HTTP (HyperText Transfer Protocol)

- **Layer**: Application layer of TCP/IP.
- **Purpose**: Protocol for the request and transfer of data over the web.

Key Properties

1. **Connectionless**:
    - The client terminates the connection with the server after receiving a response.
    - **Advantages**: Saves resources by not maintaining connections.
    - **Disadvantages**: Inefficient for multiple requests from the same client.
2. **Stateless**:
    - The server does not retain information about the client.
    - To maintain a stateful connection, information is stored using:
        - **Cookies**: Key-value data stored in the client's browser (e.g., login info, shopping carts).
        - **Sessions**: Information stored on the server, offering more security but with potential server-side overhead.

HTTP Message Structure

- **Request Line/URI**
- **Status Line (status code)**
- **Header**
- **Blank Line** (Divides the header and body)
- **Body** (Usually empty except for POST requests)

Common Status Codes

- **200**: Success
- **4xx**: Client-side errors (e.g., 404 - Not Found)
- **5xx**: Server-side errors (e.g., 500 - Internal Server Error)

HTTPS (HyperText Transfer Protocol Secure)

- HTTP with enhanced security through **SSL/TLS encryption**.

Encryption Protocols

- **SSL (Secure Socket Layer)**: Original encryption protocol for secure communications.
- **TLS (Transport Layer Security)**: The modern, upgraded version of SSL.
- SSL/TLS operates between the **Application layer** and **Transport layer**.

Connecting to a Webpage: Process Overview

1. **DNS Resolution**:
    - The user-entered URL is resolved by the **Domain Name Server (DNS)**, which returns the corresponding IP address.
2. **TCP/IP Connection and HTTP Request**:
    - The browser establishes a connection using the provided IP address and sends an **HTTP request**.
3. **Server Response**:
    - The web server processes the request and replies.
4. **Content Display**:
    - The browser displays the response content.