**UNIT – 3**

**Data Link Layer Overview**

* **Purpose**: The Data Link Layer (Layer 2) is responsible for node-to-node data transfer, error detection and correction, framing, and managing access to the physical medium. It ensures that data packets are delivered error-free from the sender to the receiver within the same network.

**Data Encoding**

* **Introduction**: Data encoding is the process of converting data into a specific format for efficient transmission over a network. It is crucial for ensuring that the data can be accurately received and interpreted.

**1. Block Coding**

* **Definition**: Block coding is a technique where data is divided into fixed-size blocks for encoding. Each block is transformed into a codeword that can be transmitted.
* **Purpose**: This method adds redundancy to the data to improve error detection and correction capabilities.
* **Examples**: Common block codes include Hamming codes and Reed-Solomon codes, which can correct errors in data transmission.

**2. Cyclic Codes**

* **Definition**: Cyclic codes are a type of linear block code where any cyclic shift of a codeword results in another codeword. They are widely used in error detection.
* **Advantages**: Easier to implement in hardware and allow for efficient error detection and correction processes.
* **Example**: CRC (Cyclic Redundancy Check) is a common cyclic code used for error detection.

**3. Checksum**

* **Definition**: A checksum is a simple error-detection method that involves summing up the binary values of a data block and appending the result to the data.
* **Purpose**: It allows the receiving device to verify the integrity of the received data by recalculating the checksum and comparing it with the transmitted value.
* **Limitations**: While effective for simple error detection, checksums cannot detect all types of errors, particularly when multiple errors occur.

**Framing**

* **Definition**: Framing is the process of dividing the data stream into manageable units (frames) for transmission over the network.
* **Types of Framing**:
* **Character-oriented Framing**: Frames are delimited by special characters (e.g., ASCII).
* **Bit-oriented Framing**: Frames are defined by bit patterns (e.g., HDLC).
* **Purpose**: Framing allows for the identification of the beginning and end of each frame, facilitating synchronization and error handling.

**Channels**

* **Noiseless Channels**:
* **Definition**: Ideal communication channels that transmit data without any interference or distortion.
* **Characteristics**: High reliability and accurate data transmission.
* **Noisy Channels**:
* **Definition**: Channels that experience interference, causing potential errors in data transmission.
* **Characteristics**: Error detection and correction mechanisms (e.g., parity bits, checksums, CRC) are essential for reliable communication.

**Transmission Types**

* **Asynchronous Transmission**:
* **Definition**: Data is transmitted without a synchronized clock signal, meaning that each byte is sent independently.
* **Characteristics**: Start and stop bits are added to indicate the beginning and end of each byte, which allows for flexibility in data transmission.
* **Examples**: Commonly used in serial communication, such as RS-232.
* **Synchronous Transmission**:
* **Definition**: Data is transmitted in a continuous stream, synchronized by a clock signal shared between sender and receiver.
* **Characteristics**: No start or stop bits are needed, allowing for faster data transfer rates.
* **Examples**: Used in applications like Ethernet and high-speed serial communications.
* **Full Duplex Transmission**:
* **Definition**: Allows data to be transmitted and received simultaneously over the same channel.
* **Example**: Telephone conversations, where both parties can speak and listen at the same time.
* **Half Duplex Transmission**:
* **Definition**: Data can be transmitted in both directions, but not simultaneously. Each device must take turns to send and receive data.
* **Example**: Walkie-talkies, where one party speaks while the other listens.

**1. Binary Encoding Schemes**

These encoding schemes are critical for transmitting digital data over various mediums.

**1.1 NRZ (Non-Return to Zero)**

* **Definition**: A binary encoding method where 1s are represented by one voltage level and 0s by another. The signal does not return to zero between bits.
* **Types**:
* **NRZ-L (Level)**: Uses a specific voltage level to represent each bit.
* **NRZ-I (Inversion)**: A change in voltage level represents a 1, while no change represents a 0.
* **Advantages**: Simple implementation, efficient bandwidth usage.
* **Disadvantages**: Lacks synchronization; long runs of 0s or 1s can lead to loss of synchronization.

**1.2 Bipolar AMI (Alternate Mark Inversion)**

* **Definition**: Uses three voltage levels: positive, negative, and zero. 1s are represented by alternating positive and negative voltages, while 0s are represented by zero voltage.
* **Advantages**: Good error detection and synchronization; eliminates DC component.
* **Disadvantages**: Requires more bandwidth than NRZ.

**1.3 B8ZS (Bipolar with 8-Zero Substitution)**

* **Definition**: A modification of AMI to handle long sequences of 0s by replacing them with specific patterns that maintain synchronization.
* **Advantages**: Maintains synchronization over long sequences of zeros; reduces errors.
* **Usage**: Commonly used in T1 lines.

**1.4 HDB3 (High-Density Bipolar 3 Zeros)**

* **Definition**: Another modification of AMI, replacing sequences of three consecutive zeros with a bipolar violation (a single positive or negative pulse).
* **Advantages**: Reduces the DC component and allows for better error detection.
* **Usage**: Used in digital telecommunication systems, especially in Europe.

**2. Modulation Techniques**

These techniques convert digital signals into analog form for transmission over various mediums.

**2.1 ASK (Amplitude Shift Keying)**

* **Definition**: Represents digital data by varying the amplitude of the carrier wave.
* **Advantages**: Simple implementation.
* **Disadvantages**: Susceptible to noise; not efficient for long-distance communication.

**2.2 FSK (Frequency Shift Keying)**

* **Definition**: Uses different frequencies to represent binary data. A specific frequency is assigned for 1s and another for 0s.
* **Advantages**: More resistant to noise than ASK.
* **Disadvantages**: Requires more bandwidth than ASK.

**2.3 PSK (Phase Shift Keying)**

* **Definition**: Represents data by changing the phase of the carrier wave. Common variants include BPSK (Binary Phase Shift Keying) and QPSK (Quadrature Phase Shift Keying).
* **Advantages**: More efficient than ASK and FSK; better noise immunity.
* **Usage**: Widely used in wireless communication.

**3. Pulse Code Modulation (PCM)**

* **Definition**: A method used to digitally represent analog signals. The amplitude of the analog signal is sampled at regular intervals and converted into a digital signal.
* **Advantages**: High quality of transmitted signal; robust against noise.
* **Usage**: Used in digital telephony and audio applications.

**4. Analog Modulation Techniques**

These techniques modulate analog signals to carry information.

**4.1 AM (Amplitude Modulation)**

* **Definition**: Varies the amplitude of the carrier wave in proportion to the amplitude of the input signal.
* **Advantages**: Simple to implement; widely used in AM radio broadcasting.
* **Disadvantages**: Susceptible to noise and interference.

**4.2 FM (Frequency Modulation)**

* **Definition**: Varies the frequency of the carrier wave in accordance with the amplitude of the input signal.
* **Advantages**: Better noise resistance than AM; improved sound quality.
* **Usage**: Commonly used in FM radio broadcasting.

**4.3 PM (Phase Modulation)**

* **Definition**: Varies the phase of the carrier wave to represent the data.
* **Advantages**: Similar benefits to FM; often used in conjunction with PSK.
* **Usage**: Used in digital communication systems and in some radio applications.

**Data Link Control Overview**

Data Link Control (DLC) is responsible for reliable transmission of data frames between nodes in a network. It handles flow control and error detection/correction to ensure that data is sent and received accurately.

**1. Flow Control**

Flow control mechanisms manage the pace of data transmission between sender and receiver to prevent buffer overflow.

**1.1 Stop and Wait**

* **Definition**: A simple flow control method where the sender transmits one frame and waits for an acknowledgment (ACK) from the receiver before sending the next frame.
* **Advantages**:
* Simple implementation.
* Low overhead.
* **Disadvantages**:
* Inefficient for long-distance communication due to idle time while waiting for ACKs.
* Throughput is limited by round-trip time (RTT).
* **Use Case**: Suitable for applications with low bandwidth and latency, such as sensor networks.

**1.2 Sliding Window**

* **Definition**: A more efficient flow control technique that allows multiple frames to be in transit before requiring an acknowledgment. The sender maintains a "window" of frames that can be sent before needing to wait for ACKs.
* **Advantages**:
* Increases throughput by allowing continuous transmission of frames.
* Efficiently uses available bandwidth.
* **Mechanism**:
* The sender can transmit a number of frames defined by the window size. Once an ACK for the first frame in the window is received, the window slides forward to allow new frames to be sent.
* **Use Case**: Commonly used in high-speed networks (e.g., TCP).

**2. Error Detection**

Error detection mechanisms identify whether errors occurred during the transmission of data.

**2.1 Parity Check**

* **Definition**: A simple error detection technique that adds a parity bit to data to make the total number of 1s either even (even parity) or odd (odd parity).
* **Advantages**:
* Easy to implement.
* Requires minimal overhead.
* **Disadvantages**:
* Can only detect single-bit errors and cannot correct errors.
* **Use Case**: Often used in memory storage systems and simple communication protocols.

**2.2 Cyclic Redundancy Check (CRC)**

* **Definition**: A more robust error detection method that treats data as a polynomial and uses polynomial division to generate a checksum. The receiver performs the same division to check for errors.
* **Advantages**:
* Can detect burst errors (multiple bits in error).
* More reliable than parity check.
* **Disadvantages**:
* More complex implementation.
* **Use Case**: Widely used in Ethernet frames, disk storage devices, and protocols like HDLC and Wi-Fi.

**3. Error Control**

Error control mechanisms ensure that errors detected in transmission are either corrected or retransmitted.

**3.1 Stop and Wait ARQ (Automatic Repeat reQuest)**

* **Definition**: A combination of Stop and Wait flow control and error detection. After sending a frame, the sender waits for an ACK. If no ACK is received after a timeout, the frame is retransmitted.
* **Advantages**:
* Simple and reliable.
* **Disadvantages**:
* Inefficient due to potential timeouts and retransmissions.
* **Use Case**: Suitable for networks with low error rates.

**3.2 Go-Back-N ARQ**

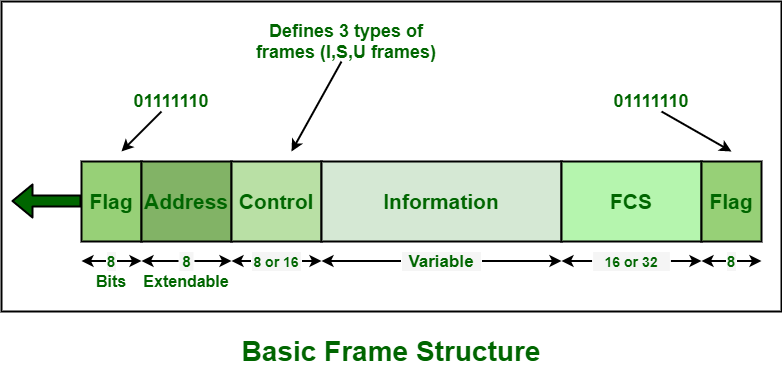
* **Definition**: A sliding window error control protocol where the sender can send multiple frames before needing an acknowledgment. If an error is detected in a frame, all subsequent frames are retransmitted.
* **Advantages**:
* Increases efficiency over Stop and Wait ARQ.
* Uses the sliding window technique for better bandwidth utilization.
* **Disadvantages**:
* Retransmitting multiple frames can waste bandwidth if many errors occur.
* **Use Case**: Useful in scenarios where the likelihood of errors is moderate.

**3.3 Selective-Reject ARQ (Selective Repeat ARQ)**

* **Definition**: Similar to Go-Back-N, but only the erroneous frames are retransmitted rather than all subsequent frames.
* **Advantages**:
* More efficient use of bandwidth since only specific frames are resent.
* Reduces the number of retransmissions compared to Go-Back-N.
* **Disadvantages**:
* More complex to implement due to the need for maintaining multiple frames in the sender and receiver buffers.
* **Use Case**: Effective in high-speed networks with higher error rates.

**1. HDLC (High-Level Data Link Control)**

* **Overview**: HDLC is a bit-oriented synchronous data link layer protocol developed by ISO. It is used for point-to-point and multipoint connections and is designed to provide reliable communication between network nodes.
* **Features**:
* **Frame Structure**: HDLC frames include a flag field, address field, control field, information field, and a frame check sequence (FCS) for error detection.
* **Modes of Operation**:
  + **Normal Response Mode (NRM)**: The primary station controls the communication and the secondary stations respond when prompted.
  + **Asynchronous Balanced Mode (ABM)**: Both stations can initiate communication, making it suitable for peer-to-peer applications.
  + **Point-to-Point Mode**: Used for direct connections between two nodes.
* **Frame Types in HDLC:**
  + **Information Frames (I-frames):** Used to carry user data and control information.
  + **Supervisory Frames (S-frames):** Used for flow and error control.
  + **Unnumbered Frames (U-frames):** Used for additional functions such as setting up and disconnecting the link.
* **Error Detection**: Utilizes the FCS for error checking, often implemented using CRC (Cyclic Redundancy Check).
* **Applications**: Used in WAN connections and point-to-point communication links (e.g., leased lines) .



**2. PPP (Point-to-Point Protocol)**

* **Overview**: PPP is a widely used protocol for establishing a direct connection between two networking nodes. It can provide connection authentication, transmission encryption, and compression.
* **Features**:
* Supports multiple network layer protocols (IPv4, IPv6, IPX, AppleTalk).
* Includes a Link Control Protocol (LCP) to establish, configure, and test the data link connection.
* Allows for optional authentication through protocols like PAP (Password Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol).
* **Error Detection**: Incorporates simple error detection using a checksum .

**3. LCP (Link Control Protocol)**

* **Overview**: A part of PPP that handles the setup and configuration of the data link layer.
* **Features**:
* Negotiates the link parameters.
* Monitors the link quality and can terminate the connection if errors exceed acceptable thresholds.
* **Functionality**: LCP is essential for ensuring that the data link layer can adapt to varying network conditions and protocols .

**4. SDLC (Synchronous Data Link Control)**

* **Overview**: Developed by IBM, SDLC is a synchronous data link protocol that provides both half-duplex and full-duplex transmission modes.
* **Features**:
* Supports error detection through CRC.
* Primarily used in IBM’s networking products.
* **Applications**: Often utilized in IBM mainframe systems for communications over serial lines .

**5. LAPB (Link Access Procedure, Balanced)**

* **Overview**: LAPB is a protocol derived from HDLC and is used in X.25 packet-switched networks.
* **Features**:
* Provides error detection and correction mechanisms.
* Supports both point-to-point and multipoint configurations.
* **Use Case**: Commonly used in older network technologies, particularly in public packet-switched networks .

**UNIT - 4**

**Network Layer**

The network layer is responsible for routing data packets across network boundaries and ensuring proper addressing for data transfer. Below are its key functions:

1. **Logical Addressing**:

* Logical addressing refers to the use of IP addresses (IPv4 or IPv6) to uniquely identify each device on a network.
* **IP Addressing**: Devices connected to the internet use a unique address, which is divided into the network and host parts. IPv4 uses a 32-bit address, while IPv6 uses a 128-bit address.
* **Private and Public IPs**: Private IP addresses are used within local networks, and public IP addresses are assigned for external internet communication.
* **Subnetting**: Dividing a network into smaller sub-networks for better management.

1. **Internetworking**:

* Internetworking connects multiple networks to create a larger network using routers and gateways.
* **Router**: A device that forwards data packets between different networks using routing tables and algorithms.
* **Routing Protocols**: Determines the best path for data. Examples include **OSPF** (Open Shortest Path First), **RIP** (Routing Information Protocol), and **BGP** (Border Gateway Protocol).
* **IP Fragmentation and Reassembly**: If a data packet is larger than the maximum transmission unit (MTU) of a network, it is fragmented and later reassembled by the receiving system.

1. **Address Mapping**:

* Address mapping converts network layer addresses (IP addresses) to data link layer addresses (MAC addresses).
* **ARP (Address Resolution Protocol)**: Translates IP addresses to MAC addresses in IPv4 networks.
* **RARP (Reverse Address Resolution Protocol)**: Maps MAC addresses back to IP addresses.
* **NAT (Network Address Translation)**: A process that allows multiple devices on a local network to share a single public IP address for external communication.

**LAN Technology**

Local Area Network (LAN) technologies define how devices within a limited area (such as a building) communicate.

1. **LAN Architecture**:

* **Topology**: The arrangement of devices in a LAN. Popular topologies include:
  + **Star Topology**: Devices are connected to a central hub or switch.
  + **Bus Topology**: All devices share a single communication line.
  + **Ring Topology**: Devices are connected in a closed loop, passing data from one device to another.
* **LAN Protocols**: The set of rules governing data communication over the network. Examples include Ethernet and Token Ring.
* **Switching**: **Layer 2 switches** operate at the data link layer, forwarding frames based on MAC addresses.

1. **IEEE 802 Standards**: The **IEEE 802** family of standards defines the physical and data link layers for LAN communication. Key standards include:

* **IEEE 802.3 (Ethernet)**:
  + The most widely used standard for LANs.
  + Uses CSMA/CD (Carrier Sense Multiple Access with Collision Detection) for data transmission.
  + Supports various transmission speeds, from **Ethernet (10 Mbps)** to **Gigabit Ethernet (1 Gbps)** and beyond.
* **IEEE 802.11 (Wi-Fi)**:
  + The standard for wireless LAN (WLAN) communication.
  + Defines how devices communicate over a wireless medium using radio frequencies.
  + **Wi-Fi Versions**: Includes **802.11a, b, g, n, ac, and ax** for various data rates and coverage.
* **IEEE 802.5 (Token Ring)**:
  + Uses a token-passing method for controlling access to the network.
  + Primarily used in older networks, less common today.
* **IEEE 802.15 (Bluetooth)**:
  + Standard for short-range wireless communication between devices, often used for personal area networks (PANs).

**Ethernet and Medium Access Control**

1. **Ethernet Overview**:

* Ethernet is a widely used LAN technology that connects devices in a local area network (LAN) using a common communication medium.
* **Standardization**: The IEEE 802.3 standard governs Ethernet specifications, which include physical layer and data link layer protocols.
* **Topology**: Ethernet networks can utilize various topologies, but star topology with a central switch is common.

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

* **Medium Access Control Protocol**: CSMA/CD manages how devices on a shared network medium can transmit data without interfering with each other.
* **Collision Detection**: Devices listen to the network (carrier sensing) before transmitting. If two devices transmit simultaneously, a collision occurs.
* **Collision Handling**: Upon detecting a collision, devices stop transmitting, wait a random period (backoff algorithm), and then attempt to retransmit.
* **Limitations**: CSMA/CD is less efficient under high network traffic, leading to increased collisions and delays.

1. **Ethernet Variants**:

* **Standard Ethernet**: Originally supported 10 Mbps data rates over coaxial cables.
* **Fast Ethernet (IEEE 802.3u)**: Improved standard supporting data rates up to 100 Mbps using twisted pair and fiber optic cables.
* **Gigabit Ethernet (IEEE 802.3z)**: Further enhancement that supports 1 Gbps speeds, leveraging both copper and fiber optics.
* **10 Gigabit Ethernet (IEEE 802.3ae)**: Supports data rates of 10 Gbps, primarily using fiber optic technology.
* **100 Gigabit Ethernet (IEEE 802.3ba)**: An even higher speed option that is used for backbone connections in large networks.

**Other LAN Systems**

1. **Token Ring**:

* Developed by IBM, Token Ring uses a token-passing protocol where a token circulates around the network.
* Only the device holding the token can transmit data, reducing the chances of collisions.
* Operates at speeds of 4 Mbps or 16 Mbps but is less common today due to the rise of Ethernet.

1. **FDDI (Fiber Distributed Data Interface)**:

* A high-speed network technology that uses fiber optic cables and operates at speeds of 100 Mbps.
* Uses a dual ring topology for redundancy, allowing for fault tolerance. If one ring fails, the other can continue to transmit data.
* Typically used for backbone connections in enterprise networks due to its high speed and reliability.

1. **ATM (Asynchronous Transfer Mode)**:

* A cell-switching technology that transmits data in fixed-size cells (53 bytes).
* Designed to support various traffic types (voice, video, and data) with guaranteed bandwidth and low latency.
* Not commonly used in LANs today but is still relevant for WAN connections.

1. **Fiber Channel**:

* A high-speed network technology primarily used for storage area networks (SANs).
* Supports data transfer rates from 1 Gbps up to 128 Gbps.
* Provides reliable and efficient data transfer between storage devices and servers, using fiber optic cables for higher performance.

**Wireless LANs (WLANs)**

* **Overview**: Wireless Local Area Networks allow devices to connect and communicate without physical cables. They use radio waves to transmit data.
* **IEEE 802.11 Standards**:
* Various standards define WLAN technology, including **802.11a, b, g, n, ac, and ax** (Wi-Fi 6).
* **802.11a**: Operates at 5 GHz with speeds up to 54 Mbps.
* **802.11b**: Operates at 2.4 GHz with speeds up to 11 Mbps.
* **802.11g**: Combines features of a and b, operating at 2.4 GHz with speeds up to 54 Mbps.
* **802.11n**: Introduced MIMO (Multiple Input Multiple Output) technology, allowing speeds up to 600 Mbps.
* **802.11ac**: Works in the 5 GHz band, offering speeds up to several Gbps.
* **802.11ax (Wi-Fi 6)**: Provides improved efficiency, capacity, and performance, especially in dense environments.
* **Security**:
* WPA2 and WPA3 encryption protocols are essential for securing wireless connections.
* Techniques like **MAC address filtering** and **SSID hiding** enhance security.

**Bridges**

* **Definition**: Bridges are networking devices that connect two or more network segments, allowing them to function as a single network.
* **Types**:
* **Transparent Bridges**: Operate at Layer 2, making decisions based on MAC addresses without modifying the data packets.
* **Source Route Bridges**: Operate in Token Ring networks, using the path information contained in the packets.
* **Learning Bridges**: Learn which devices are on which segments to make intelligent forwarding decisions, reducing unnecessary traffic.
* **Functions**:
* Reduce collision domains, improving network performance.
* Filter traffic to ensure data packets are sent only to the intended recipient segments.

**Latest Trends in LAN Technologies**

* **Increased Adoption of Wireless Technologies**: WLANs continue to gain popularity with the advancement of wireless standards like Wi-Fi 6 and upcoming Wi-Fi 7.
* **SDN (Software-Defined Networking)**: Offers centralized control over network resources, improving flexibility and management.
* **IoT Integration**: Expanding LANs to support Internet of Things devices, necessitating robust security and network management solutions.
* **Network Slicing**: Involves creating multiple virtual networks on a single physical network infrastructure, optimizing resource use.
* **Advanced Security Protocols**: Increased focus on zero-trust security models, ensuring that every access request is authenticated and authorized.

**LAN Devices: Specifications of L2 and L3 Switches**

1. **Layer 2 Switches (L2)**:

* Operate at the data link layer, forwarding data based on MAC addresses.
* **Specifications**:
  + **Port Density**: Commonly available in 24, 48, or 96 port configurations.
  + **Throughput**: Typically supports non-blocking throughput for all ports (e.g., 1 Gbps to 10 Gbps).
  + **Features**:
  + **VLAN Support**: Allows segmentation of networks for improved performance and security.
  + **Spanning Tree Protocol (STP)**: Prevents loops in network topologies.
  + **Link Aggregation**: Combines multiple network connections for increased bandwidth.

1. **Layer 3 Switches (L3)**:

* Operate at the network layer, enabling routing capabilities based on IP addresses.
* **Specifications**:
  + **Routing Protocol Support**: Can support protocols like OSPF, EIGRP, and BGP.
  + **Routing Capacity**: Offers high-performance routing capabilities, often at the same speed as Layer 2 switching.
  + **Multicast and Broadcast Support**: Can efficiently manage multicast traffic using IGMP snooping.
  + **Advanced Features**:
  + **Quality of Service (QoS)**: Prioritizes critical network traffic.
  + **Access Control Lists (ACLs)**: Provides security by controlling the traffic flow based on predefined rules.

| **Basis** | **Layer 2 Switch** | **Layer 3 Switch** |
| --- | --- | --- |
| **OSI Layer** | Operates at **Layer 2** (Data Link Layer) of the OSI model. | Operates at **Layer 3** (Network Layer) of the OSI model. |
| **Forwarding Method** | Forwards data based on **MAC addresses** (hardware address). | Routes data based on **IP addresses** (network address). |
| **Functionality** | Primarily performs **switching** of frames within a LAN. | Can perform both **routing** and switching functions. |
| **Speed** | **Faster** as it does not inspect Layer 3 packet information. | **Slower** compared to Layer 2 as it checks Layer 3 info. |
| **Traffic Handling** | Used to **reduce traffic** on local networks. | Often used for **inter-VLAN routing** and managing traffic across networks. |
| **Broadcast Domain** | **Single broadcast domain**, all connected devices share the domain. | **Multiple broadcast domains**, more control over network segmentation. |
| **Communication Scope** | Communicates **within the same network** (no IP routing). | Can communicate **within and across networks** using routing. |
| **Use Case** | Ideal for **local traffic management** within a network. | Commonly used to manage **VLANs** and larger networks with routing needs. |
| **Routing Capability** | **No routing** functionality, only works with MAC addresses. | Can perform **routing** between different IP subnets. |
| **Broadcast Traffic Handling** | All devices connected share the same broadcast traffic. | Can **separate broadcast traffic** between VLANs. |

**IPv4 (Internet Protocol Version 4)**

* **Overview**: IPv4 is the fourth version of the Internet Protocol and is widely used to identify devices on a network through an addressing system.
* **Address Format**: Uses a 32-bit address format, typically represented in decimal as four octets (e.g., 192.168.1.1).
* **Address Space**: Provides approximately 4.3 billion unique addresses, which has led to exhaustion due to the growth of the Internet.
* **Address Classes**:
* Class A: Large networks (0-127)
* Class B: Medium-sized networks (128-191)
* Class C: Small networks (192-223)
* Class D: Multicast (224-239)
* Class E: Experimental (240-255)
* **Subnetting**: Allows division of networks into smaller, manageable segments to improve efficiency and security.

**IPv6 (Internet Protocol Version 6)**

* **Overview**: IPv6 is the successor to IPv4, designed to address the limitations of IPv4, particularly address exhaustion.
* **Address Format**: Uses a 128-bit address format, represented in hexadecimal (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
* **Address Space**: Provides a vastly larger address space, allowing for approximately 340 undecillion addresses.
* **Features**:
* Simplified header format for improved processing efficiency.
* Built-in support for IPsec, enhancing security.
* Automatic address configuration, enabling devices to self-configure their addresses.
* **Transition Mechanisms**: Techniques like dual-stack, tunneling, and translation facilitate the gradual transition from IPv4 to IPv6.

**IP Multicasting**

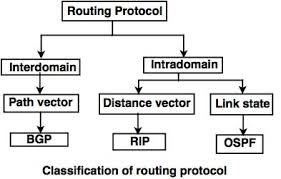
* **Definition**: IP Multicasting allows data to be sent from one sender to multiple receivers simultaneously, reducing bandwidth consumption compared to broadcasting.
* **Addressing**: Uses Class D addresses (224.0.0.0 to 239.255.255.255) to identify multicast groups.
* **Applications**: Commonly used for streaming media, online gaming, and teleconferencing.
* **Protocols**:
* **IGMP (Internet Group Management Protocol)**: Manages the membership of hosts in multicast groups.
* **PIM (Protocol Independent Multicast)**: Enables multicasting across various routing protocols.

**Principles of Routing**

* **Routing Definition**: The process of selecting paths in a network along which to send network traffic.
* **Static vs. Dynamic Routing**:
* **Static Routing**: Manual configuration of routes, useful for small networks with stable traffic patterns.
* **Dynamic Routing**: Automatically adjusts routes based on current network conditions, providing flexibility in larger, more complex networks.
* **Routing Tables**: Data structures that contain information about the paths to various network destinations.

**Routing Protocols**

* **Overview**: Protocols that facilitate the exchange of routing information between routers.
* **Common Protocols**:



**a. OSPF (Open Shortest Path First)**

* **Type**: Link-State Routing Protocol.
* **Purpose**: OSPF is used within large enterprise networks for **intra-domain routing**. It finds the shortest path to each destination using Dijkstra's shortest path algorithm.
* **Operation**:
  + OSPF routers maintain a **link-state database** (LSDB) that stores information about the network’s topology.
  + Routers use **Link-State Advertisements (LSAs)** to share information with other routers in the same OSPF area.
  + OSPF divides a large network into **areas**, reducing the overhead of routing information by limiting the scope of updates to each area.
* **Advantages**:
  + **Faster convergence** than RIP, as it updates immediately when the topology changes.
  + **Supports hierarchical design** with the use of areas.
  + **Scalable** for large networks.
* **Metrics**: OSPF uses **cost** (based on link bandwidth) as a metric to determine the best path.
* **Use Case**: Typically used in large enterprise networks where dynamic, efficient routing is necessary.

**b. RIP (Routing Information Protocol)**

* **Type**: Distance-Vector Routing Protocol.
* **Purpose**: RIP is one of the earliest routing protocols designed for **small, simple networks**. It helps routers share information about the network and compute the best routes.
* **Operation**:
  + RIP uses **hop count** as its only metric to determine the best path, with a maximum limit of **15 hops**. If the hop count exceeds 15, the network is considered unreachable.
  + Routers send the entire routing table to their neighbors every **30 seconds**.
* **Advantages**:
  + **Simple** to configure and easy to implement in small networks.
  + **Widely supported** by most routers.
* **Disadvantages**:
  + **Slow convergence** due to periodic updates every 30 seconds.
  + **Limited scalability** due to the 15-hop count limit.
  + **Suboptimal routing** because it does not consider metrics like bandwidth or delay.
* **Use Case**: Best suited for small networks with low complexity.

**c. BGP (Border Gateway Protocol)**

* **Type**: Path-Vector Protocol.
* **Purpose**: BGP is the protocol used to route traffic between different autonomous systems (ASes) on the **internet**, making it the backbone of internet routing.
* **Operation**:
  + BGP exchanges **prefixes** (networks) between autonomous systems, which allows routers to learn about IP ranges and their respective paths.
  + BGP routers use **path attributes** (such as AS path, next-hop, etc.) to determine the best route for data to travel.
  + Unlike OSPF and RIP, which are used within a single domain, BGP is used for **inter-domain routing** between multiple domains (such as ISPs).
* **Advantages**:
  + **Highly scalable** and suitable for very large networks (like the entire internet).
  + Provides **fine control** over routing decisions using multiple attributes.
* **Disadvantages**:
  + **Complex to configure** and maintain.
  + **Slower convergence** compared to OSPF and RIP.
* **Use Case**: Used by ISPs, large organizations, and network providers to control traffic routing between different parts of the internet.

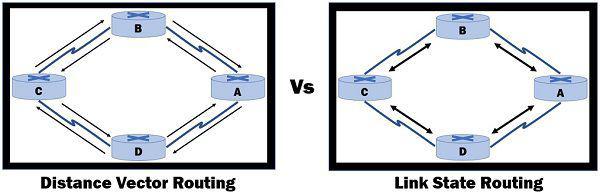
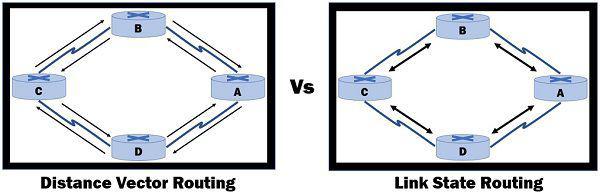
**Link-State Routing**

* **Definition**: A type of routing protocol where routers share information about their immediate neighbors and the state of their links.
* **Process**:
* Routers send link state advertisements (LSAs) to all other routers in the network.
* Each router constructs a complete view of the network topology.
* Uses algorithms like Dijkstra’s to compute the shortest path to each destination.
* **Examples**: OSPF and IS-IS (Intermediate System to Intermediate System).

**Distance Vector Routing**

* **Definition**: A type of routing protocol where routers send their routing table to their neighbors at regular intervals.
* **Process**:
* Each router maintains a table of the best known distance to each destination.
* Updates are based on the information received from neighboring routers.
* Routers use the Bellman-Ford algorithm to calculate the best paths.
* **Challenges**:
* Slow convergence and susceptibility to routing loops.
* Uses techniques like split horizon and poison reverse to mitigate these issues.
* **Examples**: RIP and EIGRP (Enhanced Interior Gateway Routing Protocol).

| **Basis** | **Link State Routing** | **Distance Vector Routing** |
| --- | --- | --- |
| **Bandwidth Usage** | Requires **more bandwidth** due to flooding of link-state packets and large-sized packets. | Requires **less bandwidth** as it only shares local updates and smaller packets. |
| **Knowledge** | Based on **global knowledge** of the network, each router has a complete view of the network topology. | Based on **local knowledge**, routers update routing tables using information from neighboring routers only. |
| **Algorithm Used** | Utilizes **Dijkstra’s Algorithm** to find the shortest path. | Utilizes **Bellman-Ford Algorithm** for routing decisions. |
| **Traffic** | Typically generates **more traffic** due to the need for regular link-state updates. | Generates **less traffic** as it only exchanges information with neighbors. |
| **Convergence Speed** | **Converges faster** because every router knows the entire network. | **Converges slowly**, especially in case of network changes, where bad news spreads slower (count-to-infinity issue). |
| **Looping** | **No persistent loops**; only transient loops may occur temporarily. | Has the risk of **persistent looping**, especially in the case of routing failures. |
| **Count-to-Infinity Problem** | **No count-to-infinity problem** due to full network knowledge. | Faces the **count-to-infinity problem**, especially in RIP (Routing Information Protocol). |
| **Implementation** | Commonly implemented in **OSPF (Open Shortest Path First)** and **IS-IS (Intermediate System to Intermediate System)** protocols. | Commonly implemented in **RIP (Routing Information Protocol)** and **IGRP (Interior Gateway Routing Protocol)**. |
| **Update Method** | Sends link-state updates to all routers in the network using **flooding**. | Sends routing updates only to **neighboring routers** at regular intervals. |
| **Suitability** | More suitable for **large, complex networks** where global network awareness is essential. | More suitable for **smaller networks** with less complexity. |



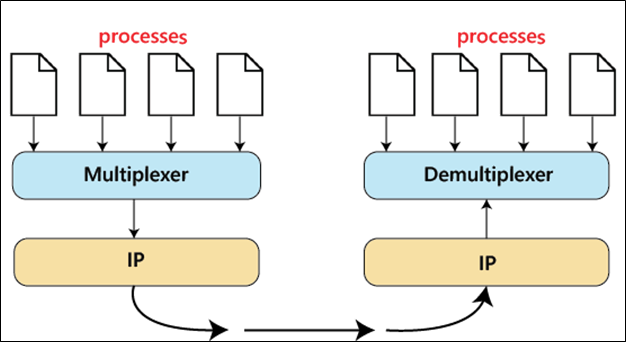
**UNIT - 5**

**Transport Layer in Computer Networks**

The transport layer is responsible for providing process-to-process communication services and ensuring reliable data transfer between applications. It lies above the network layer and deals with end-to-end communication.

1. **Process-to-Process Delivery**

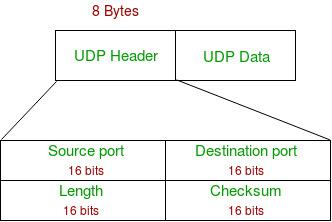
* The transport layer delivers data **from one specific process** (application) on a device to another specific process on a remote device.
* It uses port numbers to identify specific processes on a host.
* This enables communication between applications like a web browser and a web server, email clients, etc.
* Two main types of services are provided: **Connection-oriented (TCP)** and **Connectionless (UDP)**.



2. **UDP (User Datagram Protocol)**

* **Connectionless protocol**, meaning it does not establish a connection before sending data.
* **Best-effort delivery**, no guarantee of data delivery or order.
* Useful for applications requiring **fast transmission** without reliability (e.g., online gaming, live streaming, DNS, VoIP).
* **Key features:**
  + No connection setup.
  + No error recovery.
  + Smaller header (8 bytes), making it efficient for simple communication.
  + Suitable for applications where **low latency** is more important than reliability.
* **UDP Header**

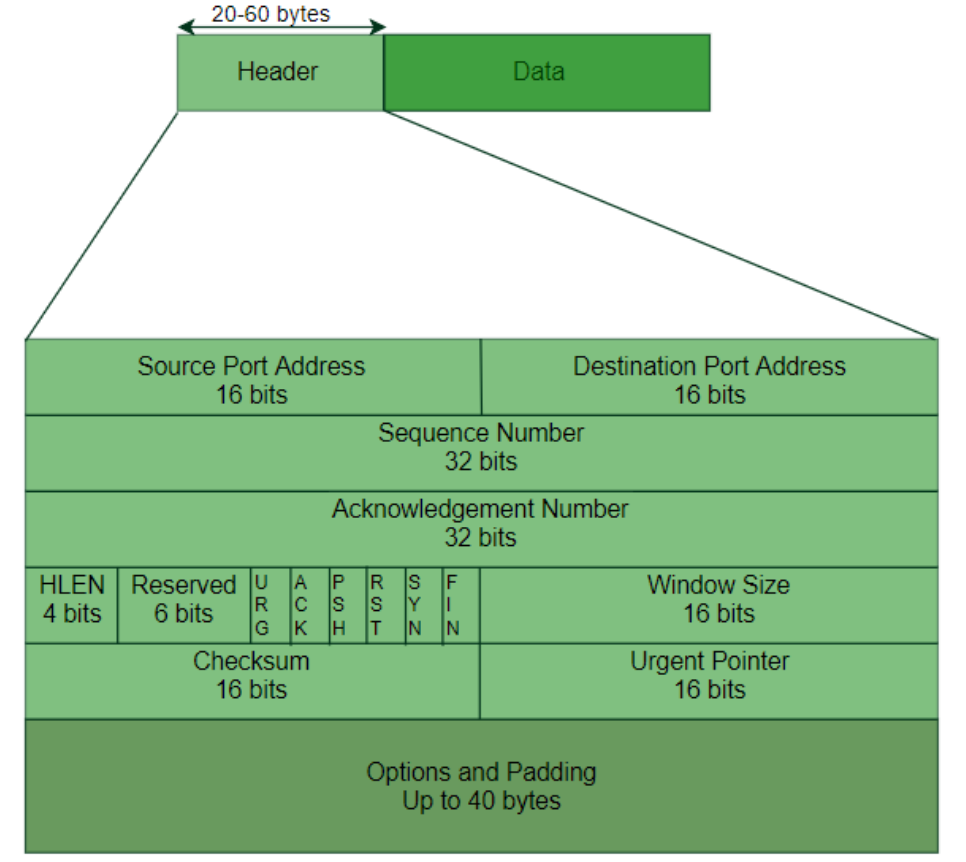
UDP header is an 8-byte fixed and simple header, while for TCP it may vary from 20 bytes to 60 bytes. The first 8 Bytes contain all necessary header information and the remaining part consists of data. UDP port number fields are each 16 bits long, therefore the range for port numbers is defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or processes.

****

* **Source Port:**Source Port is a 2 Byte long field used to identify the port number of the source**.**
* **Destination Port:**It is a 2 Byte long field used to identify the port of the destined packet.
* **Length:**Length is the length of UDP including the header and the data. It is a 16-bits field**.**
* **Checksum:**Checksum is 2 Bytes long field. It is the 16-bit one’s complement of the one’s complement sum of the UDP header, the pseudo-header of information from the IP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

3. **TCP (Transmission Control Protocol)**

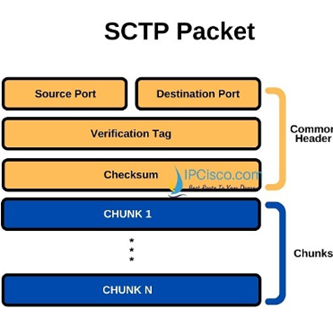
* **Connection-oriented protocol**, meaning it establishes a connection before data transmission.
* Provides **reliable communication** by ensuring data delivery, error recovery, and proper sequencing of packets.
* It handles **flow control** (adjusting the rate of data transmission) and **congestion control** (preventing network overload).
* **Key features:**
  + **Three-way handshake** to establish a connection.
  + **Error detection** and correction through acknowledgment.
  + **Flow control** using sliding windows.
  + **Congestion control** using algorithms like Slow Start, Congestion Avoidance, and Fast Recovery.
  + **Stream-oriented**, data is transmitted as a continuous stream rather than discrete messages.
* **TCP Segment Structure**
* A TCP segment consists of data bytes to be sent and a header that is added to the data by TCP as shown:



* The header of a TCP segment can range from 20-60 bytes. 40 bytes are for options. If there are no options, a header is 20 bytes else it can be of upmost 60 bytes. Header fields:
* **Source Port Address:**A 16-bit field that holds the port address of the application that is sending the data segment.
* **Destination Port Address:**A 16-bit field that holds the port address of the application in the host that is receiving the data segment.
* **Sequence Number:**A 32-bit field that holds the sequence number, i.e. the byte number of the first byte that is sent in that particular segment. It is used to reassemble the message at the receiving end of the segments that are received out of order.
* **Acknowledgement Number:**A 32-bit field that holds the acknowledgement number, i.e. the byte number that the receiver expects to receive next. It is an acknowledgement for the previous bytes being received successfully.
* **Header Length (HLEN):**This is a 4-bit field that indicates the length of the TCP header by a number of 4-byte words in the header, i.e. if the header is 20 bytes (min length of TCP header), then this field will hold 5 (because 5 x 4 = 20) and the maximum length: 60 bytes, then it’ll hold the value 15 (because 15 x 4 = 60). Hence, the value of this field is always between 5 and 15.
* **Control flags:**These are 6 1-bit control bits that control connection establishment, connection termination, connection abortion, flow control, mode of transfer etc. Their function is:
  + URG: Urgent pointer is valid
  + ACK: Acknowledgement number is valid (used in case of cumulative acknowledgement)
  + PSH: Request for push
  + RST: Reset the connection
  + SYN: Synchronize sequence numbers
  + FIN: Terminate the connection
* **Window size:**This field tells the window size of the sending TCP in bytes.
* **Checksum:**This field holds the checksum for error control. It is mandatory in TCP as opposed to UDP.
* **Urgent pointer:**This field (valid only if the URG control flag is set) is used to point to data that is urgently required that needs to reach the receiving process at the earliest. The value of this field is added to the sequence number to get the byte number of the last urgent byte.

| **Basis** | **TCP (Transmission Control Protocol)** | **UDP (User Datagram Protocol)** |
| --- | --- | --- |
| **Connection Type** | Connection-oriented (establishes a connection before data transfer). | Connectionless (no connection setup before data transfer). |
| **Reliability** | Reliable; ensures data is delivered and acknowledges receipt. | Unreliable; no guarantees for delivery or acknowledgment. |
| **Data Transmission** | Data is transmitted in a sequence; guarantees order of data packets. | Data packets can arrive out of order or may be lost. |
| **Overhead** | Higher overhead due to error checking, flow control, and connection setup. | Lower overhead, leading to faster transmission. |
| **Error Handling** | Provides error detection and correction mechanisms (e.g., retransmissions). | Basic error detection; no error correction. |
| **Flow Control** | Has built-in flow control using sliding windows to prevent data overload. | No flow control; data sent without regulation. |
| **Speed** | Slower compared to UDP because of reliability features. | Faster, as there’s no need to establish or maintain a connection. |
| **Use Cases** | Suitable for applications where **accuracy** is critical (e.g., file transfer, web browsing, email). | Suitable for applications where **speed** is critical (e.g., live streaming, online gaming). |
| **Header Size** | Larger header (**20-60 bytes**) due to control information. | Smaller header (**8 bytes**), making it lightweight. |
| **Examples** | HTTP, FTP, SMTP, Telnet. | DNS, VoIP, DHCP, Video/Audio streaming. |

4. **SCTP (Stream Control Transmission Protocol)**

* SCTP is a **transport layer protocol** that is **connection-oriented** and used for transmitting **multiple streams of data simultaneously** between any two endpoints that have established a connection in a computer network.
* Provides features of both TCP and UDP, but with additional capabilities like **multi-streaming** and **multi-homing**.
* **Key features:**
  + Allows transmission of multiple independent streams within the same connection.
  + Supports **multi-homing**, meaning a device can be identified by multiple IP addresses.
  + Provides **reliable, in-sequence transport** of data.
  + **Ideal for telecommunication applications**, such as signaling in mobile networks (SS7).
* **SCTP Packet**
  + SCTP protocol packet consist of two main parts - **Header** and **Payload**. The Header is common but Payload have variable chunks.
  + The Common SCTP header is 12 byte long and made of the 4 parts
  + **Source Port:** shows the sending port
  + **Destination Port:** shows the receiving port
  + **Verification tag:** a 32 bit random value which differentiate the packets from the previous connection
  + **Checksum:** a CRC32 algorithm for detection of error.

5. **Data Traffic**

* Refers to the amount of data being transmitted over a network at any given time.
* The transport layer protocols (TCP, UDP) are responsible for **efficient handling** of data traffic to avoid issues like congestion and packet loss.
* Data traffic is characterized by its **rate** (how much data is being sent per second) and **burstiness** (the variation in traffic rate).

6. **Congestion**

* Congestion occurs when the network is **overloaded with more data** than it can handle, leading to **packet loss**, **delays**, and **performance degradation**.
* Symptoms of congestion:
  + Long delays in packet delivery.
  + High packet loss rates.
  + Decreased throughput (effective data transmission rate).
* **Causes:**
  + Excessive data being injected into the network at high rates.
  + Insufficient bandwidth or network capacity.
  + Inefficient routing protocols.
  + Buffer overflow in routers and switches.

7. **Congestion Control**

* It refers to the methods used to prevent network overload and ensure smooth data flow. When too much data is sent through the network at once, it can cause delays and data loss.
* Congestion control techniques helps to manage the traffic, so all users can enjoy a stable and efficient network connection. These techniques are essential for maintaining the performance and reliability of modern networks.
* **Congestion Control Techniques:**
* **TCP Congestion Control:**
  + **Slow Start**: Gradually increases the transmission rate until congestion is detected.
  + **Congestion Avoidance**: Adjusts the sending rate based on network conditions to avoid congestion.
  + **Fast Retransmit and Fast Recovery**: Quickly retransmits lost packets without waiting for a timeout.
* **Queue Management Techniques:**
  + **RED (Random Early Detection)**: Drops packets early before the queue gets full to avoid congestion.
  + **Tail Drop**: Drops packets when the queue is full, which can lead to increased packet loss and retransmissions.
* **Traffic Shaping**: Controls the rate of outgoing traffic to prevent congestion (e.g., **Token Bucket Algorithm**).
* **Load Balancing**: Distributes traffic across multiple paths or servers to avoid overloading a single route.

**Application Layer in Computer Networks**

The application layer is the topmost layer in the OSI model and the TCP/IP model, providing the interface between user applications and the network services. It includes various protocols that help with specific tasks like web browsing, file transfer, and email.

1. **Principles of Internetworking**

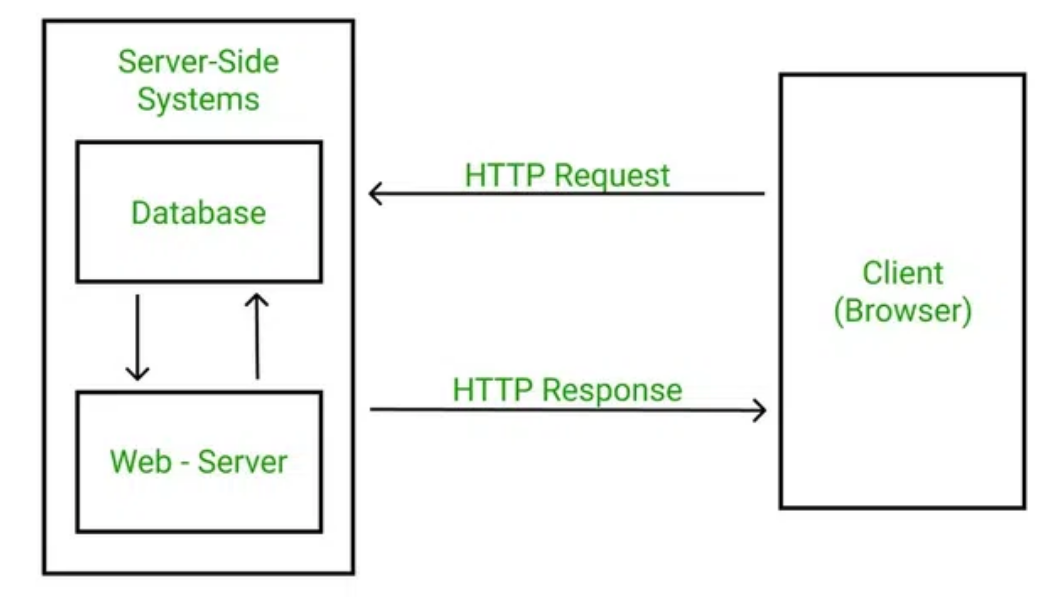
* Internetworking involves connecting multiple networks to form a larger network, allowing data exchange between devices.
* **Key concepts:**
  + **Routing**: Determines the path for data packets between different networks.
  + **Protocols**: Use standardized rules for communication (e.g., IP, TCP, UDP).
  + **Scalability**: Internetworks can handle a large number of devices and traffic.
  + **Addressing**: Every device has a unique IP address to ensure proper data delivery.
  + The **Internet** is the most prominent example of internetworking, which combines millions of smaller networks.

2. **Connectionless Internetworking**

* **Connectionless internetworking** is a communication method that allows data to be transferred between network endpoints **without establishing a dedicated connection** between the two devices. In this method, the sending endpoint does not try to ensure that the receiving endpoint is available to accept the data.
* **Here are some characteristics of connectionless internetworking:**
  + **Data packets:** The data is broken down into independent packets called **datagrams**. Each datagram has the address of the intended destination device.
  + **Network path:** Each datagram takes a **different path** through the network, depending on the decisions made by the routers.
  + **Data delivery:** The network **does not guarantee** that the data will be delivered from the host device to the destination device.
  + **Error correction:** Some protocols allow for error correction **by requesting a retransmission if necessary**.
  + Some examples of connectionless protocols used on the internet include HTTP, IP, UDP, ICMP, IPX, and TIPC.
* **Advantages:**
  + Lower overhead, as no connection setup is needed.
  + Faster transmission for applications that don’t require guaranteed delivery (e.g., VoIP, DNS queries).
* **Disadvantages:**
  + No error checking or recovery mechanisms.
  + Suitable for applications where speed is critical, and occasional packet loss is acceptable.

3. **HTTP (Hypertext Transfer Protocol)**

* HTTP is the protocol used for **transferring web pages** on the **World Wide Web** (WWW).
* **Stateless**: Each request/response pair is **independent**.
* Operates over **TCP**, usually on port **80** (or **443** for HTTPS).
* **Key concepts:**
  + **Request Methods**: GET (retrieve data), POST (send data), PUT (update data), DELETE (remove data).
  + **Headers**: Metadata about the request or response (e.g., content type, length).
* HTTPS (HTTP Secure) provides encrypted communication using **SSL/TLS**.

****

4. **WWW (World Wide Web)**

* The WWW is an information system of interlinked hypertext documents accessed via the internet.
* **Components:**
  + **Web pages**: Documents written in HTML, accessible via URLs.
  + **Browsers**: Software applications like Chrome or Firefox used to access web content.
  + **Hyperlinks**: Links that connect web pages.
* HTTP is the primary protocol enabling communication within the WWW.

5. **FTP (File Transfer Protocol)**

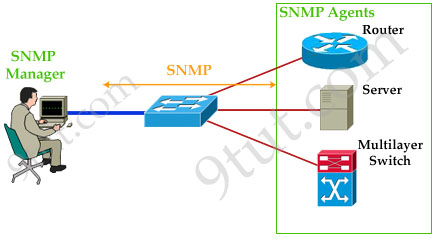
* FTP is used to transfer files between a client and a server on a network.
* Operates over **TCP** (typically ports **20** and **21**).
* Provides both **anonymous access** and **user authentication** for secure file transfers.
* **Modes:**
  + **Active Mode**: The **client** opens a port and waits for the server to establish a connection.
  + **Passive Mode**: The **server** opens a port and waits for the client to initiate the connection.
* FTP allows **directory navigation, uploading** and **downloading files**.

6. **SMTP (Simple Mail Transfer Protocol)**

* SMTP is a protocol for **sending email messages** from a client to a server or between servers.
* Works over **TCP**, usually on port **25**.
* SMTP is only responsible for sending emails (not retrieving).
* It uses a **queue-based system**, where messages are sent, and if the recipient is unavailable, the message is queued and retried later.
* Email clients typically use **POP3** or **IMAP** to **retrieve emails**.

7. **SNMP (Simple Network Management Protocol)**

* SNMP is used for **managing devices on IP networks** (e.g., routers, switches, servers).
* Operates over **UDP** (usually port **161**).
* Uses a **centralized management** **system** where a manager monitors and controls devices (called agents) using SNMP.
* **Components:**
  + **MIB (Management Information Base)**: A database with the hierarchical structure of variables that the manager can control.
  + **Trap messages**: Alerts sent by agents to the manager when certain conditions are met.
* Commonly used in **network monitoring and management systems**.



8. **POP3 (Post Office Protocol v3)**

* POP3 is used by **email clients to retrieve messages** from a mail server.
* Operates over **TCP** (usually on **port 110**).
* **Key features:**
  + Downloads emails from the server and usually deletes them from the server after retrieval.
  + **Does not support synchronization** between multiple devices (e.g., if an email is deleted on one device, it won’t reflect on others).
  + Simpler than **IMAP** but less flexible in terms of managing emails on the server.

9. **MIME (Multipurpose Internet Mail Extensions)**

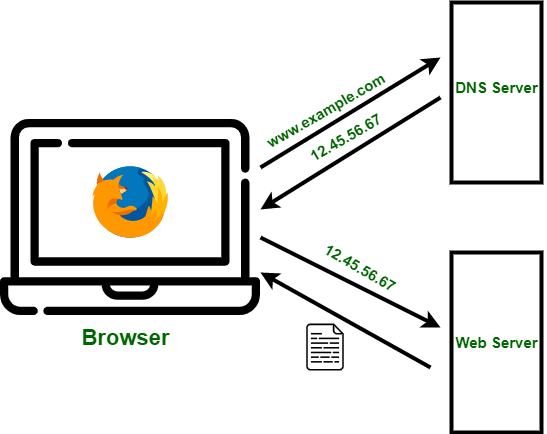
* MIME is an extension of email protocols like SMTP to allow the transfer of **multimedia content** (e.g., audio, video, images) within an email.
* It allows email messages to include:
  + Text in **character sets** other than ASCII.
  + Non-text attachments like images, videos, and application files.
  + Multipart message bodies.
  + Non-ASCII header information (e.g., non-English language characters).
* **Features of MIME –**
  + It is able to send multiple attachments with a single message.
  + Unlimited message length.
  + Binary attachments (executables, images, audio, or video files) may be divided if needed.
  + MIME provided support for varying content types and multi-part messages.

10. **Firewall**

* A firewall is a **network security device** that monitors and filters **incoming and outgoing network traffic** based on an organization’s previously established security policies. Firewalls can be either **software** or **hardware-based**.
* **Types of Firewalls:**
  + **Packet Filtering Firewall**: Filters traffic based on IP addresses, ports, and protocols.
  + **Stateful Inspection Firewall**: Monitors the state of active connections and makes decisions based on the context of the traffic.
  + **Application Layer Firewall**: Inspects traffic at the application layer to detect and block malicious content.
* Firewalls helps to prevent unauthorized access and protect the internal network from threats.

11. **DNS (Domain Name System)**

* DNS translates **human-readable domain names** (e.g., www.example.com) into IP addresses.
* Operates over **UDP** (port **53**) but can use **TCP** for larger responses.
* **Components:**
  + **DNS Resolver**: The client-side of DNS, responsible for querying DNS servers.
  + **DNS Server**: Responds with the appropriate IP address for a requested domain.
  + **Name Space**: A hierarchical structure of domain names, starting from the root (.), followed by top-level domains (TLDs) like .com, .org, etc.
* DNS enables easy access to websites without needing to remember numeric IP addresses.



12. **Gateways**

* A gateway is a **network device** that acts as a **translator** between two different protocols or networks.
* It is used when devices on two different networks need to communicate, even if they use different protocols.
* **Types of Gateways:**
  + **Network Gateway**: Connects different networks of the OSI network layer.
  + **Application Gateway (Proxy)**: Works at the application layer, translating between different protocols (e.g., HTTP to HTTPS).
* **Functions:**
  + Enables communication between different systems.
  + Provides protocol translation, IP address mapping, and other services needed for cross-network communication.