Day 2 – 13/08/25

Methods of data transmission in networks – Circuit switching, Packet switching

Circuit Switching

- **Definition:** A dedicated communication path is established between two endpoints for the entire duration of the communication.
- **Analogy:** Like making a **phone call** on an old landline — the line is reserved for you until you hang up.
- **How it works:**
    1. Path setup (connection established).
    2. Data transmitted along the reserved path.
    3. Path released after completion.
- **Pros:**
    1. Guaranteed bandwidth.
    2. Predictable latency.
- **Cons:**
    1. Wastes resources if no data is being sent.
    2. Slower setup time.
- **Example protocols:** Traditional telephone network (PSTN), ISDN.

Packet Switching

- **Definition:** Data is broken into packets, each packet may take a different route to the destination, where they're reassembled.
- **Analogy:** Like sending letters — each one can travel by different routes but all reach the same address.
- **How it works:**
    1. Data split into packets.
    2. Packets sent individually through the network.
    3. Destination reassembles packets into original message.
- **Pros:**
    1. Efficient use of bandwidth (no reserved path).
    2. Robust — if one path fails, packets can reroute.
- **Cons:**
    1. Variable latency (packets may arrive out of order).
    2. Requires more complex protocols for reliability (e.g., TCP).
- **Example protocols:** Internet (IP), VoIP, most modern data networks.

Feature	Circuit Switching	Packet Switching
Connection setup	Required before data transfer	Not required
Dedicated path	Yes	No
Bandwidth usage	Fixed/reserved	Shared/dynamic
Latency	Predictable	Variable
Reliability	High (if path intact)	Depends on routing & protocol
Examples	PSTN, ISDN	Internet, VoIP, LAN/WAN

In WLANs:

- **Packet switching** is used — data is sent in frames/packets over the air using protocols like TCP/IP.
- **Circuit switching** is rare, but can appear in legacy cellular voice calls.

What is IP?

- **Full name:** Internet Protocol
- **Purpose:** Defines how data is **addressed** and **routed** across networks.
- **Layer:** Network Layer (Layer 3) in the OSI model.
- **Key job:** Deliver data packets from a **source IP address** to a **destination IP address** across multiple networks.

Think of IP as the **postal system** — it puts addresses on envelopes (packets) and makes sure they get routed to the right mailbox.

How IP Works

1. **Encapsulation** – IP wraps data from TCP/UDP into packets with source & destination IP.
2. **Routing** – Routers look at the destination IP and forward the packet toward the next hop.
3. **Delivery** – When the packet reaches the destination host, IP passes it to the right transport layer protocol.

Limitations of IP – No reliablility, No ordering, No security

IP Packet Structure (IPv4)

Field	Purpose
Version	IP version (4 or 6)
Header Length	Size of the header
Total Length	Size of packet (header + data)
Identification, Flags, Fragment Offset	Handling packet fragmentation
TTL (Time To Live)	Limits packet lifetime (hops)
Protocol	Upper layer protocol (TCP=6, UDP=17, ICMP=1)
Header Checksum	Error checking for header
Source IP	Sender's IP address
Destination IP	Receiver's IP address
Data	Encapsulated payload (e.g., TCP segment, UDP datagram)

IP Addressing

- IP address - Unique identifier address of devices in a computer network
- Notation – binary/decimal/hexadecimal eg:192.168.1.10
- 2 parts – network id & host id 192.168.1.0 , 10

Versions of IP addresses:

- **IPv6 (Internet Protocol version 6)** and **IPv4 (Internet Protocol version 4)** are two different versions of the Internet Protocol used for addressing devices in a network.

IPv4	IPv6
32 bits ip addr	128 bits ip addr
4 octets of 8 bits each	8 groups of 16 bits each
Dotted decimal notation	Dotted hexadecimal notation
Uses subnet mask component (32 bits) to identify n/w & host addr	Fixed bits for n/w & host addr --> 64 bits each.
IP addr: 192.168.1.10 Subnet mask: 255.255.255.0	2001:0000:0db8:85a3:0000:8a2c:3233:3312

IPv4 Addressing:

- **Format**: 32-bit numbers, represented in **dotted-decimal format** (e.g, 192.168.1.1).
- **Address Space**: IPv4 provides approximately **4.3 billion unique addresses**
- **Address Classes**: IPv4 addresses are divided into classes (A, B, C, D, and E), and the addressing scheme supports both private and public addresses.

IPv6 Addressing:

- **Format**: 128-bit numbers, represented in **hexadecimal format** (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
- **Address Space**: IPv6 provides a vastly larger address space, offering about **340 undecillion (3.4 × 10^38)** unique addresses, ensuring that every device on the planet (and beyond) can have a unique address.
- **No Need for NAT**: Because of the vast address space, IPv6 does not require **NAT (Network Address Translation)**, unlike IPv4.
- **Simpler Header**: IPv6 has a simplified header structure that improves processing efficiency.

Key Difference:

IPv4 uses 32-bit addresses, while **IPv6** uses 128-bit addresses, providing a significantly larger address space and additional features like built-in security and simplified routing.

6.2 Subnet mask

- Distinguish n/w addr from host addr
- 32 bit
- Dotted decimal, binary or /n notation
- Sets bit at each network bit position, resets host bits

IP addr: 192.168.1.10

Subnet mask: 255.255.255.0
192.168.1.0/24 -->
n/w addr – 192.168.1.0
Bc addr – 192.168.1.255
Available host address – 254
192.168.1.1 to 192.168.1.254

6.3 Subnetting

- Large IP n/w to smaller independent networks with a range of IP addrs
- IP addrs classes – class A,B,C,D (Multicast),E (Reserved) based on no:of bits for network id.

	8 bits	8 bits	8 bits	8 bits
Class A	n/w	host	host	host
Class B	n/w	n/w	host	host
Class C	n/w	n/w	n/w	host

- 1st addr – n/w addr, last addr – broadcast addr
  - 192.168.0.0/24 network to 3 subnets with 50,30,20 hosts

hosts	Add + 2	Nearest ^2	Subnet mask	Subnet allocation	VLAN alloc
50	52	64 (2^6)	/26 (255.255.255.192)	sn1– 192.168.0.0/26 Host – 192.168.0.1 to 192.168.0.62 Bc – 192.168.0.63	VLAN 1 IP 192.168.0.0/26
30	32	32 (2^5)	/27 (255.255.255.224)	sn1– 192.168.0.64/27 Host – 192.168.0.65 to 192.168.0.94 Bc – 192.168.0.95	VLAN 2 IP 192.168.0.64/27

20	22	32 (2^5)	/27 (255.255.255.224)	sn1– 192.168.0.96/27 Host – 192.168.0.97 to 192.168.0.126 Bc – 192.168.0.127	VLAN 3 IP 192.168.0.96/27

- **CIDR (Classless Inter-Domain Routing) Notation**

CIDR (Classless Inter-Domain Routing) is a way to write an IP address and its subnet mask together.

Format:

`<IP_address>/<prefix_length>`

- `/prefix_length` tells how many bits in the subnet mask are **1s** (network bits).
- Example:
  - `/24` → `255.255.255.0` (first 24 bits are network bits).
  - `/26` → `255.255.255.192` (first 26 bits are network bits).

Example:

```
192.168.10.0/26
```

- **Network bits:** 26
- **Host bits:** 6 (because IPv4 has 32 bits total)
- **Number of hosts:** $2^6 - 2 = 62$ usable IP addresses.

Quick CIDR Table

CIDR	Subnet Mask	Hosts (usable)
/24	255.255.255.0	254
/25	255.255.255.128	126
/26	255.255.255.192	62
/27	255.255.255.224	30
/28	255.255.255.240	14

Static IP addressing

- **Definition:** You manually assign a fixed IP address to a device.
- **How it works:**
  - You go into the device's network settings and set:
    - IP address (e.g., `192.168.1.50`)
    - Subnet mask (e.g., `255.255.255.0`)
    - Gateway (e.g., `192.168.1.1`)

        - DNS servers
    - This stays the same until you manually change it.
- **Advantages:**
    - Predictable — great for servers, printers, CCTV cameras, or any device that must always be reachable at the same address.
    - No dependency on DHCP.
- **Disadvantages:**
    - More manual setup.
    - Risk of **IP conflicts** if two devices are given the same address.
- Eg. Office printer set to `192.168.1.10` so everyone can always print to the same address.

Dynamic IP addressing

- **Definition:** A device automatically gets an IP address from a **DHCP server** (Dynamic Host Configuration Protocol).
- **How it works:**
    1. Device sends a **DHCP Discover** broadcast.
    2. DHCP server replies with an available IP + subnet mask + gateway + DNS.
    3. Address is leased for a certain time (lease time).
- **Advantages:**
    1. No manual setup.
    2. Reduces IP conflicts.
    3. Easy for networks with many users (e.g., coffee shop Wi-Fi).
- **Disadvantages:**
    1. IP may change over time (not great for servers unless you use DHCP reservation).
    2. If DHCP server fails, new devices can't join the network.
- Eg. Your phone gets a random IP (e.g., `192.168.1.25`) each time you connect to your home Wi-Fi.

Public IP addresses

- globally unique IP addresses assigned to devices that need to be directly accessible over the internet.
- routable on the global internet-connected
- issued by Internet Service Providers (ISPs) or assigned by the Internet Assigned Numbers Authority (IANA).
- typically used by servers, websites, and other services that need to be accessed by users anywhere on the internet.

Private IP addresses

- used by devices like computers, printers, and smartphones within private local area networks (LANs).

- not routable over the internet, meaning that devices using private IPs cannot be accessed directly from the internet
- designed for use within an organization's internal network.
- The specific ranges for private IP addresses are defined by RFC 1918:
  - 10.0.0.0 to 10.255.255.255
  - 172.16.0.0 to 172.31.255.255
  - 192.168.0.0 to 192.168.255.255
- These devices can access the internet through Network Address Translation (NAT), which allows them to share a single public IP address to connect to the internet. This helps conserve global IP address space and improves security by preventing direct access to private devices from outside the local network.

TCP/UDP

Feature	TCP	UDP
Connection Type	Connection-oriented (needs connection before sending data)	Connectionless (no handshake, just sends packets)
Reliability	Reliable — ensures data delivery, re-transmits lost packets	Unreliable — no guarantee of delivery or order
Ordering	Packets arrive in order (sequence numbers)	No guarantee of order
Speed	Slower (overhead due to acknowledgments & checks)	Faster (less overhead)
Error Handling	Detects and re-sends lost/corrupted packets	Minimal error checking (checksum only)
Use Cases	Web browsing, file transfer, emails	Streaming, gaming, VoIP
Example Protocols	HTTP, HTTPS, FTP, SMTP	DNS, DHCP, video streaming

TCP Handshake (Three-Way Handshake):

The 3-way handshake is the process TCP uses to establish a reliable connection between client and server.
Steps:

1. **SYN** → Client sends a **SYN** (synchronize) packet to server to request a connection.
2. **SYN-ACK** → Server replies with **SYN-ACK** (synchronize + acknowledge).
3. **ACK** → Client sends an **ACK** (acknowledge) back to the server to confirm.

After this, the connection is established, and data transfer can start.

Diagram:

```
Client                    Server
   | ------ SYN ------>  |
```

```
| <---- SYN-ACK ----  |
| ------ ACK ------>  |
```

Common TCP/IP Ports in WLAN

Protocol	Port Number	TCP/UDP	Purpose
HTTP	80	TCP	Web browsing
HTTPS	443	TCP	Secure web browsing
FTP	21	TCP	File transfer
DNS	53	TCP/UDP	Domain name resolution
DHCP	67, 68	UDP	IP address assignment
SMTP	25	TCP	Email sending
IMAP	143	TCP	Email retrieval
POP3	110	TCP	Email retrieval
Telnet	23	TCP	Remote terminal access
SSH	22	TCP	Secure remote access

ICMP

ICMP (Internet Control Message Protocol) is a network layer protocol used by devices (like routers, hosts) to send control messages and error reports — it's not for sending user data, but for diagnosing and managing network communication.

Key Points Protocol type: Network Layer (works alongside IP)

- **Purpose:** Send messages about:
    - Network errors
    - Connectivity issues
    - Debugging and testing
- **Common tools using ICMP:** `ping`, `traceroute`

How It Works

- When a device detects an issue (like unreachable host, TTL expired, or packet dropped), it sends an **ICMP message** back to the sender.
- ICMP packets are usually encapsulated inside **IP packets**.
- ICMP messages contain:
    - **Type** (e.g., Echo Request, Echo Reply)
    - **Code** (more detail on the type)
    - **Checksum**
    - Optional data (e.g., part of the original IP packet)

Common ICMP Message Types

Type	Name	Use Case
0	Echo Reply	Response to ping
3	Destination Unreachable	No route to host/network
8	Echo Request	Sent by ping to test reachability
11	Time Exceeded	TTL expired (used in traceroute)

Real-Life Example

When you run:

```
ping google.com
```

1. Your system sends an **ICMP Echo Request** to Google's server.
2. The server responds with an **ICMP Echo Reply**.
3. The round-trip time is measured, helping you check connectivity.

Traceroute is a network diagnostic tool that shows the path packets take from your device to a destination, and the time it takes to reach each hop (router) along the way.

How It Works

1. **Sends packets with gradually increasing TTL (Time-To-Live)**
    - TTL starts at `1`. Each router that handles the packet decreases TTL by `1`.
    - When TTL reaches `0`, the router **drops the packet** and sends back an **ICMP "Time Exceeded"** message.
2. **Records the source of the ICMP reply** (the router's IP).
3. Increments TTL and repeats, discovering the **next hop**.
4. Stops when the packet reaches the final destination (which replies with an ICMP Echo Reply or TCP/UDP response).

Example

```
traceroute google.com     # Linux / macOS
tracert google.com        # Windows
```

Sample output:

```
 1  192.168.1.1      1.234 ms
 2  10.0.0.1         4.567 ms
 3  203.0.113.5      15.321 ms
 4  142.250.46.46    30.456 ms
 5  google.com       31.002 ms
```

Key Details

- **Linux/macOS**: Uses UDP packets by default (port 33434+).

- **Windows**: Uses ICMP Echo Requests.
- **Output Columns**:
    - **Hop Number** (1, 2, 3…)
    - **Router IP / Hostname**
    - **RTT** (Round Trip Time) — usually 3 measurements per hop.

Real-Life Use Cases

- Checking where a network slowdown happens.
- Identifying routing issues between ISPs.
- Seeing how far a server is in network terms (number of hops).

Jitter

- The **variation** in latency between packets.
- Represents inconsistency.
- In milliseconds (ms)

Latency

- The **time it takes** for a packet of data to travel from source to destination.
- In milliseconds (ms)

Data rate

- The **maximum possible speed** the network link can transfer data under ideal conditions.
- **Measured in: bits per second** (bps, Kbps, Mbps, Gbps).
- **Example:**
  Your Wi-Fi is rated at **100 Mbps** — that's the *theoretical* maximum data rate.

Throughput

- The **actual speed** at which useful data is successfully transferred over the network.
- **Measured in:** Same units (bps, Mbps, Gbps).
- **Example:**
  On your 100 Mbps Wi-Fi, you actually measure **72 Mbps** during a speed test — that's throughput.

Throughput is always ≤ data rate because of:

- Protocol overhead (headers, error checks)
- Network congestion
- Latency/jitter
- Signal interference (in wireless links)