwarding table.

2. Spanning Tree Bridges

- **Purpose**: Prevent loops in a network that contains redundant bridges.
- **Protocol**: Uses **Spanning Tree Protocol (STP)** to ensure there's only one active path between two network devices.

3. Repeaters

- **Function**: A repeater regenerates and amplifies signals over the same network before they become too weak or corrupted.
- **Use Case**: Extending the distance of LANs.

4. Hubs

- **Definition**: A hub is a basic networking device that connects multiple devices in a LAN.
- **Limitation**: It operates at the physical layer and sends incoming data packets to all ports (broadcasts).

5. Bridges

- **Definition**: Bridges connect different LANs and reduce collisions by dividing traffic into segments.
- **Layer**: Operates at the Data Link Layer (Layer 2).

6. Switches

- **Definition**: A switch is like an advanced bridge with multiple ports.
- **Function**: It uses MAC addresses to forward data only to the intended recipient.
- **Layer**: Data Link Layer.

7. Routers

- **Definition**: Routers connect different networks and route data between them.
- **Layer**: Operates at the Network Layer (Layer 3).
- **Function**: Uses IP addresses and routing algorithms.

8. Gateways

- **Definition**: Gateways connect networks using different protocols.
- **Function**: Performs protocol conversion.
- **Layer**: Operates across multiple layers, including Application Layer.

Multiplexing

Definition:Multiplexing is a technique used to combine multiple signals over a single medium/channel.

Types:

1. **Frequency Division Multiplexing (FDM)**: Different frequencies for each signal.
2. **Time Division Multiplexing (TDM)**: Divides time into slots and assigns slots to signals.
3. **Wavelength Division Multiplexing (WDM)**: Used in fiber optics. Different wavelengths carry different data.
4. **Code Division Multiplexing (CDM)**: Uses unique codes to differentiate between signals.

Network Layer

Design Issues:

- **Store and forward packet switching**
- **Services provided to transport layer** (connectionless or connection-oriented)
- **Routing and addressing**
- **Packet handling and switching**

Routing Algorithms

1. Shortest Path Routing

- **Uses**: Algorithms like **Dijkstra's** to find the shortest path.
- **Metric**: Based on hops, delay, bandwidth, etc.

2. Flooding

- **Definition**: Every incoming packet is sent to all outgoing links except the one it arrived on.
- **Use Case**: Robust but inefficient.

3. Hierarchical Routing

- **Concept**: Network is divided into regions for scalability.
- **Advantage**: Reduces the size of routing tables.

4. Broadcast Routing

- **Purpose**: Sends packets to all nodes in the network.
- **Methods**: Flooding, multicasting to all.

5. Multicast Routing

- **Purpose**: Delivers data to a group of destinations.
- **Protocols**: DVMRP, PIM.

6. Distance Vector Routing

- **Concept**: Each router shares its routing table with neighbors.
- **Algorithm**: Bellman-Ford.
- **Problem**: Count to infinity issue.

7. Link State Routing

- **Concept**: Each router knows the entire topology.
- **Algorithm**: Dijkstra.
- **Example**: OSPF (Open Shortest Path First).

8. Path Vector Routing

- **Used in**: BGP (Border Gateway Protocol).
- **Concept**: Routers maintain the path information that gets updated as routes change.

Congestion Control Algorithms

Purpose: Prevent the network from being overloaded with traffic.

Examples:

1. **Leaky Bucket Algorithm**: Regulates data flow by allowing packets at a constant rate.
2. **Token Bucket Algorithm**: Allows burst traffic but limits the average rate.
3. **Choke Packet**: A router sends a warning packet back to the sender.
4. **Load Shedding**: Dropping packets when the buffer is full.
5. **Backpressure**: Inform previous nodes to slow down transmission.

Quality of Service (QoS)

Definition: QoS ensures reliable and predictable network performance.

Parameters:

- **Bandwidth**
- **Delay (Latency)**
- **Jitter** (variation in delay)
- **Packet Loss**

Techniques:

- **Traffic shaping**
- **Priority queuing**
- **Resource reservation (e.g., RSVP)**
- **Differentiated Services (DiffServ)**
- **Integrated Services (IntServ)**

MODULE-4

INTERNETWORKING TOPICS

1. Logical Addressing

Logical addressing is how devices are identified on a network using IP addresses. Unlike physical (MAC) addresses, logical addresses can change and are structured based on network hierarchy (network ID and host ID). They help route data between different networks.

2. Internet Protocols

Internet protocols are the rules for communication between devices over the internet. Key protocols include:

- **IP (Internet Protocol)** – Delivers packets from source to destination.
- **ICMP (Internet Control Message Protocol)** – Sends error messages and operational info.
- **IGMP (Internet Group Management Protocol)** – Manages multicast group memberships.

3. IP Address

An **IP address** is a unique identifier for a device on a network. It comes in two versions:

- **IPv4**: 32-bit address, written in dotted decimal (e.g., 192.168.0.1).
- **IPv6**: 128-bit address, written in hexadecimal (e.g., 2001:0db8:85a3::8a2e:0370:7334).

4. CIDR (Classless Inter-Domain Routing)

CIDR replaces traditional class-based IP addressing by allowing more flexible allocation. Instead of using fixed classes (A, B, C), CIDR uses **prefix notation** (e.g., 192.168.0.0/24) to define the subnet.

5. IPv4 Addressing

IPv4 addresses are divided into five classes (A to E) and consist of a network and host part. Subnetting allows breaking large networks into smaller ones using subnet masks.

6. IPv6 Protocol Addressing

IPv6 solves IPv4 exhaustion with 128-bit addresses. Features:

- Larger address space
- Built-in security (IPsec)
- Simplified header
- No need for NAT (Network Address Translation)

7. Address Mapping

Translating addresses between logical and physical:

- **ARP (Address Resolution Protocol)**: Maps IP to MAC address.
- **RARP (Reverse ARP)**: Maps MAC to IP address (now obsolete).
- **DHCP (Dynamic Host Configuration Protocol)**: Dynamically assigns IP addresses to devices.

8. ICMP (Internet Control Message Protocol)

Used for network diagnostics and error reporting. Common tools using ICMP include:

- `ping` – Checks connectivity.
- `traceroute` – Tracks the route taken by packets.

9. IGMP (Internet Group Management Protocol)

Used by IP hosts and adjacent routers to manage group memberships for multicast data distribution (e.g., streaming).

10. ARP and RARP

- **ARP**: Used to find the MAC address corresponding to an IP address.
- **RARP**: Used to find the IP address corresponding to a MAC address. Not commonly used today (superseded by DHCP).

11. DHCP (Dynamic Host Configuration Protocol)

Automatically assigns IP addresses and other network configuration details (gateway, DNS) to devices on a network.

TRANSPORT PROTOCOLS

1. Process-to-Process Delivery

Transport layer ensures data is delivered from one process on a source host to a specific process on the destination host, identified using **port numbers**.

2. UDP (User Datagram Protocol)

- Connectionless and unreliable
- Fast but no error recovery
- Used in real-time applications like video streaming, VoIP
- No sequencing or flow control

3. TCP (Transmission Control Protocol)

- Connection-oriented and reliable
- Provides sequencing, flow control, error checking
- Ensures data delivery with ACKs and retransmissions
- Used in web, email, file transfers

4. TCP Service Model

TCP offers services such as:

- **Reliable delivery**: Through retransmissions and acknowledgments.
- **Ordered delivery**: Ensures packets arrive in sequence.
- **Full-duplex communication**: Data can flow in both directions simultaneously.
- **Connection establishment and termination**: Using a **three-way handshake**.

5. TCP Sliding Window

A **flow control** mechanism that:

- Allows multiple packets to be sent without waiting for an ACK
- Adjusts window size dynamically
- Improves throughput

6. TCP Congestion Control

Controls network congestion by adjusting the rate of data transmission:

- **Slow start**
- **Congestion avoidance**
- **Fast retransmit**
- **Fast recovery**

7. Congestion Control and Quality of Service (QoS)

- **Congestion control**: Prevents network overload by managing data transmission rate.
- **QoS**: Mechanisms to guarantee bandwidth, delay, jitter, and reliability. Important for real-time and multimedia applications.

Conclusion

Understanding these topics helps B.Tech students grasp how data travels across networks and how the internet ensures reliable, efficient, and secure communication. These concepts form the foundation for advanced study in computer networking, cybersecurity, and cloud computing.

Let me know if you'd like visual diagrams or slides to support this explanation.

MODULE-5

Application Layer

1. Introduction to the Application Layer

- The **Application Layer** is the **topmost layer** in the **OSI** and **TCP/IP models**.
- It provides **services directly to the end users** or applications like browsers, email clients, etc.
- This layer handles **network processes to applications**, not actual data transfer (which lower layers handle).
- Examples: Web browsing, email, file transfers.

---

2. Providing Services

Application Layer provides various **services**, such as:

- **File transfer:** Sending and receiving files (e.g., FTP).
- **Email services:** Sending, receiving, and storing emails (e.g., SMTP, POP3).
- **Web services:** Accessing web content via HTTP/HTTPS.
- **Remote login:** Accessing remote systems (e.g., TELNET).
- **Name resolution:** Translating domain names to IP addresses (e.g., DNS).

The services provided are **application-specific**, meaning different applications use different protocols.

---

3. Client-Server Model

Most application layer protocols follow the **Client-Server Architecture**:

- **Client:** Initiates the request (e.g., web browser, email client).
- **Server:** Responds to the request (e.g., web server, mail server).

 Example: When you type a URL:

- The browser (client) sends an HTTP request to the web server.
- The server processes the request and sends back the response.

 Note: Some applications follow **P2P (peer-to-peer)** model (e.g., BitTorrent), but it's less common in basic application protocols.

---

4. Standard Client-Server Applications

Let's go over some **important application protocols** in detail:

A. HTTP (Hypertext Transfer Protocol)

- Used for: **Web browsing** (e.g., accessing websites).
- Port: **80 (HTTP), 443 (HTTPS)**
- Works on **request-response** model.

Example:

- Client (browser) sends a `GET` request to fetch a webpage.
- Server responds with HTML content.

HTTPS is HTTP + SSL/TLS, providing encryption and security.

B. FTP (File Transfer Protocol)

- Used for: **Transferring files** over a network.
- Ports: **21 (control), 20 (data)**
- Supports **authentication** (username/password).
- Works in **two modes**: Active and Passive.

FTP uses two separate connections:

1. **Control connection** (commands)
2. **Data connection** (file transfer)

C. Electronic Mail (Email)

Involves **3 major protocols**:

1. **SMTP (Simple Mail Transfer Protocol)**
    - Used to **send** emails.
    - Port: **25**
    - From client to server or server to server.

2. **POP3 (Post Office Protocol v3)**
    - Used to **retrieve** emails.
    - Port: **110**
    - Downloads mail to client and deletes it from the server.
3. **IMAP (Internet Message Access Protocol)**
    - Port: **143**
    - Allows **managing emails directly on the server** (without deleting).

D. TELNET (Telecommunication Network)

- Used for: **Remote login** to another computer.
- Port: **23**
- Sends data in **plain text** (not secure).
- Mostly replaced by **SSH (Secure Shell)**.

Example:

- Network admins use it to control servers remotely via command-line.

E. DNS (Domain Name System)

- Translates **domain names** (e.g., `www.google.com`) to **IP addresses**.
- Port: **53**
- Acts like the **phonebook of the internet**.

Types of DNS servers:

- **Root DNS**
- **TLD DNS (.com, .org, etc.)**
- **Authoritative DNS**

When you type a URL, the system queries DNS to get the corresponding IP address to connect to.

Quick Summary Table

Protocol	Use Case	Port	Secure?
HTTP	Web browsing	80	✗
HTTPS	Secure web browsing	443	✓(SSL/TLS)
FTP	File transfer	20/21	✗
SMTP	Sending email	25	✗(with TLS ✓)
POP3	Retrieve email (delete)	110	✗
IMAP	Retrieve email (manage)	143	✗
TELNET	Remote login	23	✗
DNS	Domain name resolution	53	✗

Key Takeaways

- Application layer is all about **user-facing services**.
- It relies on **protocols** to manage communication.
- Most services follow **client-server architecture**.
- Understanding protocols helps in **network troubleshooting**, **web development**, and **system admin tasks**.

Thank You...