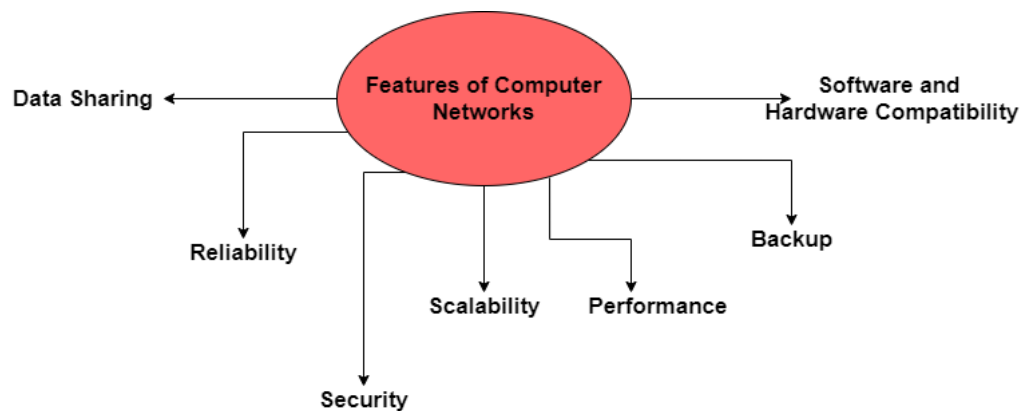


UNIT-1

Computer Network is a connection of two or more devices that are connected through a medium in order to exchange information. With the help of a Computer Network, you can easily send or receive data to or from a computing device.

Important features of Computer Networks:

- Data Sharing
- Reliability
- Scalability
- Security
- Performance
- Backup
- Software and Hardware Compatibility



1. Data Sharing

- Computer Network is that with their help you can easily share the data between different systems that are connected with each other through transmission media.
- Thus, resources can be easily shared between two or more devices over a network.

2. Reliability

With the help of Computer Networks there are fewer chances for the occurrence of failure and in case if there is any failure then recovery is fast.

3. Security

- A computer network should be secured so that the data transmission over a network should be safe from unauthorized access.

- The data sent by the **sending node** should be received as it is at the **receiving node**, which simply means there should be no loss of data during the transmission of the data.

4. Scalability

The scalability of a computer network simply means that we can add new nodes or components to the network easily.

Any computer network must be scalable so we can extend it easily by adding new nodes. After adding new nodes to the network the speed of connection decreases which leads to a decrease in the speed of transmission of data.

This problem can be resolved with the help of routing devices like Switch, Hub, Router, Repeater.

Let us take an example of this: Suppose there are 50 computers in a company for their 50 employees. Let's say the company hires another 100 employees and then the company adds 100 computers to the existing local area network then in that case LAN should allow the same.

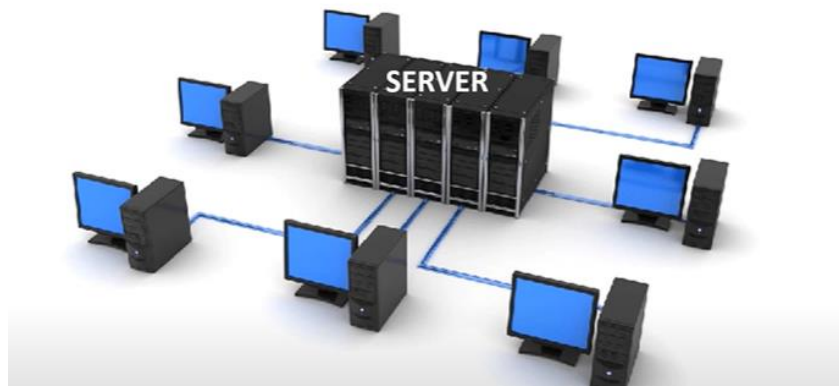
5. Performance

- The performance of a computer network is measured in terms of response time.
- For better performance, the response time of sending and receiving data from one node to another should be minimum.



6. Backup

A Computer Network must have a central server that keep backup of all data that is to be shared over a network so that at the time of failure in the network this central server helps in the faster recovery.



7. Software and Hardware Compatibility

Computer Network must not limit all the computers in a computer network to use the same software and hardware.

computer network should allow better compatibility between different software and hardware configuration.

With the help of a Computer network, it becomes very easy for all of us to share our knowledge and ideas with others for example: via email messaging, via video conferencing over the internet, everything becomes easy.

Network Programming Language

1. Perl

The 30-year old programming language continues to be popular among techies. With the new common gateway interface (CGI) scripting, Perl's popularity has increased over time.

2. Bash

The native shell of **Unix-based systems** including Linux, macOS is the **command-line interface** called **Bash**.

3. Tcl

Tool Command Language (Tcl) is among the most mature programming languages. Speed and power are the reasons why Tcl skills have gained popularity.

4. Go

Google created programming language, Go has its unique strengths. The programming language is used at the foundation of Docker, Kubernetes, Cloudflare, Netflix, and Uber.

5. Python

The **open-source programming language** is the top choice of open source community. Network admins use Python for **automating system administration** tasks.

6. Java

Writing Java programs that talk to Internet servers is easy. Java's core library includes classes for communicating with Internet hosts using the **TCP** and **UDP** protocols of the

TCP/IP family. You just tell Java what IP address and port you want, and Java handles the low-level details.

Platforms and Tools:

- Wireshark

Wireshark is a **network protocol analyzer**, or an application that captures packets from a network connection)

- Nmap

As we see the rise in **IoT devices** and therefore, the networks are getting more complex for the companies using IoT devices. In this situation, Nmap comes into view where it can be used for **auditing the network traffic** between web servers of the organization and the IoT devices.

- iPerf3

iPerf is an open source, free, network performance measurement tool. It works by simply **sending traffic from one host to another** and measuring the bandwidth that can be achieved. In addition to the throughput measurement it can give metrics such as **packet loss**, **jitter** (*the time delay between when a signal is transmitted and when it is received*), and **traffic distribution**

- Cisco Packet Tracer

Gain real skills with our **powerful network simulation tool** where you practice networking, IoT, and cybersecurity skills in a virtual lab – no hardware needed

- Netstat

The netstat command is used **to show network status**. Traditionally, it is used more for problem determination than for performance measurement.

- PRTG Wifi Monitor:

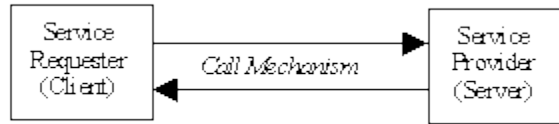
PRTG is a network monitoring tool that **helps you to ensure that your computer systems are running smoothly and that no outages occur**. Network monitoring is also important to increase the efficiency of your network by knowing bandwidth and resource consumption.

Client/Server Applications

- A client/server application consists of a **client** program that consumes services provided by a **server** program.
- The client requests services from the server by calling functions in the server application.

- In a distributed computing environment, where the client program and the server program execute on different machines and possibly even on different platforms, the client and server communicate through a communications layer that is often called *middleware*.

Figure: Basic client/server architecture



Client/Server Application Benefits:

Most of the benefits of using client/server architecture for enterprise applications relate to flexibility of deployment and relative ease of maintenance. For example, using client/server architecture you can typically:

- Re-use existing code for the business logic
- Distribute the processing and network loads.
- Quickly and easily change business logic procedures without changing the client program or user interface
- Multiple user can work on same server no dependency injection.
- Modification is easier.
- Use development tools that are designed to work together

A typical approach is to split the logical functions of the application into three:

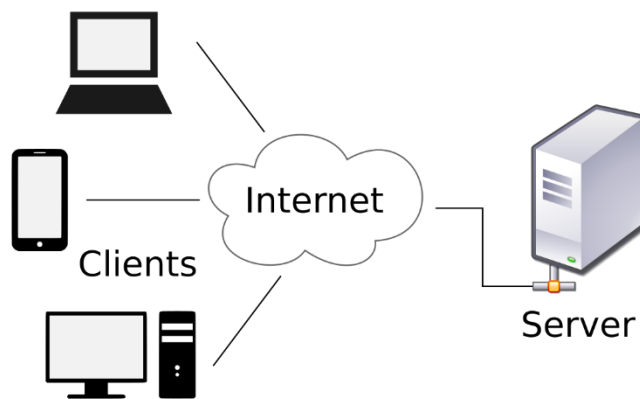
- User interface logic (screen handling)
- Business logic (data processing)
- Data access logic (file or database handling)

Common Client-Server Application:

- **Web Applications:** Typically use a web browser (client) to communicate with web servers. They often follow the n-tier architecture.
- **Email Services:** Use email clients (like Outlook or Gmail) to communicate with mail servers using protocols like SMTP, IMAP, and POP3.
- **Database Systems:** Database clients interact with database servers to query, update, and manage data.
- **File Sharing:** Clients use protocols like FTP to interact with file servers for uploading and downloading files.

Client Server Architecture:

Client-server architecture, alternatively called a client-server model, is a network application that breaks down tasks and workloads between clients and servers that reside on the same system or are linked by a computer network



Client-server architecture typically features multiple users' workstations, PCs, or other devices, connected to a central server via an Internet connection or other network. The client sends a request for data, and the server accepts and accommodates the request, sending the data packets back to the user who needs them.

This model is also called a client-server network or a network computing model.

- First, the client sends their request via a network-enabled device
- Then, the network server accepts and processes the user request
- Finally, the server delivers the reply to the client

Purpose of Client-Server Architecture

- Client-server architecture provides the exact framework that today's organizations need to meet the challenges of a rapidly evolving IT world.
- The client-server network model brings a higher level of processing.

Examples of Client-Server Architecture

Email Server:

File Server:

Web Server:

Advantages of Client-server networks:

- **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.
- **Security:** These networks are more secure as all the shared resources are centrally administered.
- **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.
- **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

Disadvantages of Client-Server network:

- **Traffic Congestion** is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.
- It does not have a robustness of a network, i.e., when the server is down, then the client requests cannot be met.

3-Tier Client-Server Architecture?

The three-tier client-server architecture consists of a presentation tier known as the User

Interface layer, an application tier called the Service layer, and a data tier comprising the database server. Three-tier architecture can be divided into three parts:

- Presentation layer (or Client Tier): This layer takes care of the User Interface.
- Application layer (or Business Tier): This layer handles the detailed processing.
- Database layer (or Data Tier): This layer stores the information.

The Client system controls the Presentation layer; the Application server looks after the Application layer, and the Server system supervises the Database layer.

Here's a model of 3-Tier client-server architecture:



Strength and weakness of java as network programming language

Strengths

1. **Platform Independence:** Java programs run on any device with a Java Virtual Machine (JVM).
2. **Rich Standard Library:** Java has extensive libraries for network programming.
3. **Security Features:** Java provides a secure execution environment.
4. **Concurrency Support:** Java handles multiple tasks simultaneously with its multi-threading capabilities.
5. **High Performance:** Java uses a Just-In-Time (JIT) compiler for optimized execution and supports scalable I/O operations.
6. **Community Support:** Java has a large community and many resources available.

Weaknesses

1. **Complexity:** Java's syntax is verbose and can be harder to maintain.
2. **Performance Overhead:** The JVM introduces some performance overhead.
3. **Memory Consumption:** Java applications use more memory due to the JVM and garbage collection.
4. **Steeper Learning Curve:** Advanced features can be challenging to master.
5. **Limited Low-Level Control:** Java abstracts many low-level operations, limiting fine-grained control.
6. **JVM Dependency:** Java requires the JVM to be installed on the host machine.

Installing VS Code

Install Visual Studio Code

Installing Java in Visual Studio Code.

1. Go to <https://code.visualstudio.com/docs/languages/java>

2. *Download Package*

Install Visual Studio Code for Java

To help you set up quickly, we recommend you use the **Coding Pack for Java**, which is the bundle of VS Code, the Java Development Kit (JDK), and a collection of suggested extensions by Microsoft. The Coding Pack can also be used to fix an existing development environment.

Install the Coding Pack for Java - Windows

Install the Coding Pack for Java - macOS

Note: The Coding Pack for Java is only available for Windows and macOS. For other operating systems, you will need to manually install a JDK, VS Code, and Java extensions.

3. Open VS Code. Now open command ctrl+shift+p (in VS Code)
4. Select in dropdown (Java:create java project)
5. Select (No built tools)
6. Select your Folder Location
7. Type name of your project given in screenshot

TestProj

Input a Java project name (Press 'Enter' to confirm or 'Escape' to cancel)

```
1 public class App {  
    Run | Debug  
2     public static void main(String[] args) {  
3         System.out.println(x: "Hello, Wo  
4     }  
5 }  
6
```