COMPUTER NETWORKS

N Praveen Kumar - Assistant professor,CSE

LEARNING OBJECTIVES

- To understand the functions of key Internet protocols like DNS, HTTP, FTP, and SMTP.
- To learn how services like DDNS, Telnet and SNMP support network communication and management.
- To examine the process of file transfer, remote access, and information sharing across networks.
- To examine the role of Firewalls in ensuring network security and access control.

DNS(Domain Name System)

DNS (Domain Name System) is a system that translates human readable domain names (like www.google.com) into IP addresses (like 142.250.190.78) so that computers can locate and communicate with each other over networks.

Objectives of DNS:

1.Name to Address Resolution:

Converts readable domain names into numerical IP addresses that computers use to locate resources on the internet.

2.Hierarchical and Distributed Naming System:

DNS is organized hierarchically and distributed across multiple servers, reducing congestion and increasing reliability.

3.Fault Tolerance and Redundancy:

Uses multiple servers to ensure continuity and reliability even if one DNS server fails.

4.Load Balancing:

DNS can distribute network traffic among multiple servers (using round-robin techniques) which improves performance and speed.

5.Ease of Use and Scalability:

Simplifies the process of locating internet services while being scalable to handle billions of domain names worldwide.

Main Components of DNS:

1.Domain Name Space:

A tree-like structure that organizes all domain names hierarchically.

2.DNS Zones:

Portions of the DNS namespace managed by specific organizations or administrators.

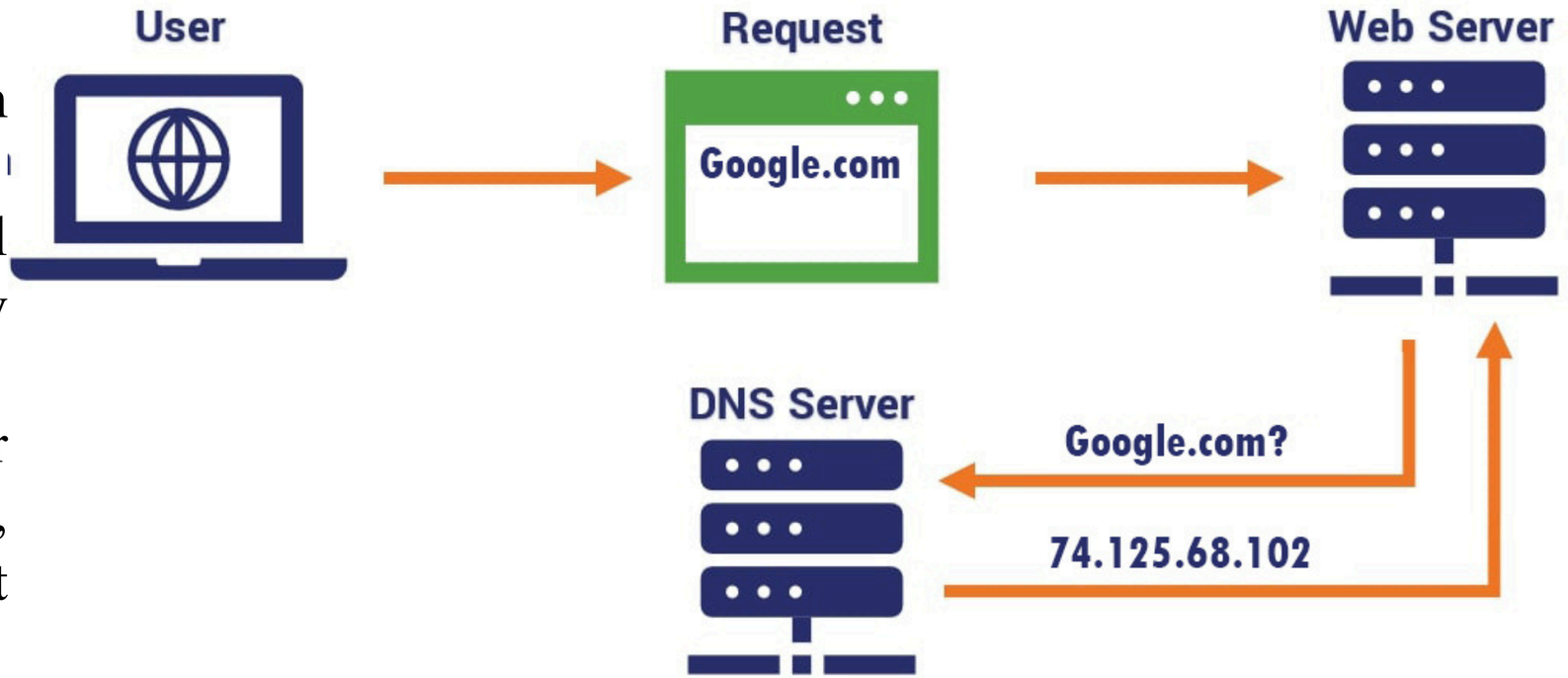
3.DNS Servers:

- **Root DNS Server:** Top-level servers that know where to find TLD servers.
- **Top-Level Domain (TLD) Server:** Handles domains like .com, .org, .in.
- **Authoritative DNS Server:** Stores DNS records for specific domains.
- **Caching/Resolver Server**: Temporarily stores DNS query results to speed up future lookups.

4.DNS Records (Resource Records):

- **A Record:** Maps a domain name to an IPv4 address.
- **AAAA Record:** Maps a domain name to an IPv6 address.
- **CNAME:** Canonical name (alias for another domain).
- **MX Record:** Mail exchange record (for email servers).
- **NS Record:** Identifies the authoritative name servers for a domain.
- **PTR Record(Pointer Record):** Used for reverse DNS lookups (IP to name).

How DNS Works



- You enter www.example.com in a browser.
- The request goes to the local DNS resolver (usually provided by your ISP).
- If not cached, the resolver queries the Root DNS server, which points to the relevant TLD server (for .com).
- The TLD server directs the query to the Authoritative DNS server for example.com.
- The authoritative server returns the IP address of www.example.com.
- Your browser connects to that IP address and loads the website.

Advantages of DNS

- Simplifies internet access by using names instead of IPs.
- Increases reliability with distributed servers.
- Improves performance via caching.

DDNS(Dynamic Domain Name System).

- DDNS stands for Dynamic Domain Name System.
- It is an extension of the standard DNS that allows automatic updation of a domain name's IP address whenever the IP changes. This is especially useful for networks where the IP address is not static (commonly home or small business networks).

Need for DDNS:

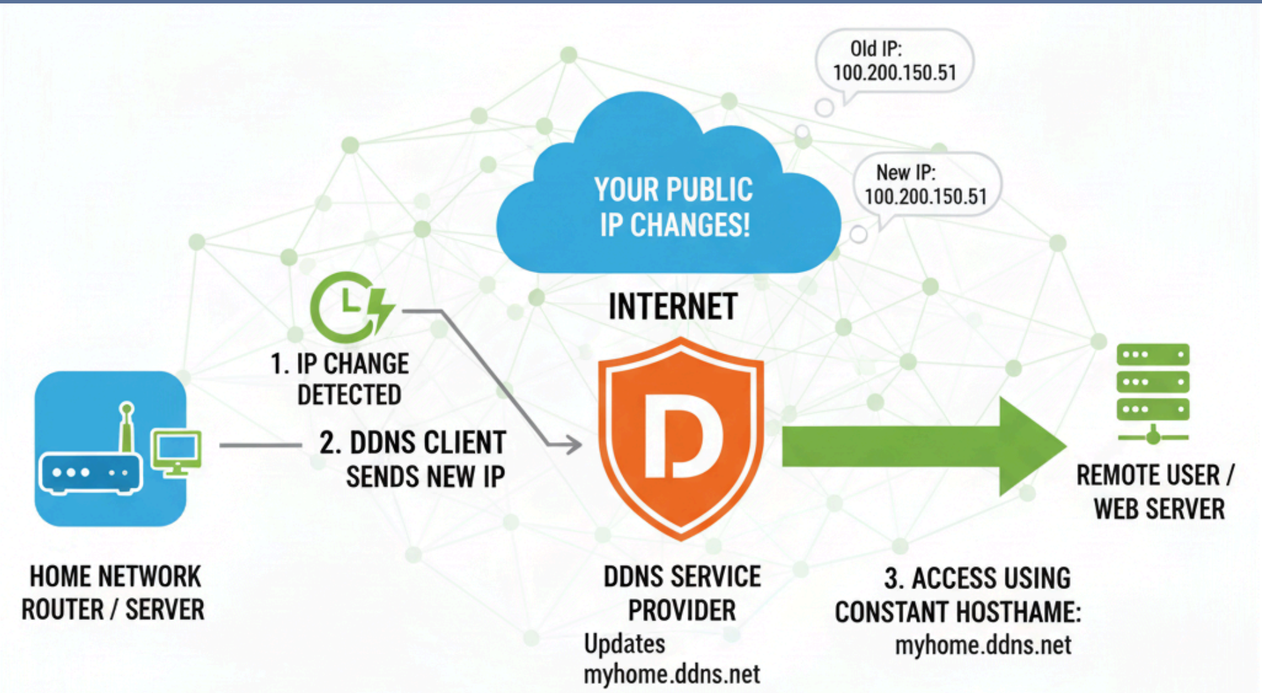
- **Dynamic IPs:** Most home networks and small businesses use dynamic IPs assigned by ISPs. DDNS ensures that even if the IP changes, the domain name still points to the right device.
- **Remote Access:** This allows users to reach home servers, IoT devices surveillance systems using a consistent domain name.
- **IoT & Cloud Integration:** It is crucial for smart devices that need to be accessible over the internet despite changing IPs.

How It Works:

- Client Device(e.g., router or PC) detects a change in its public IP.
- DDNS Client Software sends the new IP to the DDNS provider.
- DDNS Provider updates the DNS record for the domain name.
- Users can access the device using the same domain name, even after IP changes.

Example:

- Your public IP is 102.55.21.88 today.
- You register myhome.ddns.net with a DDNS provider.
- Tomorrow, your ISP changes your IP to 102.55.21.120.
- DDNS automatically updates your domain's record, so myhome.ddns.net still points to your device correctly.



Use Cases:

- Accessing home networks or servers remotely.
- Hosting websites or game servers on a dynamic IP.
- Connecting to security cameras or IoT devices from outside.
- Running VPN or FTP servers at home.

Advantages:

- Keeps your domain accessible even with changing IPs.
- Ideal for home servers, CCTV systems, IoT devices, and remote access.
- Reduces need for a static IP (which is usually costly).

Disadvantages:

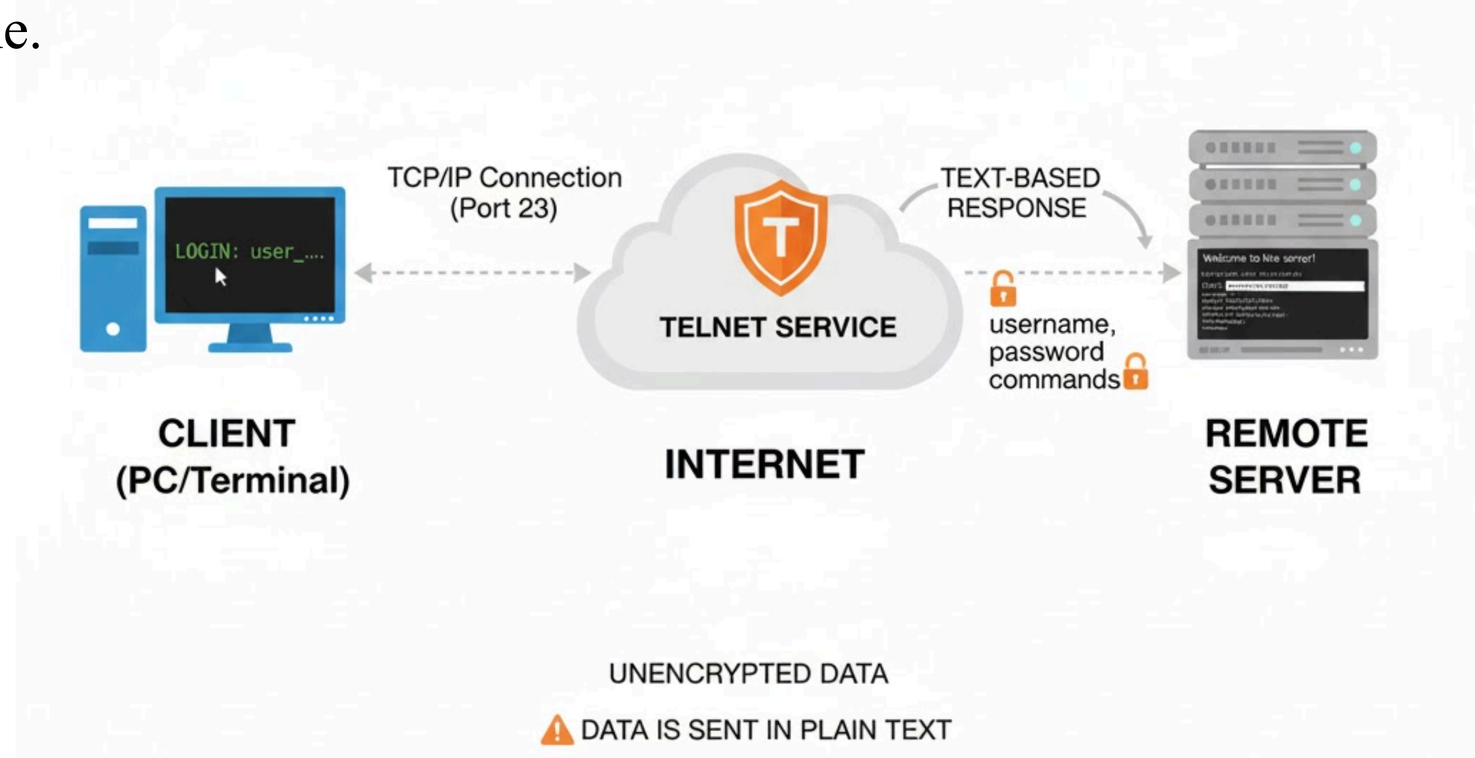
- Relies on third-party DDNS service providers.
- May have downtime or delays in updating IPs.
- Less secure if updates are not encrypted.

Popular DDNS Providers:

- No-IP (noip.com)
- DuckDNS (duckdns.org)
- Dynu (dynu.com)
- DynDNS (now part of Oracle)

TELNET(Telecommunication Network)

- TELNET is mainly used for remote login and administration of servers, routers, and other network devices.
- It enables text-based communication between two systems typically between a client (user's computer) and a server.
- It operates over port 23 and uses the TCP/IP model to establish a connection between a local and remote machine.



TelNet Working Principle:

1.Connection Establishment

- The Telnet client initiates a connection to the Telnet server using the server's IP address and port 23.
- TCP establishes a reliable session between the two devices.

2. Virtual Terminal Emulation

- Telnet creates a virtual terminal session, meaning the client behaves as if it's directly interfacing with the server's command line.
- This is done using Network Virtual Terminal (NVT) standards, which define how characters and commands are exchanged.

3.Command Transmission

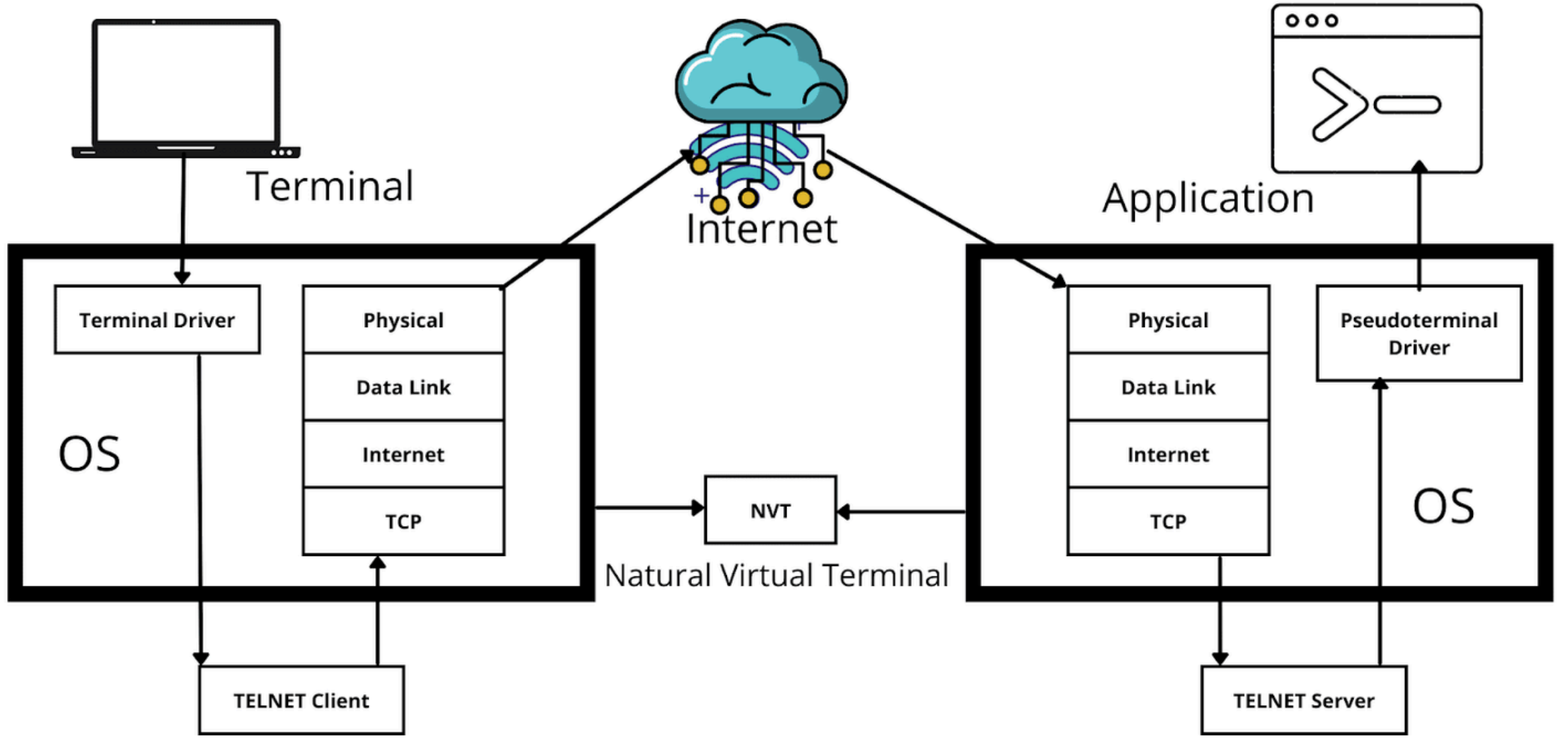
- The user types commands on the client.
- These commands are sent as plain text to the server.
- The server executes the commands and sends the output back to the client.

4. Session Termination

- The session continues until the user logs out or the connection is closed.
- TCP ensures all data is delivered accurately before the session ends.

Example:

If you type in the command prompt:
telnet 192.168.1.10
This connects you to the device at IP 192.168.1.10.
After successful login, you can execute commands directly on that device.



Common Uses:

- Managing network devices (like routers/switches).
- Testing open ports or connections.
- Debugging network services.
- Accessing old legacy systems that still support Telnet.

EMAIL

- EMAIL (Electronic Mail) is a method of sending and receiving digital messages over the internet.
- It allows users to exchange text, files, images, and other attachments quickly between computers, smartphones, or other devices.

Email Structure

An email address has two main parts:
username@domainname

Example:
abc123@gmail.com

abc123 → username
gmail.com → domain name (mail server)

Purpose:

- Personal and professional communication
- Sharing documents, files, and media
- Notifications and alerts from online services
- Official correspondence and business communication

How It Works

Email works based on a client-server model and uses specific protocols for sending and receiving messages.

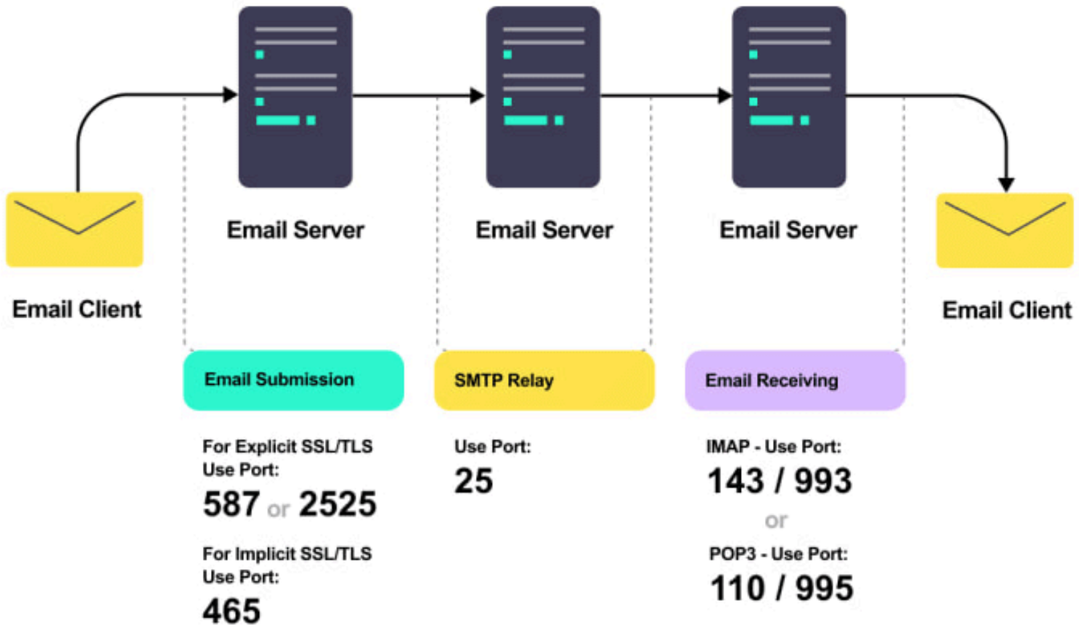
Composing: The user writes an email in an email client (like Gmail, Outlook, or Thunderbird).

Sending: The client sends the message to an SMTP server (Simple Mail Transfer Protocol).

Routing: The SMTP server finds the recipient's mail server.

Receiving: The recipient's mail server stores the message using POP3 (Post Office Protocol) or IMAP (Internet Message Access Protocol).

Reading: The recipient logs into their email client and reads the message.



Protocol	Full Form	Function
SMTP	Simple Mail Transfer Protocol	Used to send emails
POP3	Post Office Protocol version 3	Used to receive and download emails
IMAP	Internet Message Access Protocol	Used to access and manage emails directly on the server

Advantages

- Fast and cost-effective communication
- Can send attachments (documents, images, videos)
- Accessible anywhere via the Internet

Disadvantages

- Spam and phishing emails can cause security risks
- Privacy can be compromised if not encrypted
- Attachments can spread malware if unsafe

Common Email Services

- Gmail (Google)
- Outlook / Hotmail (Microsoft)
- Yahoo Mail
- ProtonMail (Secure email service)

File Transfer Protocol (FTP)

FTP (File Transfer Protocol) is a network protocol used to transfer files between a client and a server over a TCP/IP network such as the Internet or a local network.

Purpose:

FTP is used for:

- Uploading files from a local computer to a remote server
- Downloading files from a remote server to a local system
- Managing files on the server (rename, delete, move, etc.)

How FTP Works:

FTP works on a client-server model:

- The client requests file operations.
- The server responds and performs the requested actions.

It uses two separate connections:

- **Control Connection (Port 21)**: For sending commands and responses
- **Data Connection (Port 20)**: For transferring actual file data

FTP Modes:

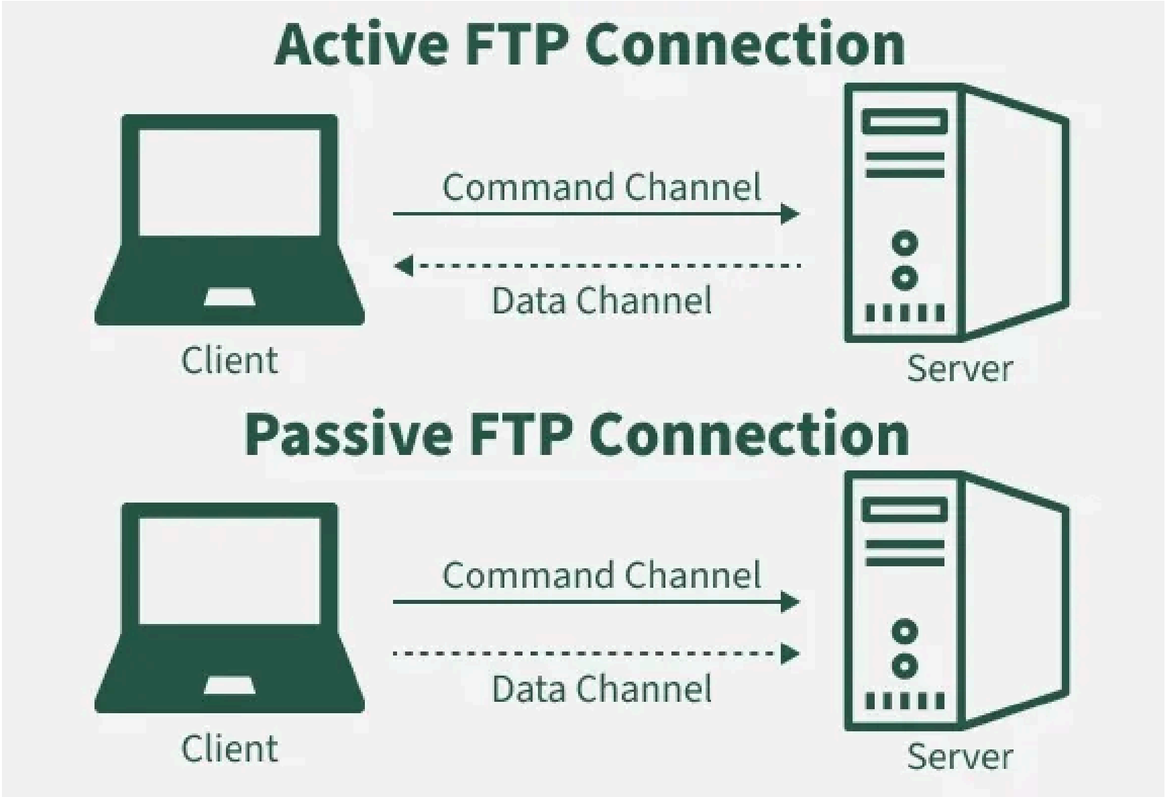
1.Active Mode:

The client opens a random port and tells the server to connect to it for data transfer.

2.Passive Mode:

The server opens a random port, and the client connects to it.

Used when clients are behind firewalls.



Common FTP Commands:

Command	Description
USER	Specifies username
PASS	Specifies password
LIST	Lists files in a directory
RETR	Retrieves (downloads) a file
STOR	Stores (uploads) a file
DELE	Deletes a file
QUIT	Ends the session

Types of FTP:

- **Anonymous FTP:** No login required (public access)
- **Password-protected FTP:** Requires valid credentials
- **Secure FTP (FTPS):** Adds SSL/TLS encryption for security
- **SFTP (SSH File Transfer Protocol):** Uses SSH for secure file transfer (different from FTP)

Advantages:

- Easy file sharing between systems
- Supports large file transfers
- Can resume interrupted transfers
- Widely supported by tools and servers

Disadvantages:

- Data (including passwords) sent in plain text in normal FTP
- Not secure without encryption
- Requires configuration of firewalls and ports

WWW

The World Wide Web (WWW) is a vast information system on the Internet where documents and multimedia content are linked together using hyperlinks.
It allows users around the world to access, share, and interact with data using web browsers.

How the WWW Works

User Request:
You type a web address (URL) like https://www.wikipedia.org in your browser.
DNS Resolution:
The browser contacts the Domain Name System (DNS) to find the IP address of the web server that hosts the site.
HTTP/HTTPS Request:
The browser sends an HTTP request to that server, asking for the web page.
Server Response:
The web server processes the request and sends back the web page files (HTML, CSS, images, scripts, etc.).
Rendering:
The browser interprets these files and displays a formatted, interactive page to the user

Types of Websites on the WWW

Type	Description	Examples
Static Websites	Show fixed content; same for every user	Company profiles, portfolios
Dynamic Websites	Display content that changes based on user input or database data	Facebook, Amazon
E-commerce Sites	Allow users to buy/sell products or services	Flipkart, eBay
Educational Sites	Offer online learning materials	Coursera
News/Blogs	Provide articles and current updates	BBC, Medium

Importance of WWW

- Information Access: Billions of pages accessible globally.
- Communication: Enables email, social media, and instant messaging.
- Commerce: Foundation of e-commerce and online transactions.
- Education: e-learning.

Key Technologies Supporting WWW

Technology	Function
HTML/CSS/JavaScript	Structure, design, and interactivity of web pages
Web Servers (Apache, Nginx)	Host and deliver web content
Databases (MySQL, MongoDB)	Store and retrieve website data
APIs (Application Programming Interfaces)	Enable data exchange between apps
Cloud Computing	Scalable web hosting and storage
Search Engines (Google, Bing)	Index and rank web pages for discovery

Security in the WWW

- HTTPS – Secures communication through encryption.
- SSL/TLS Certificates – Verify website authenticity.
- Firewalls and Anti-malware – Protect web servers.
- User Authentication – Ensures access control for sensitive data.

HTTP

- HTTP stands for HyperText Transfer Protocol.
- A set of rules that allows web browsers (clients) and web servers to exchange information.
- Whenever you open a website (like https://www.wikipedia.org), your browser uses HTTP (or its secure version, HTTPS) to request and receive web pages, images, videos, etc from a web server.

How HTTP Works

User Enters a URL:

You type a web address (e.g., http://example.com) in your browser.

Browser Sends Request:

The browser sends an HTTP request to the web server asking for the webpage.

Server Processes Request:

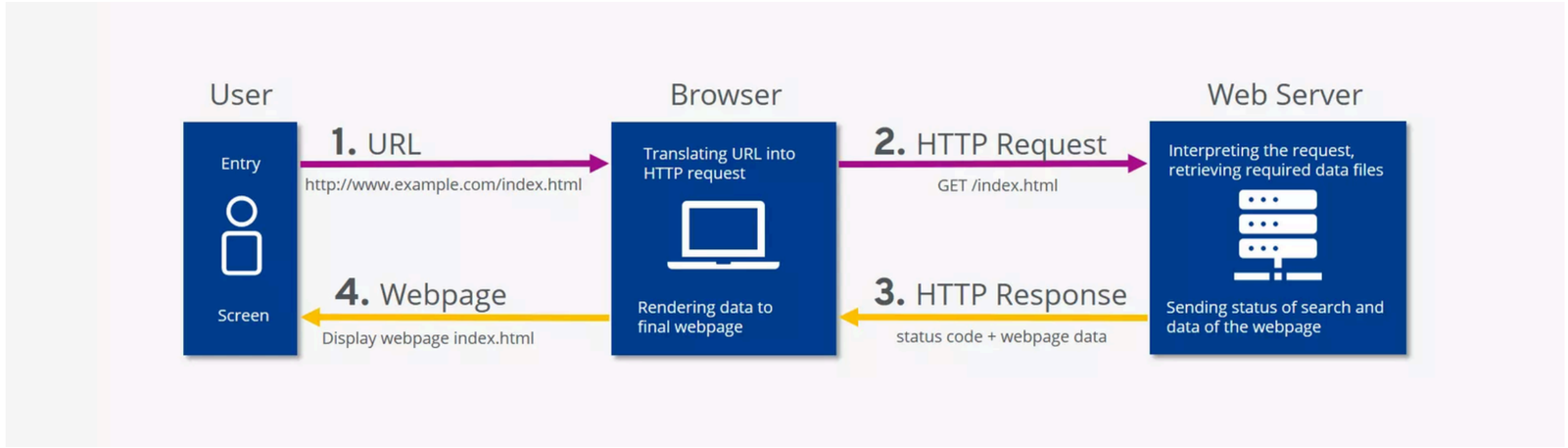
The web server finds the requested page or resource.

Server Sends Response:

The server returns an HTTP response (including the requested content and a status code).

Browser Displays the Page:

The browser interprets the response and displays it on your screen.



Structure of an HTTP Message

1.HTTP Request (from client to server)

Example:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html
```

- **Request Line:** Specifies method (GET), resource (/index.html), and protocol version (HTTP/1.1).
- **Headers:** Provide additional info like browser type, language, etc.
- **Body (optional):** Used in POST/PUT requests to send data to the server.

HTTP Response (from server to client)

HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1256

- **Status Line:** Includes protocol version and status code (e.g., 200 OK).
- **Headers:** Metadata about the response (like file type).
- **Body:** Actual content (HTML page, image, etc.).

Common HTTP Methods

Method	Description	Example Use
GET	Retrieve data from server	Load a webpage
POST	Send data to server	Submitting a form
PUT	Update an existing resource	Editing profile info

Key Characteristics of HTTP

Stateless: Each request is independent the server doesn't remember previous interactions.
Connectionless: Connection closes once data is transferred (except in HTTP/1.1+).
Flexible: Can transmit text, images, video, JSON, etc.
Extensible: Supports custom headers and authentication mechanisms.

SNMP

SNMP (Simple Network Management Protocol) is a standard protocol used in computer networks to monitor, manage, and control network devices such as routers, switches, servers, printers, and IoT devices. It operates in the Application Layer (Layer 7) of the OSI model.

SNMP allows a network administrator to collect information about network performance, detect faults, and configure devices remotely.

It works on a client-server model involving:

- **SNMP Manager (Client)** – The controlling system that requests and collects data.
- **SNMP Agent (Server)** – A software component running on the network device being monitored.
- **MIB (Management Information Base)** – A structured database of information about the device.

SNMP Architecture

Manager:

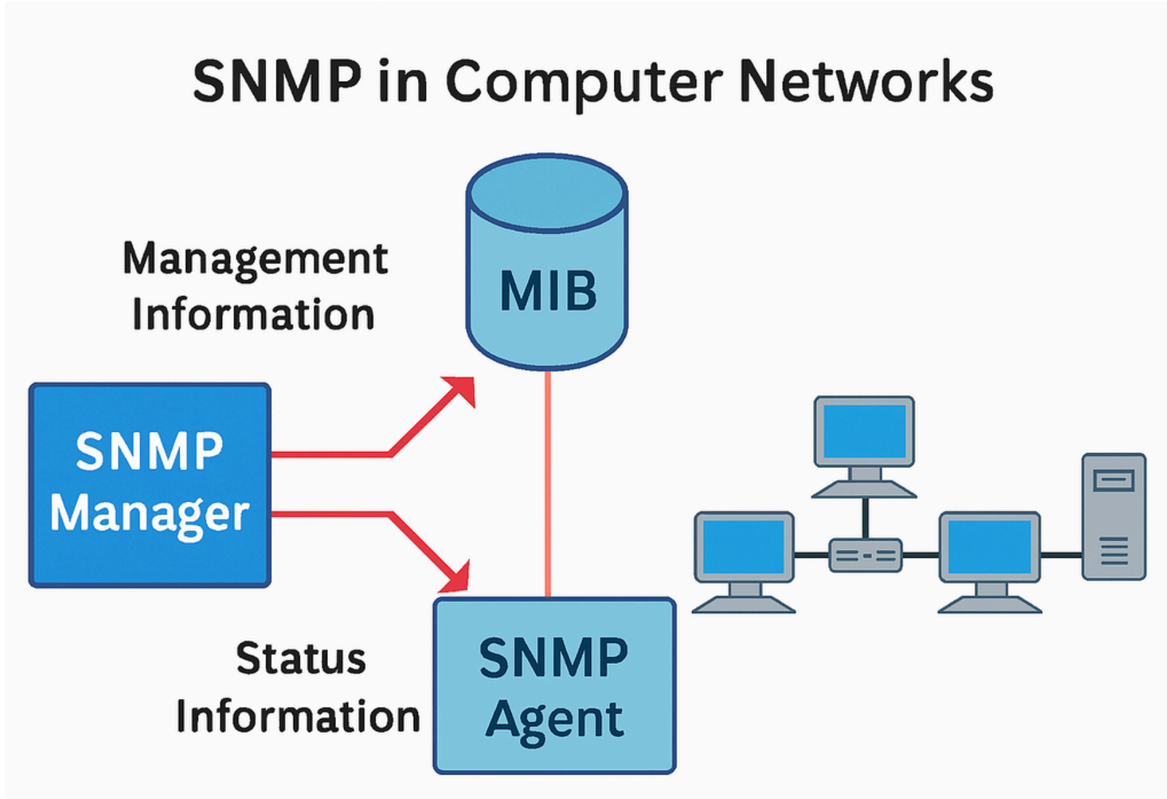
- Runs SNMP management software.
- Sends requests to agents and receives responses.
- Uses gathered data to monitor or configure the network.

Agent:

- Resides on the managed device (like a router or switch).
- Collects data from the device and stores it in the MIB.
- Sends information back to the manager upon request or automatically via traps.

MIB (Management Information Base):

- A virtual database containing a hierarchy of objects (OIDs – Object Identifiers).
- Each OID represents a specific piece of device information (e.g., CPU usage, interface status, packet count).



SNMP Communication Model

SNMP uses the UDP protocol:

- Port 161 – Used by the SNMP Manager to send requests to agents.
- Port 162 – Used by agents to send traps (unsolicited alerts) to the manager.

Example Use Cases

- Monitoring network traffic or bandwidth utilization.
- Checking device uptime or interface errors.
- Detecting link failures or CPU/memory usage alerts.
- Automating network configuration and fault management.

Advantages

- Lightweight and simple to implement.
- Platform-independent.

Disadvantages

- Complex MIB structure.
- Can generate high network traffic if not configured properly.

Fire Wall

A firewall is a network security device or software that monitors and controls incoming and outgoing network traffic based on predetermined security rules. Its main purpose is to create a barrier between a trusted internal network and untrusted external networks (like the Internet).

Functions of a Firewall

Packet Filtering – Inspects each packet and permits or blocks it based on rules (like IP address, port number, or protocol).

Proxying (Application Gateway) – Acts as an intermediary between users and the internet to inspect application-level traffic.

Stateful Inspection – Tracks active connections and allows packets that are part of an established session.

Network Address Translation (NAT) – Hides internal IP addresses from external networks for security.

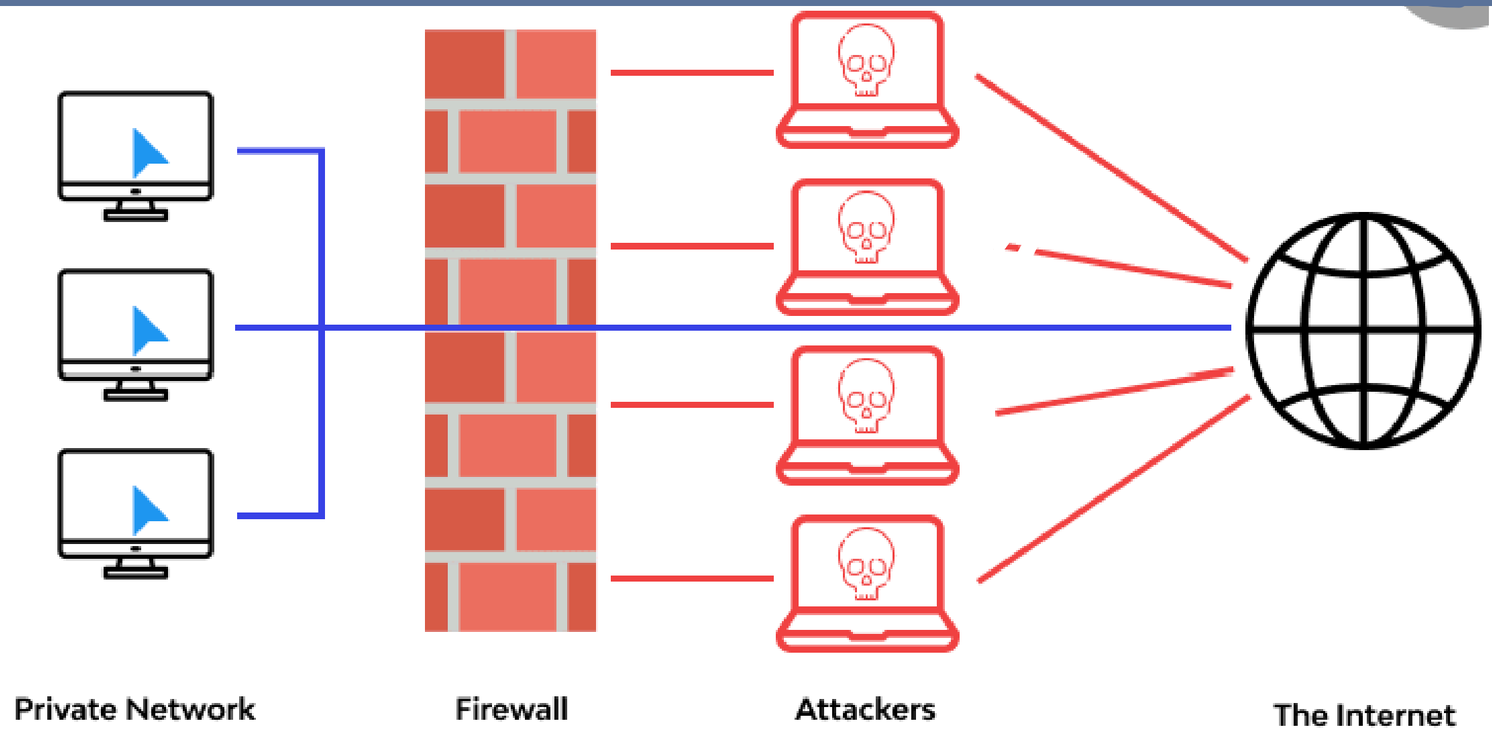
Logging and Monitoring – Records traffic details and alerts administrators of suspicious activities.

Types of Firewalls

Type	Description
Packet-Filtering Firewall	Examines headers of packets based on rules (e.g., IP, port, protocol). Simple but limited.
Stateful Inspection Firewall	Monitors connection states and ensures packets belong to valid sessions.
Proxy Firewall	Works at the application layer and filters traffic through an intermediary server.
Next-Generation Firewall (NGFW)	Combines traditional firewall functions with intrusion prevention, deep packet inspection, and threat intelligence.
Hardware Firewall	Physical device placed between the network and gateway.
Software Firewall	Installed on a host computer or server to protect that specific device.
Cloud Firewall	Virtual firewall deployed in cloud-based environments for protecting cloud infrastructure.

Advantages

- Protects against unauthorized access
- Blocks malicious traffic and cyberattacks
- Provides logging and auditing of network activity
- Enforces network usage policies



Limitations

- Cannot stop attacks that bypass the network (like phishing emails).
- Needs regular rule updates and tuning.
- Can cause performance overhead if rules are poorly optimized.

Layer No.	Layer Name	Main Functionality	Data Unit	Examples / Protocols
7	**Application Layer**	Provides user interface and network services to end-users (e.g., email, file transfer, web access).	Data	HTTP, HTTPS, FTP, SMTP, DNS, POP3
6	**Presentation Layer**	Translates, encrypts, and compresses data for application layer; ensures data format compatibility.	Data	SSL/TLS, JPEG, MPEG, ASCII, Encryption
5	**Session Layer**	Establishes, manages, and terminates sessions between communicating systems.	Data	NetBIOS, PPTP, RPC, SIP
4	**Transport Layer**	Ensures reliable data transfer, error correction, and flow control; segments and reassembles data.	Segment	TCP, UDP
3	**Network Layer**	Handles logical addressing, routing, and packet forwarding between networks.	Packet	IP, ICMP, IPsec, OSPF, BGP
2	**Data Link Layer**	Provides physical addressing (MAC), error detection, and frame synchronization within a local network.	Frame	Ethernet, PPP, ARP, Switches
1	**Physical Layer**	Transmits raw bitstreams over physical medium; defines cables, connectors, and signal standards.	Bits	Hubs, Cables, Wi-Fi, Fiber, RJ45