**OSI Model**

The **OSI (Open Systems Interconnection)** Model is a set of rules that explains how different computer systems communicate over a network. OSI Model was developed by the **International Organization for Standardization (ISO)**. The OSI Model consists of 7 layers and each layer has specific functions and responsibilities. This layered approach makes it easier for different devices and technologies to work together. OSI Model provides a clear structure for data transmission and managing network issues.

## **Layer 1 - Physical Layer**

The lowest layer of the OSI reference model is the **Physical Layer**. It is responsible for the actual physical connection between the devices. The physical layer contains information in the form of**bits.**Physical Layer is responsible for transmitting individual bits from one node to the next.

**Example** : Cables, Hubs, NICs

## **Layer 2 - Data Link Layer (DLL)**

The data link layer is responsible for the node-to-node delivery of the message. The main function of this layer is to make sure data transfer is error-free from one node to another, over the physical layer.

It Handles **framing**, **error detection**, and **correction also** Manages **MAC (Media Access Control)** addresses and controls access to the medium. The Data Link Layer is divided into **two sublayers**:

Logical Link Control (LLC), Media Access Control (MAC)

**Example**: Ethernet, PPP, ARP, Switches

## **Layer 3 - Network Layer**

The network layer works for the transmission of data from one host to the other located in different networks. It also takes care of packet routing i.e. selection of the shortest path to transmit the packet, from the number of routes available. The sender and receiver's IP[address](https://www.geeksforgeeks.org/what-is-an-ip-address) are placed in the header by the network layer. Segment in the Network layer is referred to as Packet**.**Network layer is implemented by networking devices such as [routers and switches](https://www.geeksforgeeks.org/difference-between-router-and-switch/).

**Example** : Routers, IP (IPv4/IPv6), ICMP, IGMP.

## **Layer 4 - Transport Layer**

The transport layer provides services to the application layer and takes services from the network layer. The data in the transport layer is referred to as **Segments**. It is responsible for the end-to-end delivery of the complete message. The transport layer also provides the acknowledgment of the successful data transmission and re-transmits the data if an error is found.

**Example**: Protocols used in Transport Layer are [**TCP**](https://www.geeksforgeeks.org/what-is-transmission-control-protocol-tcp/)**,**[**UDP**](https://www.geeksforgeeks.org/user-datagram-protocol-udp/)[**NetBIOS**](https://www.geeksforgeeks.org/what-is-netbios-enumeration/)**,**[**PPTP**](https://www.geeksforgeeks.org/pptp-full-form/)**.**

## **Layer 5 - Session Layer**

Session Layer in the OSI Model is responsible for the establishment of connections, management of connections, terminations of sessions between two devices. It also provides authentication and security as well as Synchronization and Dialog Control.

**Example:** Protocols used in the Session Layer are **NetBIOS, PPTP.**

## **Layer 6 - Presentation Layer**

The presentation layer is also called the**Translation layer**. The data from the application layer is extracted here and manipulated as per the required format to transmit over the network.

**Example:** Protocols used in the Presentation Layer are [JPEG](https://www.geeksforgeeks.org/difference-between-jpeg-and-png/), [MPEG](https://www.geeksforgeeks.org/mpeg-full-form/), [GIF](https://www.geeksforgeeks.org/what-is-a-gif-file/), [TLS/SSL](https://www.geeksforgeeks.org/difference-between-secure-socket-layer-ssl-and-transport-layer-security-tls/), etc.

## **Layer 7 - Application Layer**

The application layer in the OSI model is the topmost layer, serving as the interface between the user and the network. It provides the necessary protocols and services for end-user applications to access and interact with network resources, enabling actions like sending emails or browsing the web.

**Example:** Protocols used in the Application layer are [SMTP](https://www.geeksforgeeks.org/simple-mail-transfer-protocol-smtp/), [FTP](https://www.geeksforgeeks.org/file-transfer-protocol-ftp-in-application-layer/), [DNS](https://www.geeksforgeeks.org/domain-name-system-dns-in-application-layer/), etc.

**TCP/IP Model**

TCP/IP, standing for Transmission Control Protocol/Internet Protocol, is a suite of communication protocols that forms the foundation of the internet and most other networks. It defines how data should be packaged, addressed, transmitted, routed, and received. Think of it as the language and rules that allow different computers to communicate with each other. It has 4 layers

### 1. **Application Layer**

The Application Layer is the closest to the end user and is where applications and user interfaces reside. It serves as the bridge between user programs and the lower layers responsible for data transmission.

* **Function:** Provides services and interfaces for end-user applications to access network resources.
* **Key responsibilities:**
  + Supports application protocols like HTTP, FTP, SMTP, DNS, etc.
  + Enables communication between software applications across networks.
  + Handles data formatting, encryption, and session management.

### 2. **Transport Layer**

This layer ensures data is delivered reliably and in the correct order between devices. The two main protocols in this layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

* **Function:** Ensures reliable or unreliable delivery of data between hosts.
* **Key responsibilities:**
  + TCP (Transmission Control Protocol): Provides reliable, connection-oriented delivery with error checking, retransmission, and flow control.
  + UDP (User Datagram Protocol): Provides faster, connectionless transmission without guarantees.
  + Manages flow control and segmentation/reassembly of data.

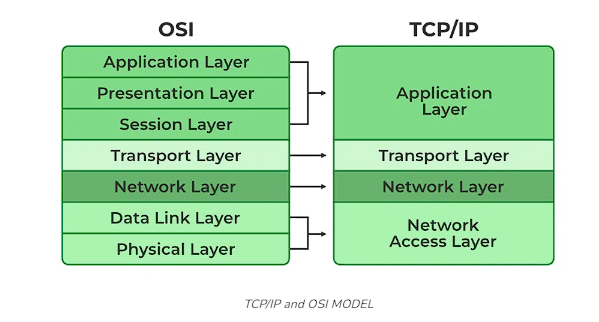
### 3. **Internet Layer**

It handles the routing of data packets across networks. It uses the Internet Protocol (IP) to assign unique IP addresses to devices and decide the most efficient path for data to reach its destination.

* **Function:** Determines the best path for data to travel across networks.
* **Key responsibilities:**
  + IP (Internet Protocol)**:** Provides addressing and routing.
  + Handles packet forwarding, fragmentation, and logical addressing (IP addresses).
  + Involves protocols like IP, [ICMP](https://www.geeksforgeeks.org/internet-control-message-protocol-icmp/)(for diagnostics), and [ARP](https://www.geeksforgeeks.org/arp-protocol/) (for address resolution).

### 4. **Network Access Layer**

This layer is the lowest layer in the model and responsible for the physical connection between devices within the same network segment.

* **Function:** Manages the physical transmission of data over the network hardware.
* **Key responsibilities:**
  + Handles how data is physically sent over cables, Wi-Fi, etc.
  + Manages MAC addressing, framing, and error detection at the physical link.
  + Includes Ethernet, Wi-Fi, and other data link technologies.

INTERNRT LAYER

|  |  |  |
| --- | --- | --- |
| **Parameters** | **OSI Model** | **TCP/IP Model** |
| **Full Form** | OSI stands for Open Systems Interconnection | TCP/IP stands for Transmission Control Protocol/Internet Protocol |
| **Layers** | It has 7 layers | It has 4 layers |
| **Usage** | It is low in usage | It is mostly used |
| **Approach** | It is vertically approached | It is horizontally approached |
| **Delivery** | Delivery of the package is guaranteed in OSI Model | Delivery of the package is not guaranteed in TCP/IP Model |
| **Replacement** | Replacement of tools and changes can easily be done in this model | Replacing the tools is not easy as it is in OSI Model |
| **Reliability** | It is less reliable than TCP/IP Model | It is more reliable than OSI Model |
| **Protocol Example** | Not tied to specific protocols, but examples include HTTP (Application), SSL/TLS (Presentation), TCP (Transport), IP (Network), Ethernet (Data Link) | HTTP, FTP, TCP, UDP, IP, Ethernet |
| **Error Handling** | Built into Data Link and Transport layers | Built into protocols like TCP |
| **Connection Orientation** | Both connection-oriented (TCP) and connectionless (UDP) protocols are covered at the Transport layer | TCP (connection-oriented), UDP (connectionless) |

|  |  |  |
| --- | --- | --- |
| **S** | **Point to point communication** | **Multipoint Communication** |
| **1.** | **Point to point communication means the channel is shared between two devices.** | **Multipoint Communication means the channel is shared among multiple devices or nodes.** |
| **2.** | **In this communication, there is dedicated link between two nodes.** | **In this communication, link is provided at all times for sharing the connection among nodes.** |
| **3.** | **In this communication, the entire capacity is reserved between these connected two devices with the possibility of waste of network bandwidth/ resources.** | **In this communication, the entire capacity isn’t reserved by any two nodes and the network bandwidth is maximum utilized.** |
| **4.** | **In this communication, there is one transmitter and one receiver.** | **In this communication, there is one transmitter and many receivers.** |
| **5.** | **In point-to-point connections, the smallest distance is most important to reach the receiver.** | **In Multi-point connections, the smallest distance is not important to reach the receiver.** |
| **6.** | **Point-to-point communication provides security and privacy because communication channel is not shared.** | **Multi-point communication does not provide security and privacy because communication channel is shared.** |
| **6.** |  |  |

## **Types of Network Topology**

### **1. Point-to-Point Topology**

Point-to-Point topology is the simplest network structure that connects two nodes directly using a single communication link. One device acts as a sender and the other as a receiver. This setup allows for a fast and direct exchange of data, providing high bandwidth. Since only two devices are involved, there is minimal chance of data collision or network congestion. An example would be a direct cable connection between a computer and a printer.

**Advantages:**

* Very simple and easy to establish.
* High-speed communication due to direct link.
* No congestion as only two devices are involved.

**Disadvantages:**

* Only suitable for a **two-device setup**.
* Not scalable for larger networks.

**2. Mesh Topology**

In Mesh topology, each device is connected to every other device in the network through dedicated links. This allows for fast and reliable communication, as data can travel on multiple paths. If one path fails, another is always available, making the network robust and secure. However, it's expensive and complex due to the number of cables and ports required.  
**Calculation:**

* **Number of links = N(N-1)/2**
* **Number of ports per device = N - 1**  
  For example, in a network with 6 devices:  
  Links = 6×5/2 = **15**  
  Ports per device = 6 - 1 = **5**

**Advantages:**

* Very reliable and fault-tolerant.
* High data security and privacy.
* Easy to find and fix faults.

**Disadvantages:**

* Expensive due to lots of cables.
* Complex installation and maintenance.

### **3. Star Topology**

In Star topology, all devices are connected to a central hub using separate cables. This central hub manages and controls all network traffic. If one cable fails, only the connected device is affected, not the whole network. However, if the hub fails, the entire network goes down.  
**Calculation:**

* **Number of cables = N** (each device connects to the hub)
* **Ports needed on the hub = N**  
  Star topology is commonly used in office LANs and wireless networks.

**Advantages:**

* Easy to set up and manage.
* Failure of one device doesn't affect the rest.
* Easy to identify faults.

**Disadvantages:**

* If the **hub fails**, the entire network goes down.
* Hub adds extra cost.

**4. Bus Topology**

Bus topology connects all devices to a single central cable called the backbone. Data travels in both directions, and only one device can transmit at a time to avoid collision. It is simple and cheap but not very reliable. If the backbone fails, the entire network is affected.  
**Calculation:**

* **Backbone cable = 1**
* **Drop lines = N** (one for each device)  
  This topology is used in small networks or cable TV systems.

**Advantages:**

* Simple and cheap to install.
* Requires less cable than mesh and star.

**Disadvantages:**

* If the **backbone fails**, the whole network fails.
* Difficult to identify faults.
* Performance degrades with heavy traffic.

**5. Ring Topology**

In Ring topology, each device is connected to exactly two other devices, forming a closed loop. Data travels in one direction (or both in dual ring topology) and passes through each device until it reaches the destination. It uses a method called token passing for data transmission. It is cost-effective and fast but prone to failure—if one device or link fails, the whole network might break.  
Ring topology is commonly used in metro or fiber networks.

**Advantages:**

* High-speed transmission.
* Minimal data collision.
* Cheap compared to mesh and star.

**Disadvantages:**

* Failure of one device affects the whole network.
* Difficult to troubleshoot.
* Adding/removing devices disturbs the whole setup.

### **6. Tree Topology**

Tree topology combines features of both Star and Bus topologies. Devices are grouped into star-configured networks connected to a central backbone cable, forming a hierarchical structure. It allows better network management and scalability. If the backbone or central hub fails, the network is affected. Tree topology is used in large organizational structures like corporate or educational institutions.

**Advantages:**

* Easy to expand and manage.
* Allows grouping of devices.
* Fault isolation is easier.

**Disadvantages:**

* If the central hub fails, the whole network fails.
* Expensive due to high cabling.

### **7. Hybrid Topology**

Hybrid topology is a combination of two or more different types of topologies, such as star, bus, mesh, or ring. It takes advantage of the strengths of multiple topologies, offering flexibility and scalability. However, it is complex and expensive to design and maintain. A real-world example would be a university campus where star topology is used in departments, bus topology in individual labs, and mesh or wireless topology for mobile devices.

**Advantages:**

* Very flexible and scalable.
* Can be customized for performance or cost.
* Fault tolerance and redundancy.

**Disadvantages:**

* Difficult to design and manage.
* High cost of devices and cables.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Feature** | **Point-to-Point** | **Mesh** | **Star** | **Bus** | **Ring** | **Tree** | **Hybrid** |
| **Structure** | Two nodes directly connected | Every device connects to all others | All devices connect to a central hub | All devices share a single backbone | Devices connected in a loop | Hierarchical star structure | Combination of two or more topologies |
| **Cabling Required** | Very low | Very high | Moderate | Low | Moderate | High | Depends on types combined |
| **Cost** | Very low | Very high | Moderate | Low | Moderate | High | High |
| **Scalability** | Not scalable | Difficult | Easy | Difficult | Moderate | Easy | Very easy |
| **Reliability** | High (simple) | Very high | Depends on central hub | Low (single point failure) | Low (loop break = failure) | Moderate (root hub is critical) | High (custom fault tolerance) |
| **Performance** | Excellent (for 2 nodes) | Excellent | Good | Poor as more nodes are added | Moderate | Good | Very good |
| **Failure Impact** | Both devices disconnected | Minimal | Entire network if hub fails | Entire network if backbone fails | Entire network if one node fails | Partial failure if upper nodes fail | Depends on combined topology |
| **Installation & Setup** | Easiest | Very complex | Easy | Easy | Moderate | Complex | Most complex |
| **Use Case Example** | PC to printer | Military, banking systems | Offices, LANs | Old Ethernet | Older LAN setups | Educational institutions | Large enterprise networks |

**What is the Bit Rate?**

Bit rate refers to the number of bits transmitted per second and is, therefore, a measure of the rapidity at which data is being transmitted over a communication channel. It is normally expressed in Kbps, Mbps, or Gbps. It will, therefore, give the relative efficiency of computer processing or handling data.

Bit Rate = Baud Rate × No. of Bits per Baud

**What is the Baud Rate?**

It is defined to be the number of signal changes or symbols sent per second over a communication channel. This decides the extent to which a transmission medium, such as a wire or a wireless spectrum, is capable of changing its state in one second. Every such change can represent one or more bits of data.

Baud Rate = Bit Rate / No. of Bits per Baud

|  |  |
| --- | --- |
| **Bit Rate** | **Baud Rate** |
| [Bit rate](https://www.geeksforgeeks.org/bit-rate/) is defined as the transmission of a number of bits per second. | [Baud rate](https://www.geeksforgeeks.org/baud-rate-and-its-importance/) is defined as the number of signal units per second. |
| Bit rate is also defined as per second travel number of bits. | Baud rate is also defined as per second number of changes in signal. |
| Bit rate emphasized computer efficiency. | While the baud rate emphasized data transmission. |
| The formula of Bit Rate is:  =Baud Rate × Number of Bits per Baud | The formula of Baud Rate is:  = Bit Rate / Number of Bits per Baud |
| Bit rate is not used to decide the requirement of bandwidth for transmission of the signal. | While baud rate is used to decide the requirement of bandwidth for transmission of the signal. |
| Bit Rate cannot determine the bandwidth. | Baud rate can determine the amount of bandwidth necessary to send the signal. |
| It counts the number of bits traveled per second such as Kbps, Mbps, Gbps, etc | It counts how many times the state of a signal is changing. |

**What is Data** :

Data refers to raw, unprocessed facts, figures, or symbols that can be used for reasoning, discussion, or calculation. It's the foundation of information and knowledge, often collected through observations, measurements, research, or analysis. Data can take various forms, including numbers, text, images, and sounds.

**What is Signal**

A "signal" generally refers to anything that conveys information, a message, or a command. It can be a physical action, sound, or even an electrical or electromagnetic current. In more technical contexts, like electronics or communications, a signal is often a time-varying voltage, current, or electromagnetic wave that carries data.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Data** | **Signal** |
| **Definition** | Data is raw information (facts, figures, instructions). | Signal is the transmission of data over a medium. |
| **Form** | Can be digital (0s and 1s) or analog (continuous values) | Can be analog (sine wave) or digital (square wave pulses) |
| **Purpose** | Represents information to be communicated or processed | Carries data from one place to another |
| **Storage** | Can be stored in memory, hard disks, etc. | Cannot be stored, only transmitted |
| **Nature** | Logical or abstract | Physical (electrical, light, radio waves) |
| **Example** | A text file, an image, a temperature value | Electrical waveform carrying voice through a wire |

**Low-Pass Channel:**

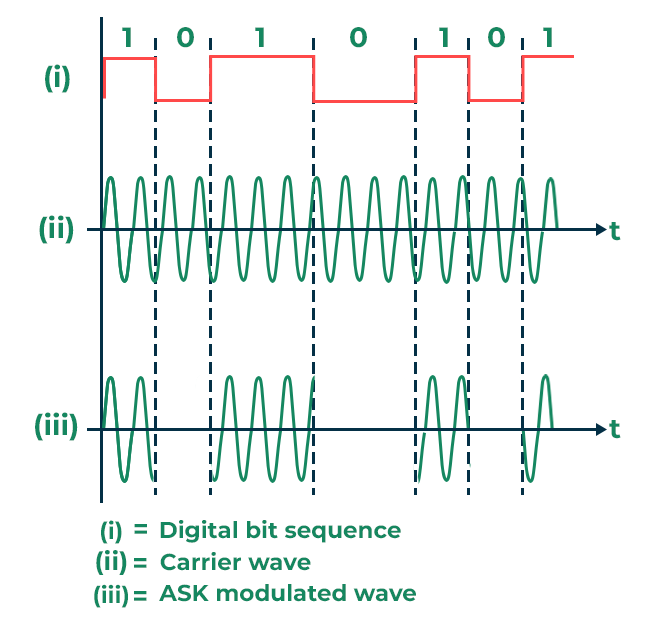
* **Definition:** It is a communication channel that transmits frequencies below a certain cut-off frequency. Basically, a low-pass channel is like a filter that lets low frequencies through.
* **Example:** A typical voice channel or a low-pass filter in an audio system.
* **Applications:** Often used for baseband transmission, where the digital signal is directly transmitted without modulation.

**Band-Pass Channel:**

* **Definition**: It is a communication channel that allows to transmits frequencies within a specific range between two cut-off frequencies. Basically, band-pass channel is like a window that lets through only a certain band of frequencies.
* **Example**: A radio channel or a band-pass filter in an audio system.
* **Applications**: Used for passband transmission, where the digital signal is modulated onto a carrier frequency before transmission.

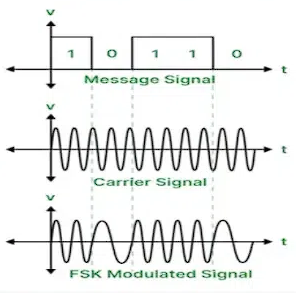
|  |  |  |
| --- | --- | --- |
| **Aspect** | **Low Pass Channel** | **Band Pass Channel** |
| **Definition** | A channel that allows **low-frequency signals** to pass and blocks high-frequency signals. | A channel that allows a **specific range of frequencies** to pass, blocking both low and high extremes. |
| **Frequency Range** | Passes from 0 Hz up to a certain cut-off frequency. | Passes signals between two frequencies (f₁ to f₂). |
| **Signal Type** | Typically used for **baseband signals** (starting from 0 Hz). | Typically used for **modulated signals** (not starting from 0 Hz). |
| **Examples** | Telephone line, Ethernet cables. | Radio communication, TV broadcasting, wireless communication. |
| **Use Case** | Data transmission over wires. | Wireless and RF communication. |
| **Channel Bandwidth** | Equal to cutoff frequency (e.g., 0 to 3 kHz → 3 kHz). | Bandwidth = f₂ - f₁ (e.g., 88 to 108 MHz → 20 MHz). |

## **Amplitude Shift Keying**

Amplitude Shift Keying (ASK) is a digital modulation technique . It transmits the digital information by varying the amplitude of a carrier signal. In ASK, a high-amplitude carrier signal is used to represent a binary '1,' and a low-amplitude carrier signal represents a binary '0.'

The carrier wave is represented by continuous signal. The carrier's amplitude changes between two levels, particularly '0' and '1'. When the input data is '0', the amplitude might be lower, and when the input data is '1', the amplitude could be higher. The changes in amplitude effects the carrier signal in accordance with the digital signal. This modulation technique is simple but can be susceptible to noise and interference, which can affect the accuracy of the received data.

**Frequency Shift Keying**

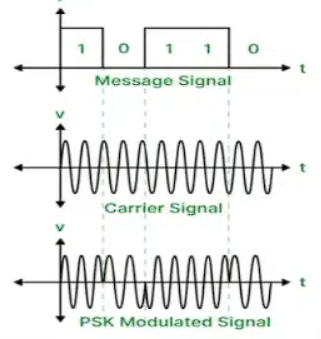
**Frequency Shift Keying (FSK)** is a digital modulation technique. It transmits digital information by **changing the frequency** of a carrier signal. In FSK, different **frequencies** are used to represent binary values.

A **high frequency** carrier wave is used to represent a binary **'1'**, and a **low frequency** carrier wave is used to represent a binary **'0'**.

The carrier wave continues without breaking, but its **frequency shifts** depending on the input data. When the input is '0', the carrier wave may have a lower frequency. When the input is '1', the carrier wave may switch to a higher frequency. This shift in frequency matches the digital data being sent.

FSK is **more reliable** than ASK in noisy environments because frequency changes are less affected by signal distortion, making it better for **wireless communication** and **modem signals.**

**Phase Shift Keying**

**Phase Shift Keying (PSK)** is a digital modulation technique. It transmits digital information by **changing the phase** of a carrier signal. In PSK, the **phase** of the carrier wave is shifted to represent binary data like **'0'** and **'1'**.

The carrier wave stays the same in terms of frequency and amplitude, but its **phase angle** is changed depending on the input bit. For example:

* A binary **'0'** might be represented by a **0° phase shift**.
* A binary **'1'** might be represented by a **180° phase shift**.

This means the wave flips its direction to indicate a different bit. In more advanced versions like **QPSK (Quadrature PSK)**, **2 bits** are sent at once using **4 different phase shifts** (0°, 90°, 180°, and 270°), which makes it faster.

PSK is **less affected by noise** than ASK, and it is commonly used in **Wi-Fi**, **satellite communication**, and **Bluetooth** systems because of its **efficiency and reliability**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **ASK (Amplitude Shift Keying)** | **FSK (Frequency Shift Keying)** | **PSK (Phase Shift Keying)** |
| **1. Basic Principle** | Information is in **amplitude** variations | Information is in **frequency** variations | Information is in **phase** variations |
| **2. Bandwidth Requirement** | Requires **less bandwidth** | Requires **more bandwidth** | Requires **less to moderate bandwidth** |
| **3. Noise Immunity** | **Poor** resistance to noise | **Better** resistance to noise | **Better** resistance to noise |
| **4. Synchronization** | **Not required** | **Not required** | **Essential** (needs phase sync between sender/receiver) |
| **5. DC Component Effect** | **High effect** of DC component | **Less effect** of DC component | **Less effect** of DC component |
| **6. Power Requirement** | Requires **more power** | Requires **moderate power** | Requires **less to moderate power** |
| **7. Bit Rate Applications** | Used in **low bit rate** applications | Used in **moderate bit rate** applications | Used in **high bit rate** applications |
| **8. Complexity of Implementation** | **Simple** implementation | **Moderately complex** implementation | **Very complex** implementation |

**Line Coding :**

* The binary such as 1 and 0 produced by a PCM encoder represented in various signalling formats for transmission over a channel.
* These signalling formats are known as line coding.
* In other words, we can say, line coding is the process that convert digital data to digital signal.

Line coding can be divided in 3 type Unipolar, Polar, Bipolar

**1**. **Unipolar Encoding**

* In unipolar encoding technique only two voltage level are used. It uses only one polarity of voltage level 0 and 1.
* Unfortunately, DC component present in the encoded signal and there is loss of synchronization for long sequence of 0 and 1. It is simple but not in use this days.

**Advantages:**

* Simple to design and understand
* Easy to implement

**Disadvantages:**

* Not good for synchronization (long strings of 0s or 1s)
* Has a DC component (not suitable for long distances)
* Poor noise immunity

### **2. Polar Encoding**

**In polar encoding**, two voltage levels are used. A binary **'1'** is represented by **+V** (like +5V), and a binary **'0'** is represented by **–V** (like -5V). There is no 0 voltage level. It’s better than unipolar for synchronization.

**Advantages:**

* Better synchronization than unipolar
* Reduces the average voltage level

**Disadvantages:**

* Still has a DC component
* Not suitable for all types of transmission media

### **3. Bipolar Encoding**

**In bipolar encoding**, three voltage levels are used: **+V**, **0V**, and **–V**. The binary **'0'** is always **0V**, while binary **'1'** alternates between **+V** and **–V**. This helps in reducing DC and improving error detection.

**Advantages:**

* No DC component (good for long-distance transmission)
* Better error detection
* Better synchronization

**Disadvantages:**

* More complex than unipolar and polar
* Still needs synchronization for long strings of 0s

Here's your content rewritten in **simple and clear paragraph form**, followed by **key points** for each layer, focusing on the **devices, their working OSI layer, and addressing**:

### 🔹 **1. Physical Layer**

The Physical layer is responsible for the actual physical connection between devices. It deals with transmission of raw bitstreams over a physical medium like wires, cables, or wireless. Devices here ensure that bits are physically transmitted from one device to another.

* **🛠 Devices:**
  + **Hub**: Connects multiple devices and sends data to all devices in the network. Works like a signal splitter.
  + **Cables**: Physical wires like coaxial, optical fiber, and twisted pair cables used to carry signals.
  + **Modem**: Converts digital signals to analog and back, enabling internet over telephone lines.
  + **Repeater**: Regenerates weak or lost signals to extend the range of the network.
* **📍 Layer**: OSI Layer 1
* **📮 Addressing Used**: **None (Raw bits only)**

### 🔹 **2. Data Link Layer**

This layer handles communication within the same local area network (LAN) using MAC addresses. It ensures error detection and frame synchronization.

* **🛠 Devices:**
  + **Bridge**: Connects two LANs that use the same protocol and filters traffic using MAC addresses.
  + **Switch**: Forwards data only to the specific device using MAC addresses. More efficient than a hub.
  + **NIC (Network Interface Card)**: A hardware component that connects a computer to a network and provides a MAC address for communication.
* **📍 Layer**: OSI Layer 2
* **📮 Addressing Used**: **MAC Address**

### 🔹 **3. Network Layer**

The Network layer is responsible for routing data from one network to another using IP addresses. It decides the best path for the data to travel.

* **🛠 Devices:**
  + **Router**: Connects different networks and forwards data based on IP addresses.
  + **Brouter (Bridge + Router)**: Works both as a bridge (for non-routable protocols) and a router (for routable ones).
* **📍 Layer**: OSI Layer 3
* **📮 Addressing Used**: **IP Address**

### 🔹 **4. Transport Layer**

This layer ensures reliable end-to-end communication between devices or applications. It handles segmentation, error control, and flow control.

* **🛠 Devices:**
  + **Gateway**: Translates protocols between different networks, used in communication between dissimilar systems.
  + **Firewall**: Controls data flow for security. Filters data packets and can act as a proxy server.
* **📍 Layer**: OSI Layer 4
* **📮 Addressing Used**: **Port Numbers**

### 🔹 **5. Application Layer**

The Application layer is the closest to the user and provides network services directly to applications like web browsers, email, or messaging apps.

* **🛠 Devices:**
  + **PCs, Smartphones, Servers**: Devices that run applications and interact directly with the network.
* **Gateway & Firewall**: Still play roles here to support and protect application-level communication.
* **📍 Layer**: OSI Layer 7
* **📮 Addressing Used**: **URL, IP, or Hostnames (depending on application)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Device** | **Working OSI Layer** | **Address Used** | **Purpose** |
| Hub | Layer 1 – Physical | None | Broadcasts signal to all ports |
| Cable | Layer 1 – Physical | None | Physical medium for data transmission |
| Modem | Layer 1 – Physical | None | Converts digital ↔ analog signals |
| Repeater | Layer 1 – Physical | None | Regenerates signals to boost strength |
| NIC | Layer 2 – Data Link | MAC Address | Connects computer to network using MAC |
| Bridge | Layer 2 – Data Link | MAC Address | Connects and filters between LAN segments |
| Switch | Layer 2 – Data Link | MAC Address | Sends data to specific device within LAN |
| Router | Layer 3 – Network | IP Address | Routes data between different networks |
| Brouter | Layer 2 & 3 – DL + Network | MAC & IP Address | Combines bridge and router functions |
| Firewall | Layer 4 – Transport | IP, Port | Protects network by filtering data packets |
| Gateway | Layer 4+ – Transport & Above | Depends on Protocol | Translates between different network protocols |
| PC/Phone/Server | Layer 7 – Application | URL, IP, Hostname | User devices running networked applications |

|  |  |
| --- | --- |
| **Circuit Switching** | **Packet Switching** |
| Circuit switching has 3 phases:  i) Connection Establishment.  ii) Data Transfer.  iii) Connection Released. | In Packet switching directly data transfer takes place. |
| In circuit switching, each data unit knows the entire path address which is provided by the source. | In Packet switching, each data unit just knows the final destination address intermediate path is decided by the routers. |
| In Circuit switching, data is processed at the source system only | In Packet switching, data is processed at all intermediate nodes including the source system. |
| The delay between data units in circuit switching is uniform. | The delay between data units in packet switching is not uniform. |
| Resource reservation is the feature of circuit switching because the path is fixed for data transmission. | There is no resource reservation because bandwidth is shared among users. |
| Circuit switching is more reliable. | Packet switching is less reliable. |
| Wastage of resources is more in Circuit Switching | Less wastage of resources as compared to Circuit Switching |
| It is not a store and forward technique. | It is a store and forward technique. |
| Transmission of the data is done by the source. | Transmission of the data is done not only by the source but also by the intermediate routers. |
| Congestion can occur during the connection establishment phase because there might be a case where a request is being made for a channel but the channel is already occupied. | Congestion can occur during the data transfer phase, a large number of packets comes in no time. |
| Circuit switching is not convenient for handling bilateral traffic. | Packet switching is suitable for handling bilateral traffic. |
| In Circuit switching, the charge depends on time and distance and not on traffic in the network. | In Packet switching, the charge is based on the number of bytes and connection time. |
| Recording of packets is never possible in circuit switching. | Recording of packets is possible in packet switching. |
| In Circuit Switching there is a physical path between the source and the destination | In Packet Switching there is no physical path between the source and the destination |
| Circuit Switching does not support store and forward transmission | Packet Switching supports store and forward transmission |
| Call setup is required in circuit switching. | No call setup is required in packet switching. |
| In circuit switching each packet follows the same route. | In packet switching packets can follow any route. |
| The circuit switching network is implemented at the physical layer. | Packet switching is implemented at the datalink layer and network layer |
| Circuit switching requires simple protocols for delivery. | Packet switching requires complex protocols for delivery. |

**Similarities between Datagram Switching and Virtual Circuit Switching:**

* Both are **packet-switching** technologies — they transmit data by dividing it into smaller packets.
* Both can **handle multiple transmissions simultaneously**, enabling efficient network usage.
* Both can support **connection-oriented and connectionless communication**, depending on the design.
* Both provide **error correction mechanisms** such as checksums or parity checks.
* Both can work with **variable-length packets**, adapting to different data sizes.
* Both are capable of **supporting Quality of Service (QoS)** to manage traffic priority.
* Both support **fragmentation**, allowing large data to be split into smaller parts for easier transfer.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Datagram Switching** | **Virtual Circuit Switching** |
| **Connection Type** | Connectionless | Connection-oriented |
| **Path Selection** | Each packet uses a separate path | All packets follow the same path |
| **Packet Order** | Packets may arrive out of order | Packets arrive in order |
| **Header Requirement** | Each packet contains full header | Only the first packet has full header |
| **Reliability** | Less reliable | Highly reliable |
| **Efficiency vs Delay** | High efficiency, more delay | Lower efficiency, less delay |
| **Implementation Cost** | Easier and cost-efficient | Expensive and complex |
| **Path Setup** | No path setup required | Path setup required before transmission |
| **Use Case** | Widely used in the Internet | Used in X.25, ATM |

**1. Data Link Layer Responsibilities**

* Provides **node-to-node** data transfer (within the same network).
* Performs **framing** to divide data into manageable units.
* Uses **MAC addressing** to identify devices in a local network.
* Ensures **error detection and correction** using methods like CRC.
* Implements **flow control** to prevent congestion.
* Manages **media access control** on shared communication channels.
* Ensures **frame synchronization** between sender and receiver.

**2. Network Layer Responsibilities**

* Provides **host-to-host communication** across different networks.
* Assigns and manages **logical addressing** (IP addresses).
* Handles **routing** to determine the best path for data.
* Responsible for **packet forwarding** through intermediate routers.
* Performs **fragmentation and reassembly** of large packets.
* Supports **error handling and diagnostics** (e.g., ICMP).

**3. Transport Layer Responsibilities**

* Provides **process-to-process communication**.
* Performs **segmentation and reassembly** of application data.
* Uses **port numbers** to identify applications.
* Ensures **reliable data transfer** (especially using TCP).
* Manages **flow control** (e.g., sliding window in TCP).
* Handles **error detection and correction**.
* Manages **connection setup, maintenance, and termination**.
* **Transmission delay, propagation delay, throughput and efficiency in stop and wait**

**🔹 1. Transmission Delay (Td):**

* **Definition**:  
  Transmission delay is the amount of time required to push all bits of a data frame from the sender into the transmission medium. It depends on the size of the frame and the bandwidth of the link.  
  A higher bandwidth or smaller frame size results in lower transmission delay.
* **Formula**:

**🔹 2. Propagation Delay (Tp):**

* **Definition**:  
  Propagation delay is the time it takes for a signal or a bit to travel from the sender to the receiver through the physical medium (like wire or fiber). It depends on the **distance** between the sender and receiver and the **propagation speed** of the medium.
* **Formula**:

**🔹 3. Throughput:**

* **Definition**:  
  Throughput refers to the actual amount of useful data successfully transmitted over a network per unit of time. In the **Stop-and-Wait ARQ** protocol, the sender must wait for an acknowledgment before sending the next frame, which reduces the throughput when delays are high.
* **Formula (Stop-and-Wait)**:

**🔹 4. Efficiency:**

* **Definition**:  
  Efficiency is the ratio of the time used for actual data transmission to the total time taken including waiting for acknowledgment. In Stop-and-Wait, since the sender waits after each frame, the efficiency is lower, especially when the propagation delay is large compared to transmission time.
* **Formula**: