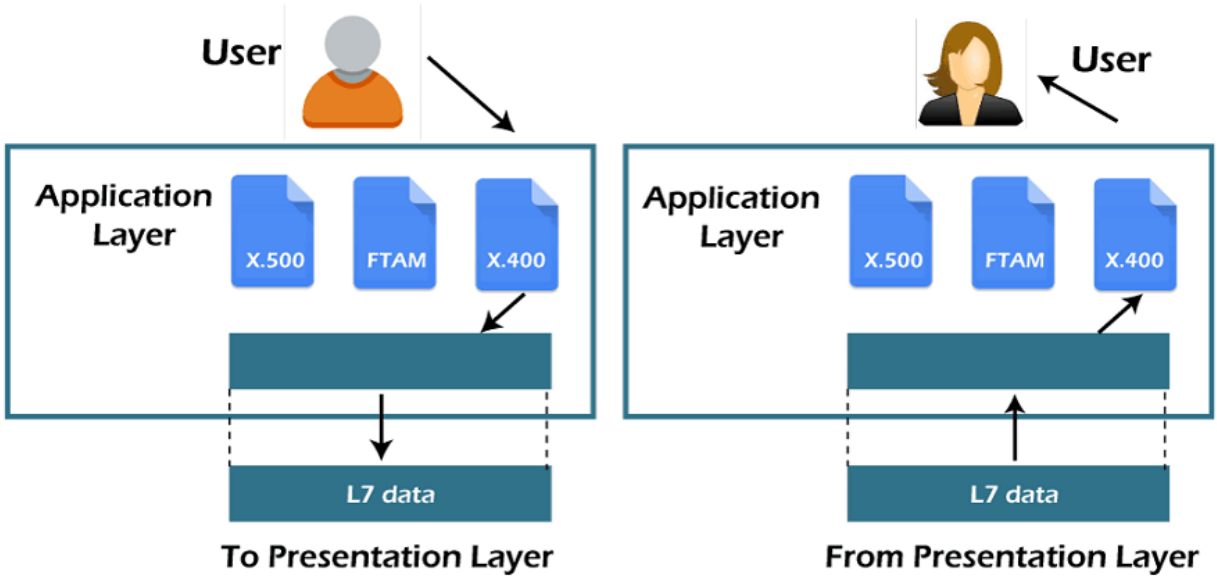
Unit-02

Application Layer

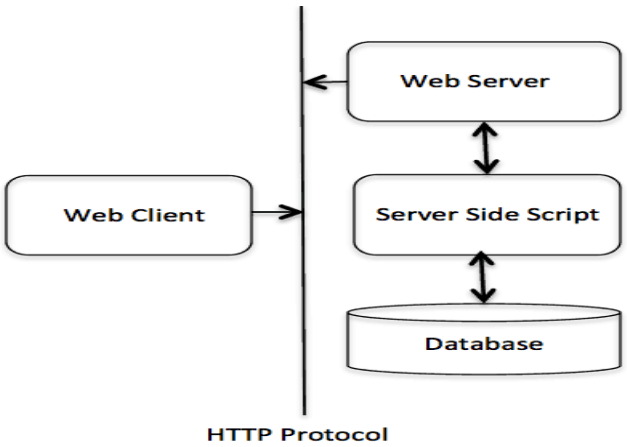
Application Layer

Application layer is the top most layer of the internet model. This is the layer where all application or software are found in this layer. Application layer allows the user to use the internet. it bases on client server model.



Over view of HTTP

HTTP is a protocol for fetching resources such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.



Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called requests and the messages sent by the server as an answer are called responses.

HTTP is an extensible protocol which has evolved over time. It is an application layer protocol that is sent over TCP, or over a TLS (Transport Layer Security,)- encrypted TCP connection, though any reliable transport protocol could theoretically be used. Due to its extensibility, it is used to not only fetch hypertext documents, but also images and videos or to post content to servers, like with HTML form results.

Http message formate

HTTP (Hypertext Transfer Protocol) messages are used to exchange data between a client (such as a web browser) and a server. HTTP messages are composed of textual information encoded in ASCII, and span over multiple lines. These messages come in two types: requests and responses. Both types of messages share a common structure consisting of a start line, headers, a blank line (CRLF), and an optional message body.

The start-line and HTTP headers of the HTTP message are collectively known as the head of the requests, whereas its payload is known as the body.

HTTP Request Message *Structure:*

<request-line>
<headers>

<optional message body>

HTTP Response Message Structure:

<status-line>
<headers>

<optional message body>

User-Server Interaction

User-server interaction involves a client (usually a web browser or application) making requests to a server, and the server responding with the requested resources or information. This interaction is fundamental to web browsing, accessing online services, and many other Internet-based activities.

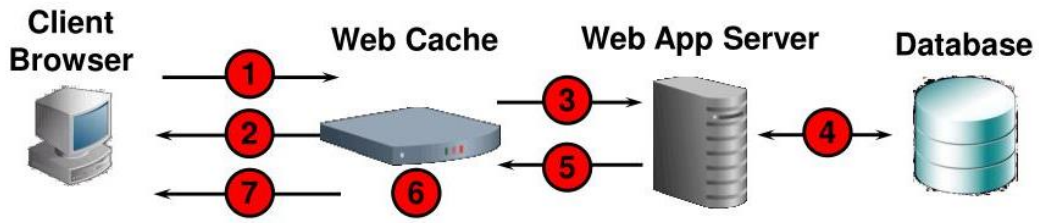
Cookies

Cookies are small pieces of data stored on a user's device by their web browser at the request of a web server. They are used to remember information about the user, such as login status, preferences, and tracking information, to provide a more personalized and efficient browsing experience. There is various type of Cookies: Session Cookies, Persistent Cookies, First-Party Cookies, Third-Party Cookies.

Web Caching

Web caching is the process of storing copies of web resources (such as HTML pages, images, and other media) in a cache, or temporary storage location, so that future requests for those resources can be served faster. By reducing the need to repeatedly fetch the same data from the original server, web caching improves the performance and efficiency of web applications. There is various type of web caching: Browser Caching, Proxy Caching, Content Delivery Network (CDN) Caching, Server-Side Caching.

How Web Cache Works



Electronic Mail in the Internet:

Email is a fundamental service on the Internet, enabling users to send and receive messages. The process involves several protocols and standards to ensure that messages are properly formatted, transmitted, and accessed.

Moden Email has many powerful features like:

- A message with attachments
- Hyperlinks • HTML formatted Text
- Embedded Photos, Audio, Video, Animation etc.

SMTP (Simple Mail Transfer Protocol)

SMTP is the protocol used for sending emails between servers. It is a push protocol that handles the transmission of email messages from the sender's mail server to the recipient's mail server.

SMTP Process:

- **Client to Server**: The sender's email client connects to the SMTP server to send the email.
- **Server to Server**: The SMTP server relays the email to the recipient's mail server.
- **Server to Client**: The recipient's email server stores the email until the recipient retrieves it.

Mail Message Formats

Email messages have a specific format defined by standards such as RFC 5322. The format includes headers and the body of the message.

Email Headers:

- **From**: The sender's email address.
- **To**: The recipient's email address.
- **Subject**: A brief summary of the email content.
- **Date**: The date and time the email was sent.
- **CC**: Carbon copy, additional recipients.
- **BCC**: Blind carbon copy, additional recipients not visible to other recipients.
- **Message-ID**: A unique identifier for the email.

Email Body:

- **Plain Text**: The main content of the email in plain text format.
- **HTML**: Optional, allows rich text formatting and multimedia content.

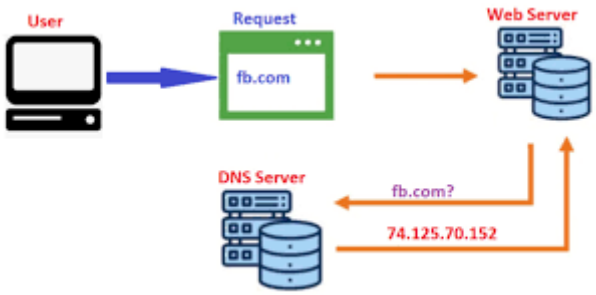
Mail Access Protocols

Mail access protocols are used by email clients to retrieve messages from the mail server. The two primary protocols are POP3 and IMAP. POP3 (Post Office Protocol 3) is used to download emails from the server to the client and optionally delete them from the server and IMAP (Internet Message Access Protocol) is use to access and manage email messages directly on the server.

DNS—The Internet's Directory Service

The Domain Name System (DNS) is a hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It translates human-readable domain names (e.g., www.example.com) into IP addresses (e.g., 192.0.2.1) that computers use to identify each other on the network. DNS operates like a phonebook for the Internet by converting a friendly domain name into an IP address.

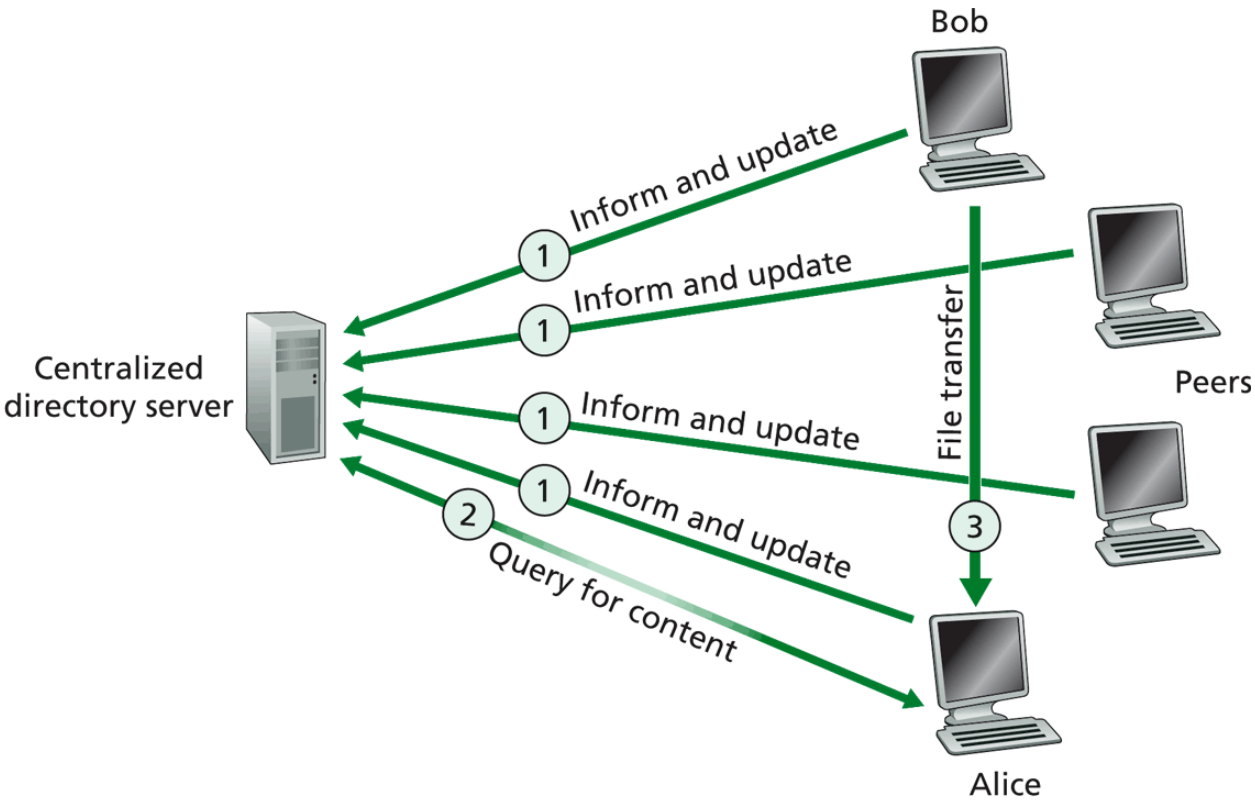
How DNS Works:



User Request: User types a domain name in the browser.
Cache Check: Browser checks its cache for the IP address.
Local DNS Resolver: If not found, browser queries the local DNS resolver (usually provided by the ISP).
Root Server: Resolver queries a root DNS server, which directs it to the appropriate TLD server.
TLD Server: Resolver queries the TLD server (e.g., for .com), which directs it to the authoritative DNS server.
Authoritative Server: Resolver queries the authoritative DNS server, which provides the IP address.
Response: Resolver returns the IP address to the browser.
Connection: Browser uses the IP address to connect to the web server and retrieve the webpage

Peer-to-Peer File Distribution

Peer-to-peer (P2P) file distribution is a decentralized method of sharing files where each participant (peer) acts as both a client and a server. P2P file distribution is efficient, scalable, and reduces the load on individual servers by leveraging the bandwidth and storage capacities of all participating peers.



Content Delivery Network

A Content Delivery Network (CDN) is a network of distributed servers that deliver web content to users based on their geographic location, the origin of the content, and a content delivery server. CDNs aim to improve the performance, reliability, and security of websites and web applications by caching content closer to end-users.

Video Streaming and Content Distribution Networks

1. Video Encoding and Storage

Video Capture: The video is recorded or created.
Encoding: The video is compressed and encoded into various formats and bitrates suitable for streaming.
The encoded video files are stored on the origin server.

2. Setting Up CDN (Content Distribution Network)

CDN Selection: Choose a CDN provider based on your needs (e.g., Akamai, Cloudflare, Amazon CloudFront).
CDN Configuration: Configure the CDN to cache and deliver your video content. This involves setting up CDN endpoints and specifying caching rules.

3. Content Distribution to Edge Servers

Replication: The CDN replicates video content from the origin server to multiple edge servers distributed geographically.
Cache Propagation: The video files are cached on these edge servers to ensure they are readily available for viewers.

4. User Requests Video

Request Initiation: A user requests to watch a video by clicking a link or opening a video player on a website or app.
DNS Resolution: The request is directed to the nearest CDN edge server through DNS resolution, based on the user's location.

5. Edge Server Delivers Content

Cache Check: The edge server checks if it has the requested video segments cached.
Cache Hit: If the segments are cached, the edge server delivers them directly to the user.
Cache Miss: If the segments are not cached, the edge server retrieves them from another edge server or the origin server, caches them, and then delivers them to the user.

6. Playback on User's Device

Buffering: The video player buffers a few seconds of video to ensure smooth playback.
Playback: The video is played on the user's device. The player continues to download and buffer video segments from the CDN as needed.

7. Analytics and Monitoring

Data Collection: The CDN and the video service collect data on video performance (e.g., playback quality, buffering events).
Monitoring: Real-time monitoring and analytics help identify and resolve performance issues, optimize content delivery, and enhance user experience.