MODULE 6: APPLICATION LAYER

Q.1) SHORT NOTE ON DNS

The **Domain Name System (DNS)** is an essential component of the internet infrastructure, functioning as a directory that translates user-friendly domain names (e.g., `www.example.com`) into machine-readable IP addresses (e.g., `192.0.2.1`).

This translation allows users to access websites, email servers, and other online services without needing to remember numerical IP addresses.

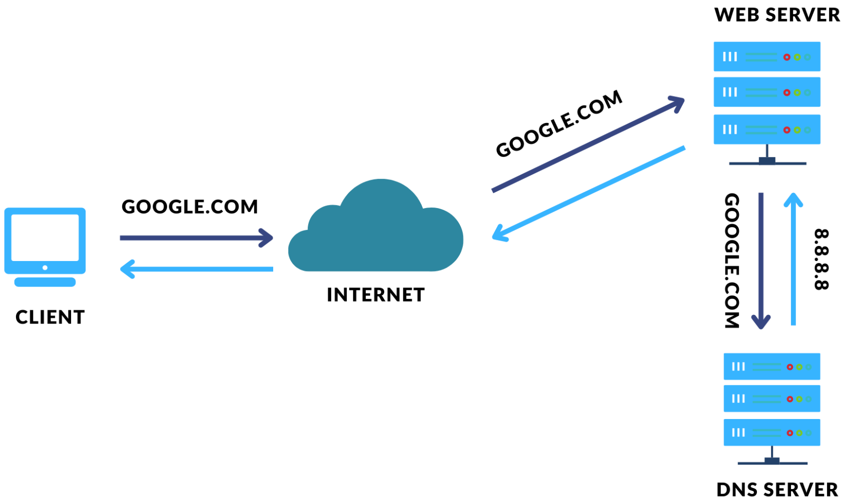
DNS operates through a globally distributed database and follows a hierarchical structure to resolve domain names efficiently.

How DNS Works:

1. **User Query**:
    - When a user types a domain name into their browser, the computer first checks its local DNS cache to see if it has a recent IP address for the domain.
    - If the IP address isn't in the cache, the query is sent to a DNS **recursive resolver**, typically provided by the user's ISP or a third-party DNS provider (e.g., Google DNS).
2. **DNS Resolution Process**:
    - The resolver queries DNS servers in a sequence to resolve the domain name into an IP address.
    - **Root Server**: The resolver first contacts a root DNS server. There are 13 root servers globally, and they contain references to **Top-Level Domain (TLD) servers** rather than the actual IP address.
    - **TLD Server**: The root server directs the resolver to a TLD server based on the domain's extension (e.g., `.com`, `.net`). TLD servers then guide the

query to the **authoritative name server** for the domain.

- **Authoritative Name Server**: This server has the DNS records for the domain and provides the final IP address associated with the domain name. The resolver then caches the IP address and returns it to the user's device, which establishes a connection to the destination server.



DNS Record Types:

DNS relies on different types of records to store and manage information about domains. Some of the most common records include:

- **A Record (Address Record)**: Maps a domain name to an IPv4 address.
- **AAAA Record**: Maps a domain name to an IPv6 address.
- **CNAME Record (Canonical Name)**: Maps an alias domain name to a true or canonical domain name (e.g., `blog.example.com` to `example.com`).
- **MX Record (Mail Exchange)**: Directs email to a specific mail server for a domain.
- **TXT Record**: Stores text information, often used for verification and security purposes, like SPF or DKIM for email security.
- **PTR Record**: Provides reverse DNS lookups, mapping an IP address back to a domain name.

DNS Caching:

To improve efficiency and speed, DNS responses are cached at various levels, including the user's browser, the operating system, and the recursive resolver.

This caching reduces the need for repetitive queries and minimizes response time for frequently accessed sites.

- **TTL (Time to Live)**: Each DNS record has a TTL, which determines how long it can be cached. Shorter TTLs mean more frequent updates but can increase traffic load on DNS servers, while longer TTLs reduce load but may slow propagation of updates.

DNS Applications:

DNS is essential for:

- **Website Browsing**: Translating domain names to IP addresses allows browsers to load websites.
- **Email Delivery**: MX records direct email to the correct mail server.
- **Service Discovery**: Various applications and devices rely on DNS to locate network services and other devices.
- **Content Delivery Networks (CDNs)**: Many CDNs use DNS to distribute content efficiently, directing users to the nearest or least-congested server for faster loading times.

Summary

DNS enables user-friendly access to the internet by translating domain names into IP addresses through a hierarchical and distributed network of servers. It relies on different types of records to manage various functions and utilizes caching for efficiency. DNS security, redundancy, and caching are critical for its reliability, while protocols like DNSSEC add essential security against data tampering. Through its structured approach, DNS supports nearly every internet application, from web browsing to email.

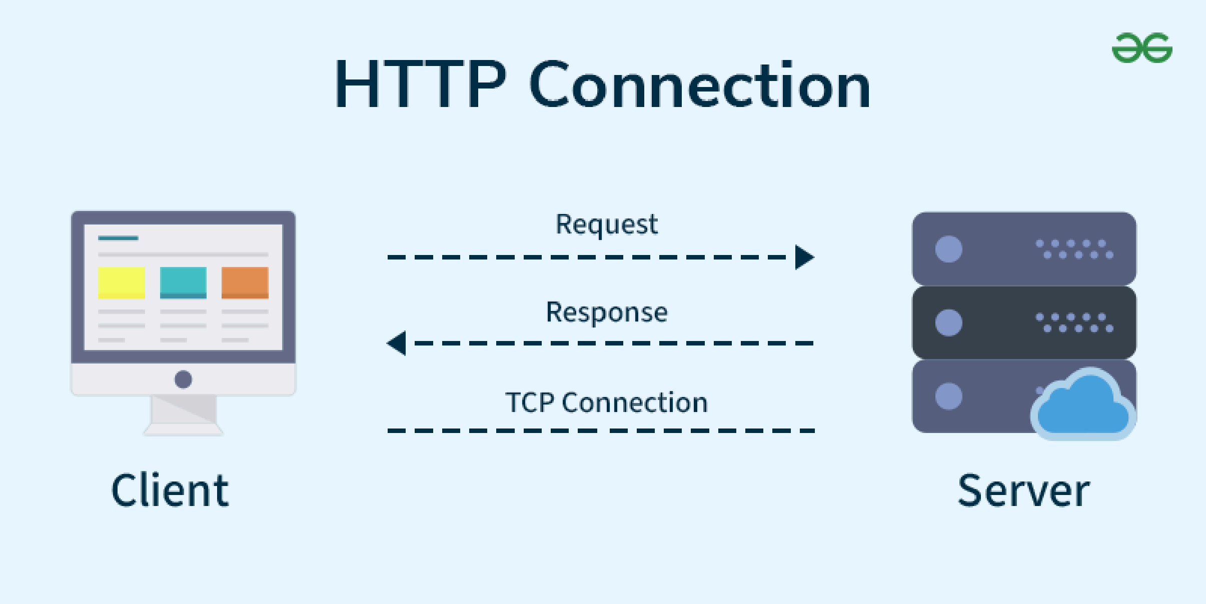
Q.2) SHORT NOTE ON HTTP

⇒

HTTP (Hypertext Transfer Protocol) is an application-layer protocol used for transmitting data over the web.

It enables the transfer of information like text, images, videos, and other multimedia files between clients (such as web browsers) and servers.

HTTP is the foundation of data communication on the internet, especially for the World Wide Web.



Key Features of HTTP:

1. **Stateless Protocol**: Each HTTP request from a client to a server is independent. The server does not retain information about previous requests, which simplifies design but can lead to challenges in maintaining user sessions (often addressed with cookies).
2. **Request-Response Model**: HTTP uses a request-response structure where a client (usually a web browser) sends a request to the server, and the server returns a response. Typical request methods include:
    - **GET**: Requests data from the server.
    - **POST**: Submits data to the server.
    - **PUT**: Updates existing data on the server.
    - **DELETE**: Removes data from the server.
3. **HTTP Headers**: HTTP headers provide additional information about the request or response. They may include details like content type, content length, server information, and caching instructions.

4. **HTTP Status Codes**: These codes provide feedback on the outcome of the HTTP request:
    - **200 OK**: Success.
    - **404 Not Found**: The requested resource does not exist.
    - **500 Internal Server Error**: The server encountered an error processing the request.
5. **Versions**:
    - **HTTP/1.1**: The most widely used version, with persistent connections and chunked transfer encoding.
    - **HTTP/2**: Introduced multiplexing, header compression, and server push to improve performance.
    - **HTTP/3**: Uses QUIC protocol, enhancing speed and security over unreliable networks.

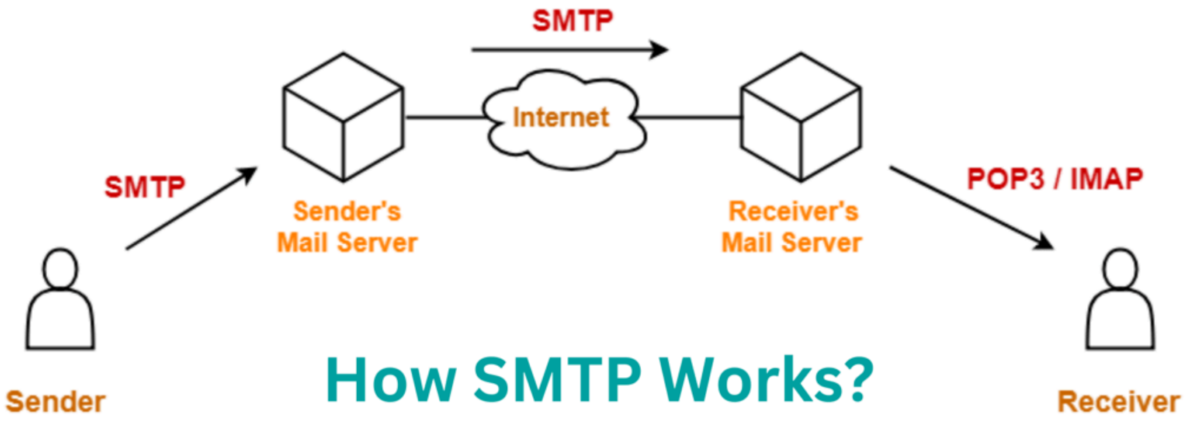
Applications:

HTTP is essential for loading web pages, supporting APIs for web-based applications, and is the primary protocol for interacting with the vast majority of online services. Secure HTTP, or HTTPS, adds a layer of security using SSL/TLS, encrypting the data transfer between client and server to protect sensitive information.

Q.3) SHORT NOTE ON SMTP

SMTP (Simple Mail Transfer Protocol) is a protocol used for sending and relaying emails over the internet.

It operates at the application layer and is essential for email communication, as it defines the way emails are transmitted between mail servers and from client devices to mail servers.



Key Features of SMTP:

1. **Email Sending and Relaying**: SMTP is responsible for the process of email delivery. It handles the sending of emails from the sender's email client to the recipient's mail server and transfers messages between mail servers if needed.
2. **Push Protocol**: SMTP is a push-based protocol, meaning it actively pushes emails from the sender to the recipient's server rather than waiting for the recipient to retrieve them. This makes SMTP ideal for sending outgoing mail.
3. **Command-Based Protocol**: SMTP operates through a series of commands and responses between the client (usually an email application or mail server) and the server. Common SMTP commands include:
   - **HELO/EHLO**: Initiates communication with the mail server.
   - **MAIL FROM**: Identifies the sender’s email address.
   - **RCPT TO**: Specifies the recipient’s email address.
   - **DATA**: Indicates that the email message data will follow.
   - **QUIT**: Ends the SMTP session.
4. **Port Numbers**: SMTP typically uses **port 25** for server-to-server email transmission, **port 587** for encrypted email transmission with STARTTLS, and **port 465** for secure SMTP over SSL/TLS.
5. **Text-Based Protocol**: SMTP messages are text-based, meaning all commands and data are sent as text, making it simpler to implement but susceptible to interception if not encrypted.

Limitations of SMTP:

- **Lack of Encryption**: Basic SMTP does not encrypt messages, though encryption can be added with STARTTLS or SSL/TLS.
- **Only Sends Outgoing Mail**: SMTP cannot retrieve incoming messages; protocols like POP3 or IMAP are used for retrieving emails.

Applications:

SMTP is widely used by email clients (e.g., Outlook, Gmail, Thunderbird) to send messages to mail servers and by servers to relay messages across networks.

With SSL/TLS encryption, SMTP is essential for secure email transmission, making it a fundamental component in email infrastructure and communication.

Q.4) SHORT NOTE ON DHCP & FTP

DHCP (Dynamic Host Configuration Protocol)

DHCP stands for Dynamic Host Configuration Protocol. It is the critical feature on which the users of an enterprise network communicate. DHCP helps enterprises to smoothly manage the allocation of **IP addresses** to the end-user clients' devices such as desktops, laptops, cellphones, etc. is an application layer protocol that is used to provide:

```
Subnet Mask (Option 1 - e.g., 255.255.255.0)
Router Address (Option 3 - e.g., 192.168.1.1)
DNS Address (Option 6 - e.g., 8.8.8.8)
Vendor Class Identifier (Option 43 - e.g.,
'unifi' = 192.168.1.9 ##where unifi = controller)
```

DHCP is based on a **client-server model** and based on discovery, offer, request, and ACK.

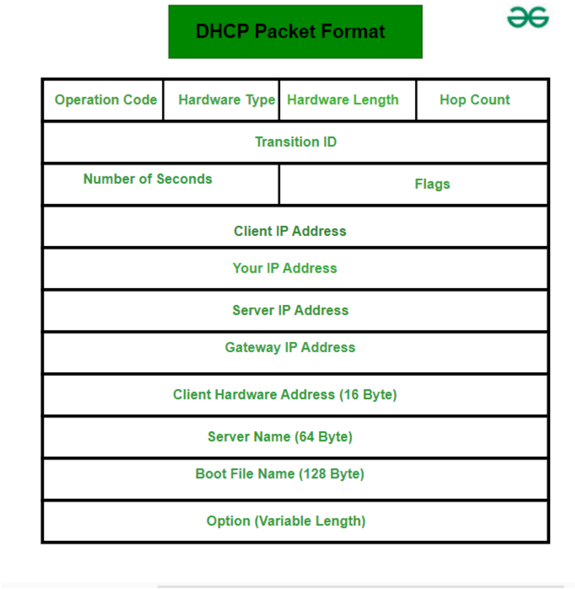
DHCP simplifies network configuration by dynamically assigning IP addresses.

Main Components of DHCP:

- **DHCP Server:** DHCP Server is a server that holds IP Addresses and other information related to configuration.

- **DHCP Client:** It is a device that receives configuration information from the server. It can be a mobile, laptop, computer, or any other electronic device that requires a connection.
- **DHCP Relay Agent:** DHCP relays basically work as a communication channel between DHCP Client and Server.
- **IP Address Pool:** It is the pool or container of IP Addresses possessed by the DHCP Server. It has a range of addresses that can be allocated to devices.

DHCP Packet Format



DHCP Packet Format

Operation Code	Hardware Type	Hardware Length	Hop Count
Transition ID			
Number of Seconds		Flags	
Client IP Address			
Your IP Address			
Server IP Address			
Gateway IP Address			
Client Hardware Address (16 Byte)			
Server Name (64 Byte)			
Boot File Name (128 Byte)			
Option (Variable Length)			

Hardware Length: An 8-bit field defining the physical address length in bytes, e.g., Ethernet is 6.

Hop Count: An 8-bit field setting the max hops a packet can travel.

Transaction ID: A 4-byte integer set by the client, used to match requests with server replies.

Number of Seconds: A 16-bit field showing seconds since the client began booting.

Flag: A 16-bit field; the leftmost bit forces a broadcast reply, the rest are set to 0.

Client IP Address: A 4-byte field for the client IP; if unknown, set to 0.

Your IP Address: A 4-byte field for the client IP, filled by the server.

Server IP Address: A 4-byte field with the server IP, set by the server in replies.

Gateway IP Address: A 4-byte field for the router IP, added by the server.

Client Hardware Address: Physical address of the client, ideally provided by the client.

Server Name: A 64-byte optional field, null-terminated, with the server's domain name. Filled with 0s if unused.

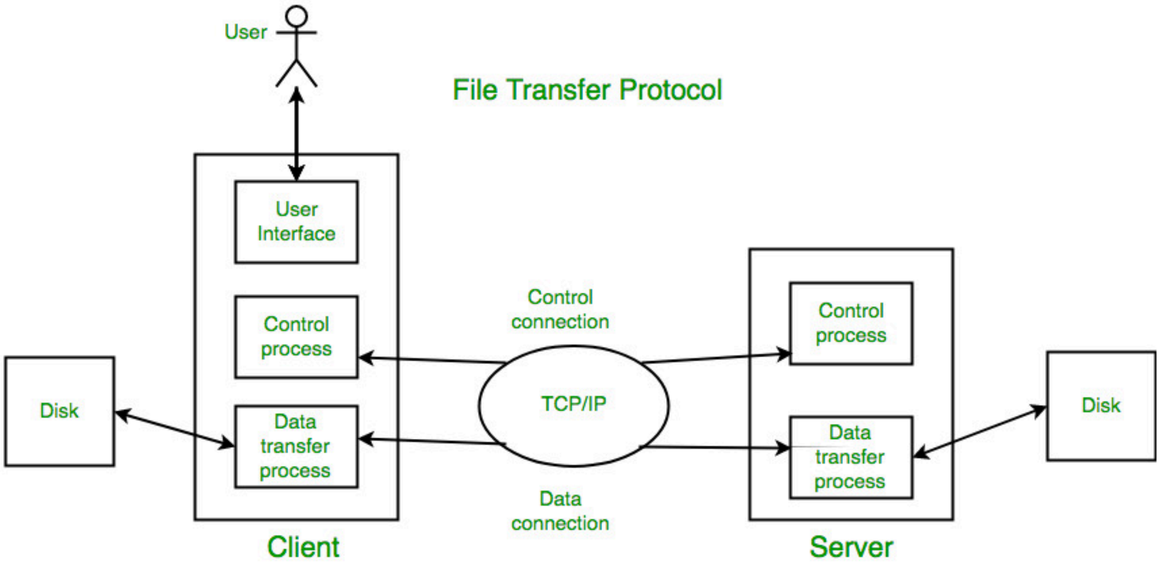
Boot Filename: A 128-byte optional field for the boot file path, null-terminated. Set to 0s if not used.

Options: A 64-byte field for additional/vendor info, used only in replies. Begins with a "magic cookie" (99.130.83.99) if options are included.

FTP (File Transfer Protocol)

Definition: FTP is a protocol for transferring files over a network, typically from a client to a server or between two remote systems.

It's one of the oldest protocols for file transfer and remains widely used, especially in web development, content management, and file sharing applications.



Types of FTP

There are different ways through which a server and a client do a file transfer using FTP. Some of them are mentioned below:

- **Anonymous FTP:** Anonymous FTP is enabled on some sites whose files are available for public access. A user can access these files without having any username or password. Instead, the username is set to anonymous, and the password is to the guest by default. Here, user access is very limited. For example, the user can be allowed to copy the files but not to navigate through directories.
- **Password Protected FTP:** This type of FTP is similar to the previous one, but the change in it is the use of username and password.
- **FTP Secure (FTPS):** It is also called as FTP Secure Sockets Layer (FTP SSL). It is a more secure version of FTP data transfer. Whenever FTP connection is established, Transport Layer Security (TLS) is enabled.
- **FTP over Explicit SSL/TLS (FTPES):** FTPES helps by upgrading FTP Connection from port 21 to an encrypted connection.
- **Secure FTP (SFTP):** SFTP is not a FTP Protocol, but it is a subset of Secure Shell Protocol, as it works on port 22.

Limitations:

- **Security**: Basic FTP lacks built-in encryption, which means that data, usernames, and passwords are sent in plain text, making it vulnerable to eavesdropping. Secure alternatives like FTPS and SFTP address this by adding encryption.
- **Firewall and NAT Compatibility**: In active mode, FTP can be blocked by firewalls or NAT due to the server-initiated connection. Passive mode, however, helps mitigate these issues.

Applications:

FTP is especially useful for:

- **Transferring Large Files:** FTP can transfer large files in one shot; thus applicable when hosting websites, backing up servers, or sharing files in large quantities.
- **Remote File Management:** Files on a remote server can be uploaded, downloaded, deleted, renamed, and copied according to the users' choices.

- **Automating File Transfers:** FTP is a great protocol for the execution of file transfers on predefined scripts and employments.
- **Accessing Public Files:** Anonymous FTP means that everybody irrespective of the identity is allowed to download some files with no permissions needed.

In essence, **DHCP** is designed to simplify IP address allocation on networks, while **FTP** facilitates file transfers across networks. Both protocols are vital in network management and internet services, each addressing a specific function in connectivity and resource sharing.

Q.5) SHORT NOTE ON TELNET

TELNET stands for Teletype Network.

It is a **client/server application protocol** that provides access to virtual terminals of remote systems on local area networks or the Internet.

The local computer uses a telnet client program and the remote computers use a telnet server program. In this article, we will discuss every point about TELNET.

TELNET is a type of protocol that enables one computer to connect to the local computer.

It is used as a standard **TCP/IP protocol** for virtual terminal service which is provided by **ISO**.

The computer which starts the connection is known as the **local computer**.

The computer which is being connected to i.e. which accepts the connection known as the **remote computer**.

During telnet operation, whatever is being performed on the remote computer will be displayed by the local computer. Telnet operates on a client/server principle.

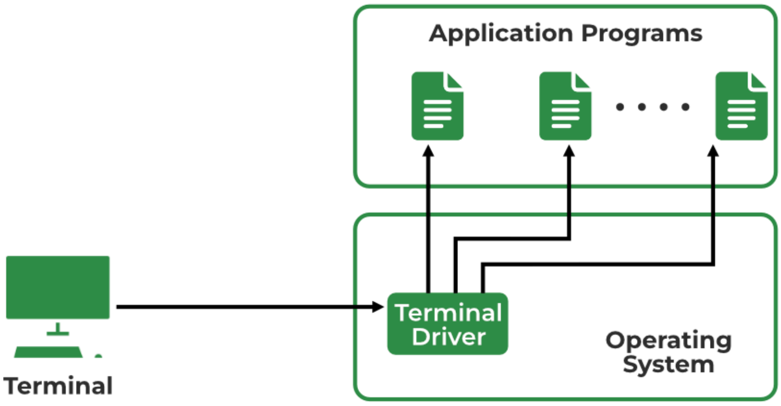
Logging in TELNET

The logging process can be further categorized into two parts:

- Local Login
- Remote Login

1. Local Login

Whenever a user logs into its local system, it is known as local login.



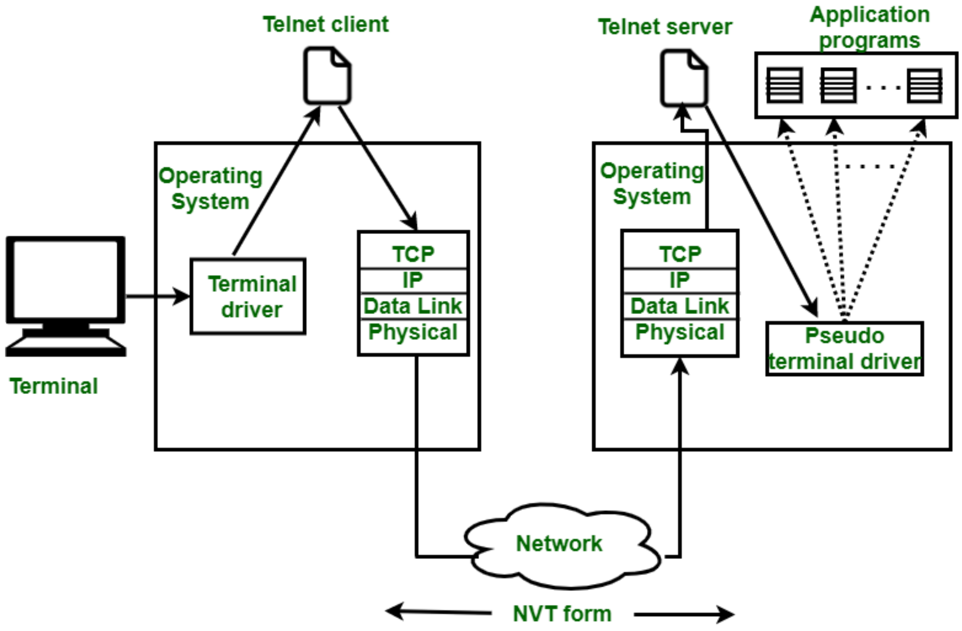
The Procedure of Local Login

- Keystrokes are accepted by the terminal driver when the user types at the terminal.
- Terminal Driver passes these characters to OS.
- Now, OS validates the combination of characters and opens the required application.

2. Remote Login

Remote Login is a process in which users can log in to a remote site i.e. computer and use services that are available on the remote computer.

With the help of remote login, a user is able to understand the result of transferring the result of processing from the remote computer to the local computer.



The Procedure of Remote Login

- When the user types something on the local computer, the local operating system accepts the character.
- The local computer does not interpret the characters, it will send them to the TELNET client.
- TELNET client transforms these characters to a universal character set called Network Virtual Terminal (NVT) characters and it will pass them to the local TCP/IP protocol Stack.
- Commands or text which are in the form of NVT, travel through the Internet and it will arrive at the **TCP/IP** stack at the remote computer.
- Characters are then delivered to the operating system and later on passed to the TELNET server.
- Then TELNET server changes those characters to characters that can be understandable by a remote computer.
- The remote operating system receives characters from a pseudo-terminal driver, which is a piece of software that pretends that characters are coming from a terminal.
- The operating system then passes the character to the appropriate application program.