**SDLC:**

The Software Development Life Cycle (SDLC) is a structured approach used by development teams to manage the software development process, ensuring high-quality software that meets customer needs. It provides a framework for systematically planning, designing, building, testing, deploying, and maintaining software.

SDLC is different from STLC which is development of a software product which involves requirement, design coding, testing and deployment.

**Why SDLC:**

SDLC is helpful to manage and to ensure successful implantation which follows the step-by-step procedure, it may also improve transparency. It’s simply a planning of groundwork to meet the requirements/expectations and minimize the risks, it also ensures security

**Stages in SDLC:**

1. **Planning:**

SDLC is a plan to evaluate and discuss among the leadership and client to set the project goals and to make clear understanding what needs to develop.

1. **Design:**

Based on planning, need to analyze the specifications to design (high/low project architecture) with existing features and user requirements to make the software easy to use.

1. **Development:**

Development is a phase where design/development the product with requirements, developers involved in this phase which includes code written in specific programming language.

1. **Testing:**

In the testing phase testers ensure to test and identify the bugs in developed project to make defect and bug free project.

1. **Deployment**

Finally, the project can deploy to end user environments like testing, pre-prod and prod.

1. **Maintenance:**

In this stage it should be maintained to be functioning without issues which includes upgrading and ensuring optimum performance.

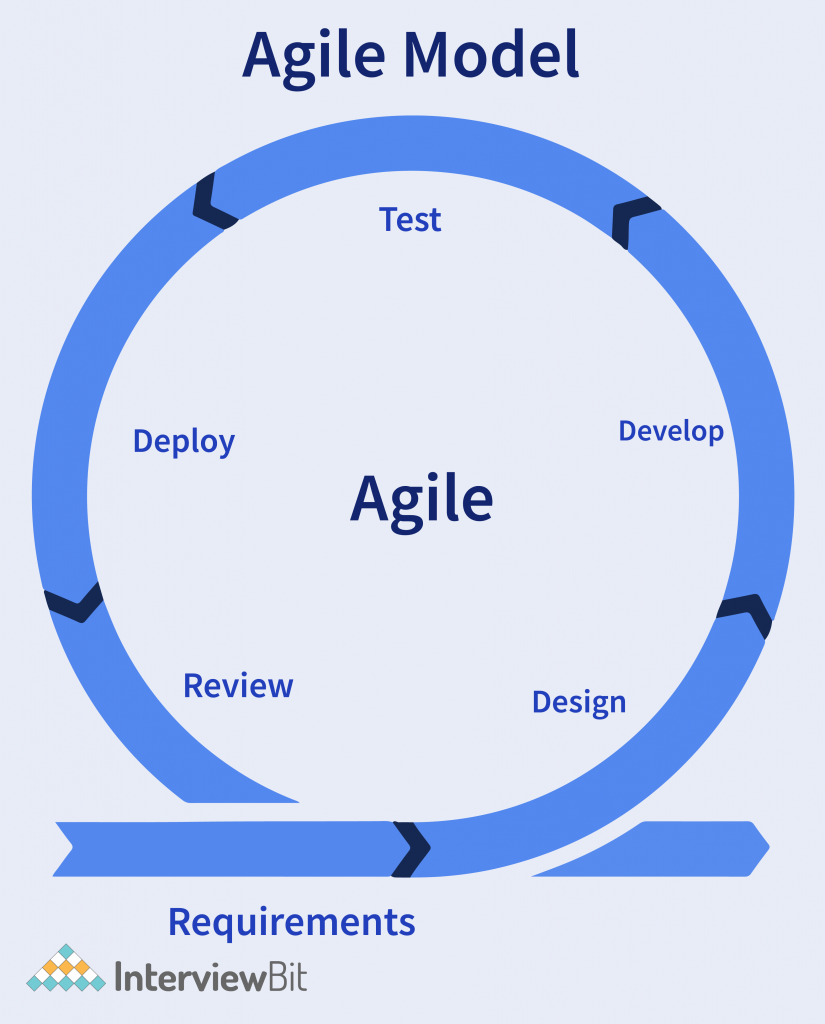
**Models in SDLC:**

1. **Agile Model:**

In Agile model, the software is divided in to small parts and then proceed with developing based on small features: which needs frequent updates based on customer feedback.

Disadvantages:

* Not ideal for fixed scope.
* Continuous user involvement can lead to frequent changes.



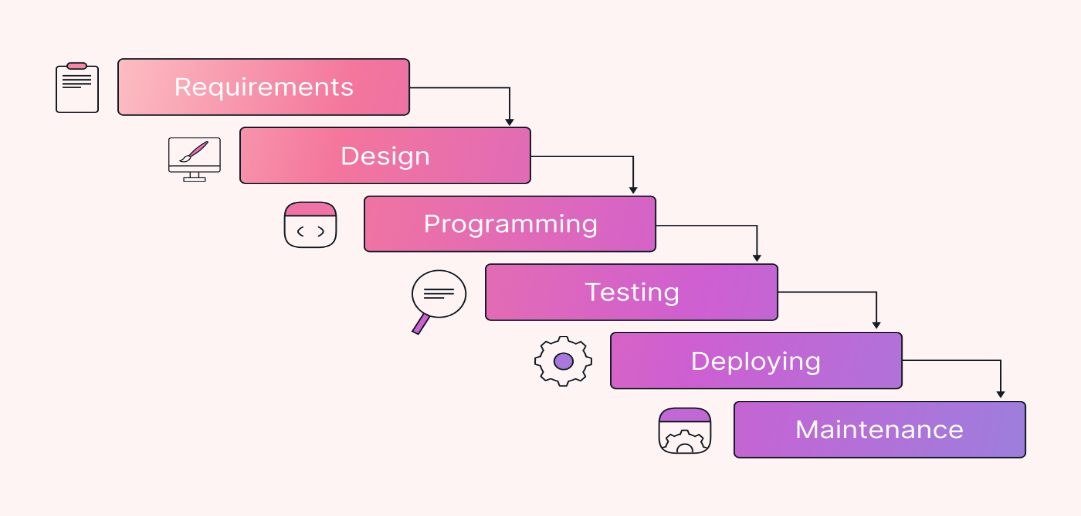
1. **Waterfall Model:**

It is sequential model where each phase completes before the next begins where output of stage is input of next stage but cannot go to the previous stage.

Application: Alarms, Traffic Systems

Advantage: Simple and easy to use

Disadvantage Very hard to go back to earlier stage.



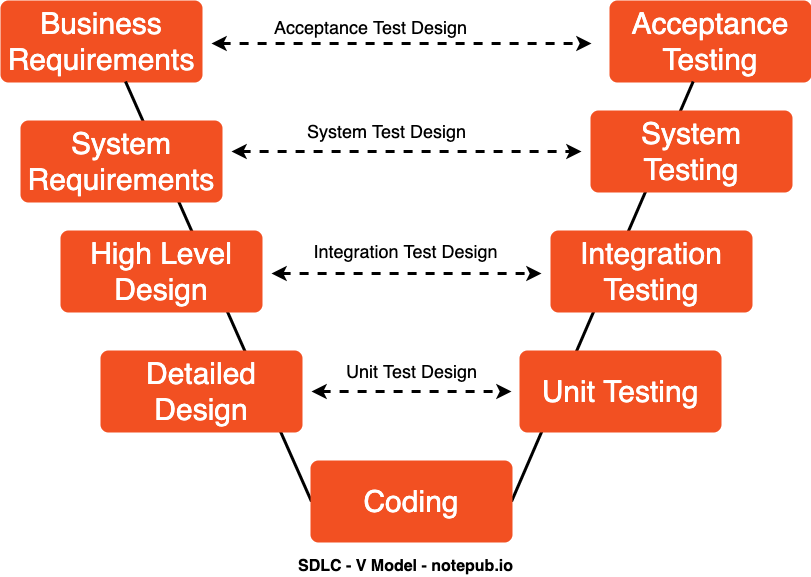
1. **V-Model:**

V-Model is associated with corresponding testing stage which includes verification and validation. It ensures quality.

Application: Automotive control systems.

Advantages: Predictions

Disadvantage: Not easy to go back previous stage and not suitable for long term.

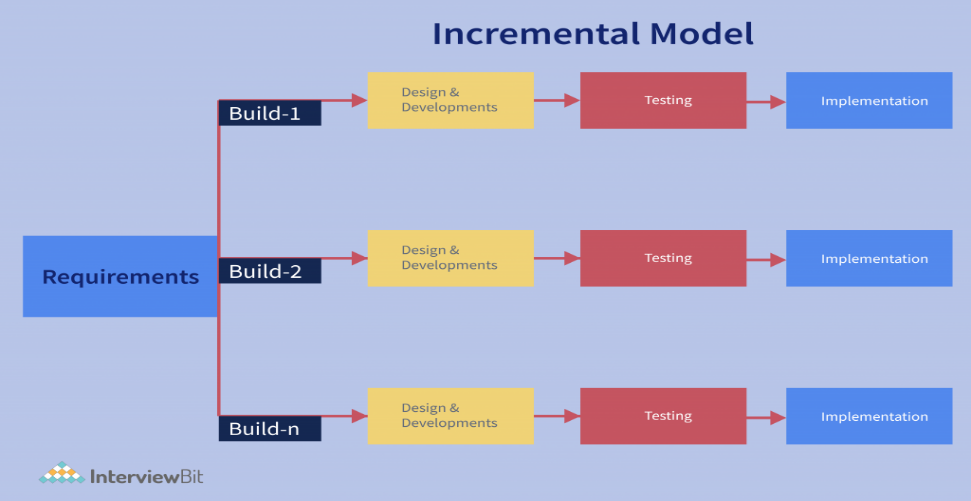


1. **Incremental model:**

Incremental model divides complicated to sequential parts and focuses on incremental improvements.

Application: Real time traffic systems, monitoring systems in medical fields.

Disadvantages: Changes in early model may affects another stage.

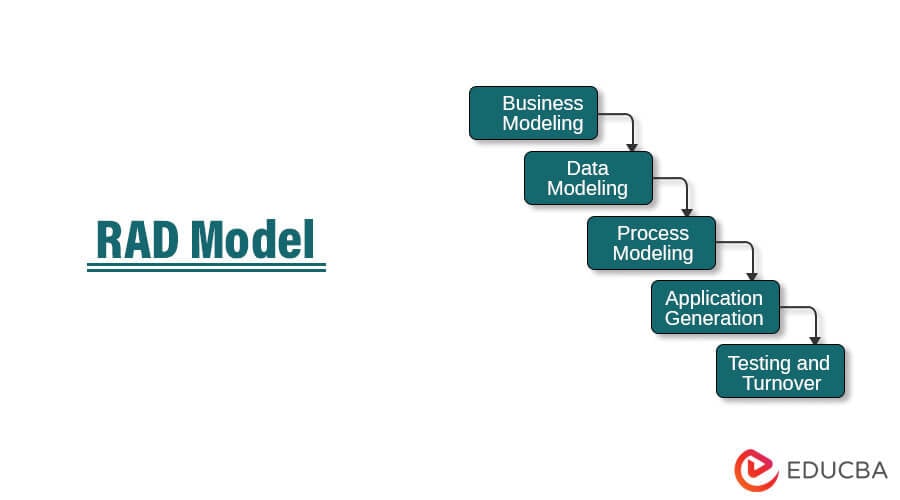


1. **RAD Model:**

Rapid Application Development (RAD) model prioritizes the prototyping and gives quick feedback to quickly build high quality. It is a user centered development

Application: Real time order system.

Disadvantages: Not suitable for large/complex system



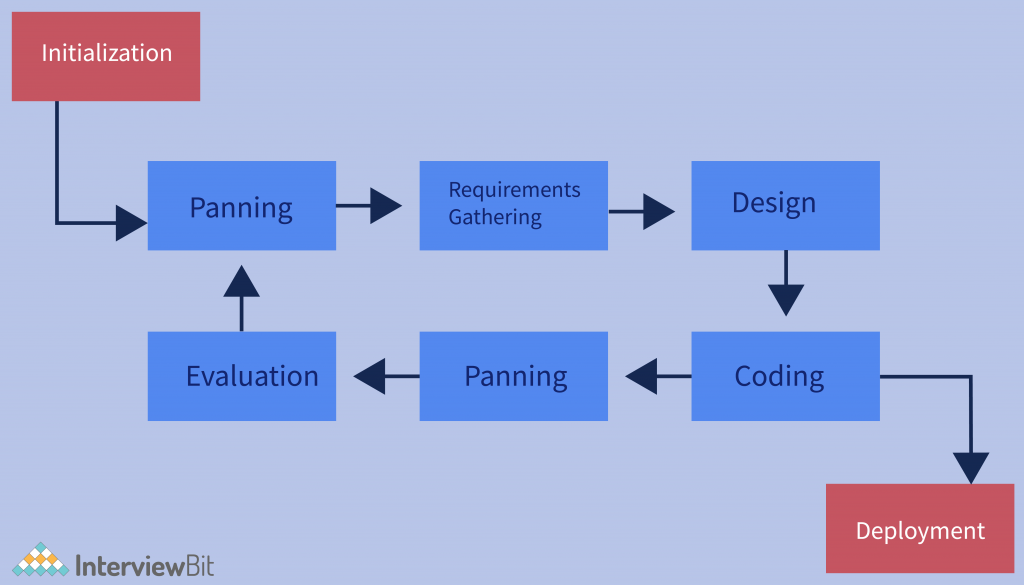
1. **Iterative Model:**

It is tested in multiple cycles with each iterations additional features may design, develop to add the functionality. It is easier to manage risk.

Application: New website creation.

Advantages: Identify the issues in early stages.

Disadvantage: Continuous changes to requirements may cause loss of focus.



1. **Spiral Model:**

It goes through improvement in each phase which is used in high risk systems which includes iterative and waterfall models.

Application: Game Development

Disadvantage: Large scale projects where timing and dependencies are critical, lengthy process



1. **Prototype Model:**

Its preliminary version of product with user’s feedback. The development continuously improves the product design through user testing.

Application: Facebook, YouTube, Amazon

Advantages: Early Error Detection, Reduced Risk of Project Failure:

Disadvantages: Time-Consuming, Insufficient Analysis, Costly.



**Scrum:**

* Scrum is a frame work to implement agile methodology which is focused on short term iterations (sprints).
* Scrum ensures to the team to manage product vision and backlogs to fix bugs.
* Clear on requirements.
* It may include standup calls, sprint planning, identify improvements.

Advantages: Projects with changing requirements/which needs continuous improvements.

**Sprint:**

It’s a time/fixed duration which is consistent across all teams which focuses on common goal. Sprint required planning, daily meetings, work, review, improvement

**Do’s:**

* Collaborating continuously
* Attend daily scrum calls
* Testing frequently in parallel with development.
* Identifying improvements and fixing bugs
* Updating task boards on time.

**Don’ts:**

* Don’t skip daily status calls
* Don’t to add new things in middle of sprint,
* Testing should be done in parallel
* Raise blockers immediately.

**Product Backlogs:**

Product backlog has to do lists which is owned by product owner. Product backlog has stories short and simple description of a feature.

**Sprint backlog**:

It’s a selected user stories/subtasks to do list owned by development team. which commits to completing in the sprint like what we could do or list.

* **Epic:** A large story that reduced to smaller stories.
* **Task:** Task includes technical steps to complete the story.
* **Acceptance:** Conditions the story must meet the requirements.
* **Increment:** Sum of all completed work
* **Latency requirements:** Maintaining response time
* Real time testing and verify real time behavior
* Continuous integration with timing.
* Timing constraints.

**Networks:**

Ports are virtual end points of a system, where all network connected devices are associated with protocols (simply numeric id)/ to route data.

**Protocols:**

Protocols are set of rules that govern data between devices like how data should exchange which ensure reliable communication/how data is transmitted

**Different network types:**

* PAN: Personal Area Network (few meters)
* LAN: Local Area Network (small campus)
* WLAN: Wireless LAN (wifi)
* MAN: Metropolitan Area Network (city wide)
* WAN: Wide Area Network (global)
* SAN: Storage Area Network (high speed data storage with backup)
* VPN: Virtual Private Network
* CAN: Campus area network

**Server:**

* It’s a system that provides services/resources over a network.
* Web server: Which hosts websites and web services- HTTP/HTTPS
* File Server: Stores and manages files for backups/storage backup- FTP
* Database Server: To store and manage database-MySQL, Oracle DB
* Mail Server: Sends/receives mails- SMTP
* Application server: Hosts and runs application

**Domain Name System**:

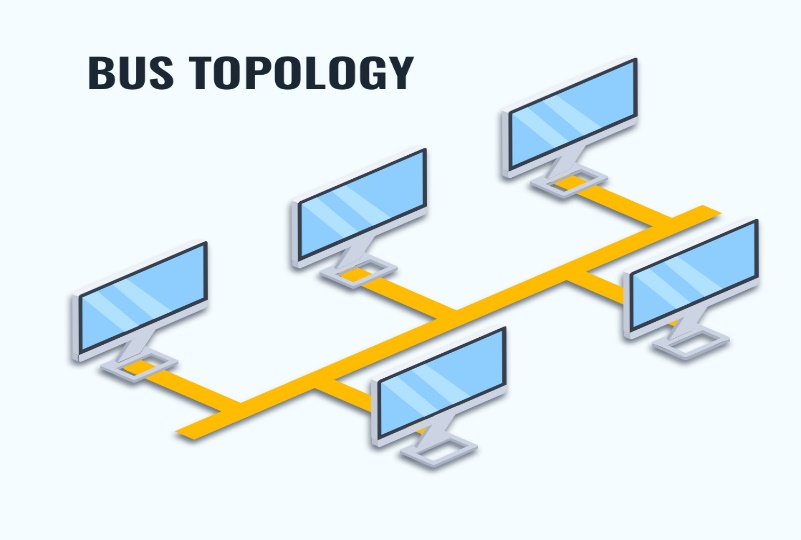
It is like to translate human friendly domain names instead of complex IP Address which can easily change host without domain name

**Topologies** **in network:**

Network topology refers to layout or interconnection of devices.

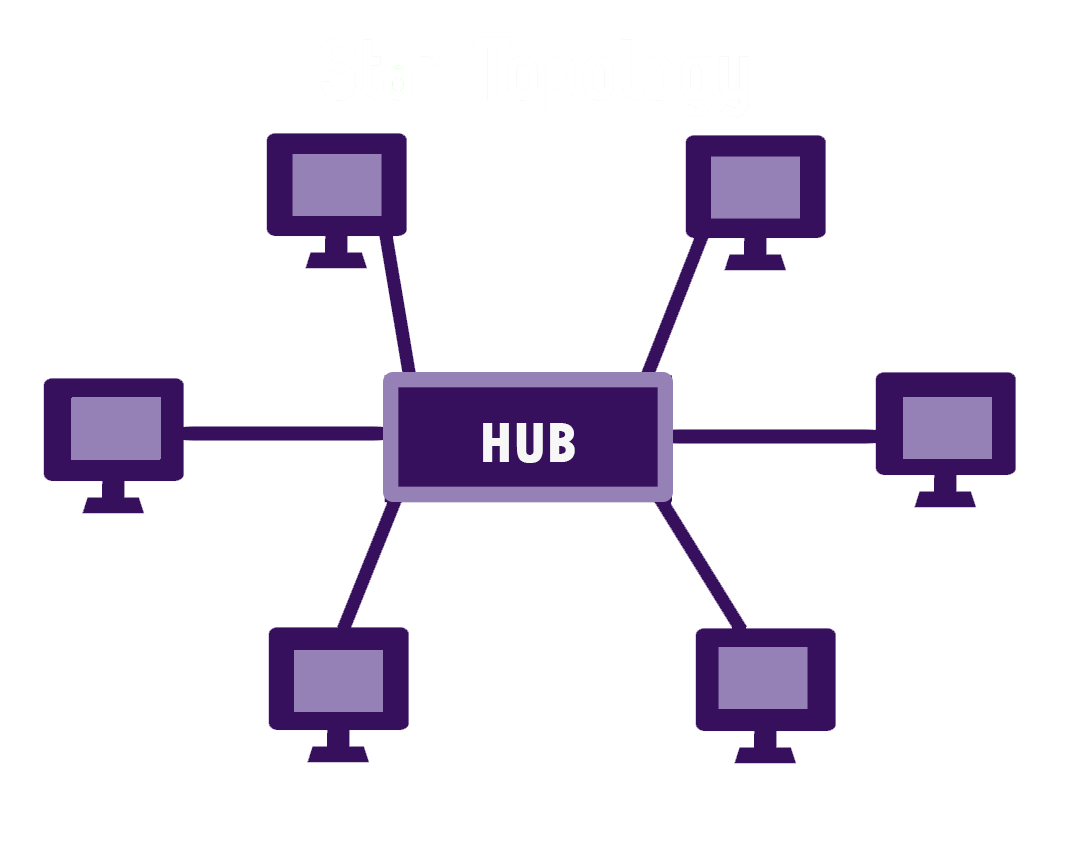
1. **Bus Topology:**

All devices in single communication line which is simple and cheap but if single break in cable interrupts the connection.



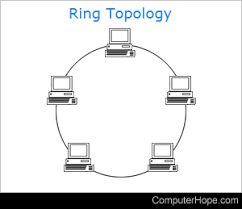
1. **Star Topology:**

All nodes connected to central device switch/hub.



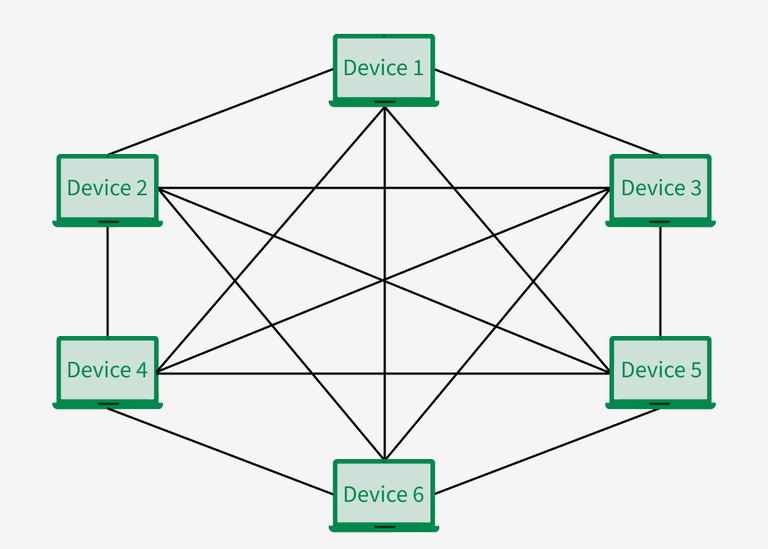
1. **Ring topology**

Each device connected to other forming circle, data travels in both directions-failure in one link cause total network down.



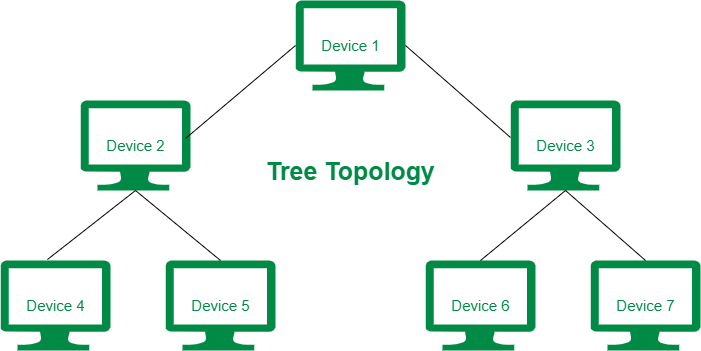
1. **Mesh Topology:**

Every device connects to other, expensive and complex-fault tolerant.



1. **Tree Topology:**

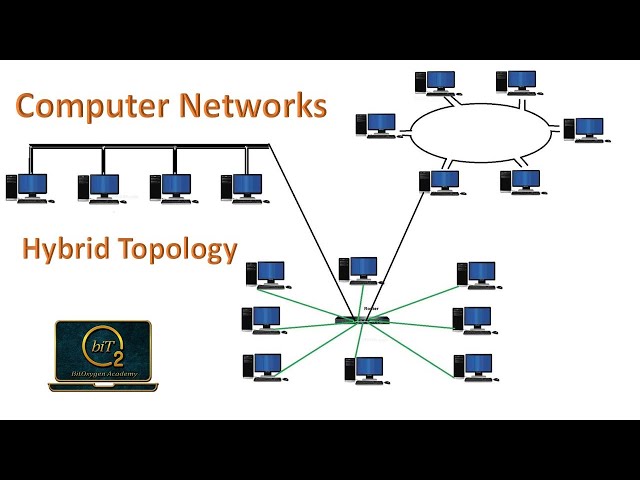
A combination of star bus which is used in large networks



1. **Hybrid Topology:**

It’s a mix of two or more topologies which offers flexibility and efficiency depending on design.

Example: Home network which can provide both Wi-Fi and ethernet.



**Open Systems Interconnection (OSI) model:**

It provides a way that how the network communication between layers. The OSI model is divided into seven distinct layers, each with specific responsibilities, ranging from physical hardware connections to high-level application interactions.

1. Physical layer: Transmits bit stream over physical medium-Ethernet
2. Data link layer: Defines the format of data on the network- Packets
3. Network layer: Defines which physical path the network will choose/handles routing
4. Transport layer: Transmitting data using protocols using TCP/UDP
5. Session Layer: Maintains sessions and responsible for controlling ports (authentication/ authorization).
6. Presentation layer: Ensures data in usable format and where data encryption occurs.
7. Application Layer: Where application can access the network services. (FTP/HTTP/SMTP/DNS)