What happen when we type URL in the browser

When you type a URL (such as `facebook.com`) into your web browser and press enter, a series of complex and fascinating steps happen behind the scenes to fetch and display the web page. This process involves several key components and technologies working together to deliver the requested data from a remote network to your computer. As a student of Computer Network, it’s very essetnial to understand what actually is going under the hood.

Application Layer (Layer 7)

Step 1: User Action

- **What You Do**: You open your web browser (like Chrome, Firefox, or Safari) and type "facebook.com" into the address bar, then press Enter.
- **What's Happening**: You're requesting to visit Facebook's website.

Step 2: DNS Request

- **Why It Happens**: Computers don't understand website names like "facebook.com"; they need an IP address (a numerical label like 157.240.22.35) to find and connect to the website.
- **How It Works**:
    1. **DNS (Domain Name System)**: Think of DNS as the internet's phonebook. When you type "facebook.com", your browser asks a DNS server to translate that name into an IP address.
    2. **Browser Sends DNS Request**: Your browser sends a request to a DNS server (usually provided by your Internet Service Provider or a public DNS service like Google's or Cloudflare's) asking, "What is the IP address for facebook.com?"

Step 3: Receiving the DNS Response

- **DNS Server Replies**: The DNS server looks up the IP address for "facebook.com" and sends it back to your browser. For example, it might reply with "The IP address for facebook.com is 157.240.22.35".

Step 4: Browser Prepares to Connect

- **Using the IP Address**: Now that your browser knows the IP address, it can proceed to make a connection to Facebook's servers using this IP address.

DNS Resolution (Layer 7)

Step 1: DNS Query

- **Browser Checks DNS Cache**:
  - **What Happens**: Your browser first checks its own cache to see if it has recently looked up "facebook.com" and already knows the IP address. If it does, it uses that IP address immediately.
  - **Why It Matters**: This cache check speeds up the process by avoiding unnecessary network requests.
- **DNS Query Sent to Resolver**:
  - **What Happens**: If the IP address isn't cached in the browser, the browser sends a DNS query to a DNS resolver. This resolver is usually provided by your Internet Service Provider (ISP) or can be a public DNS resolver like Google's (8.8.8.8) or Cloudflare's (1.1.1.1).
  - **Why It Matters**: The DNS resolver acts like a middleman that helps find the IP address you need.

Step 2: Recursive Query

- **What Happens**:
  - The DNS resolver may not know the IP address right away. Instead, it will perform a series of recursive queries, asking multiple DNS servers until it finds the one that has the information. This process involves several steps:
    1. **Root DNS Server**: The resolver first asks a root DNS server, "Where can I find the .com DNS servers?"

2. **Top-Level Domain (TLD) DNS Server**: The root DNS server responds with the IP addresses of the TLD DNS servers responsible for ".com" domains. The resolver then asks one of these servers, "Where can I find the DNS servers for facebook.com?"
3. **Authoritative DNS Server**: The TLD DNS server responds with the IP addresses of the authoritative DNS servers for "facebook.com." The resolver then queries one of these authoritative servers, "What is the IP address for facebook.com?"

Step 3: Response

- **DNS Server Responds**:
  - **What Happens**: The authoritative DNS server for "facebook.com" responds to the resolver with the IP address of Facebook's web server (e.g., 157.240.229.35).
  - **Why It Matters**: Now the resolver knows the correct IP address and can pass this information back to your browser.

Step 4: Browser Receives IP Address

- **Using the IP Address**:
  - **What Happens**: The DNS resolver sends the IP address back to your browser. Now, your browser knows where to send the request to load Facebook's website.
  - **Why It Matters**: With the IP address, your browser can connect to Facebook's server and start loading the webpage you requested.

Network Layer (Layer 3) and Transport Layer (Layer 4)

Step 1: IP Address

- **Browser Knows the IP Address**:
  - **What Happens**: Your browser now has the IP address for Facebook (e.g., 157.240.229.35) after completing the DNS resolution.
  - **Why It Matters**: This IP address allows your browser to locate Facebook's server on the internet.

Step 2: TCP Connection

- **Initiating a TCP Connection**:
  - **What Happens**: Your browser needs to establish a stable connection with Facebook's server to ensure reliable data transfer. This is done using the Transport Layer's TCP (Transmission Control Protocol).
  - **Port 443 (HTTPS)**: Since you're accessing a secure website, the browser will connect to Facebook's server on port 443, which is the standard port for HTTPS.

Step 3: Three-Way Handshake
To establish a reliable TCP connection, the browser and Facebook's server perform a three-way handshake:

- **SYN (Synchronize) Packet**:
  - **What Happens**: Your browser sends a SYN packet to Facebook's server. This packet basically says, "Hey, I want to start a conversation and here is my initial sequence number."
  - **Purpose**: This step initiates the connection and synchronizes the sequence numbers used for data transfer.
- **SYN-ACK (Synchronize-Acknowledge) Packet**:
  - **What Happens**: Facebook's server responds with a SYN-ACK packet. This packet says, "Got it! Let's synchronize. Here is my sequence number and I acknowledge your sequence number."
  - **Purpose**: This step acknowledges the receipt of the SYN packet from the client and includes the server's own initial sequence number.
- **ACK (Acknowledge) Packet**:
  - **What Happens**: Your browser sends an ACK packet back to Facebook's server. This packet says, "Acknowledged your sequence number. Let's start communicating."
  - **Purpose**: This final step completes the handshake, and the connection is established. Both sides have agreed on the initial sequence numbers and are ready to start transferring data.

Data Link Layer (Layer 2) and Physical Layer (Layer 1)

Step 1: Frame Creation (Data Link Layer)

- **Encapsulation into Frames**:
  - **What Happens**: The data that needs to be sent (HTTP/HTTPS request) is encapsulated into a frame. This frame includes important information like source and destination MAC addresses (unique hardware addresses of network interfaces).
  - **Source MAC Address**: This is the MAC address of your computer's network interface.
  - **Destination MAC Address**: This is the MAC address of the next device in the network path (e.g., your router).
- **Why It Matters**: The frame ensures that the data can be correctly addressed and managed on the local network. It includes error-checking information to ensure the data isn't corrupted during transmission.

Step 2: Physical Transmission (Physical Layer)

- **Conversion to Signals**:
  - **What Happens**: The frames created by the Data Link Layer are converted into physical signals. These signals can be electrical pulses for wired connections (Ethernet), light pulses for fiber optic connections, or radio waves for wireless connections (Wi-Fi).
  - **Transmission Medium**: The type of signals used depends on the physical medium:
    - **Ethernet**: Uses electrical signals transmitted over copper cables.
    - **Fiber Optic**: Uses light pulses transmitted through fiber optic cables.
    - **Wi-Fi**: Uses radio waves transmitted through the air.
- **Why It Matters**: Converting frames into signals is essential for the actual transmission of data over physical media, allowing your request to travel from your computer to the next device (e.g., your router) and onward towards Facebook's server.

HTTP/HTTPS Request

Step 1: Secure Communication

- **Using HTTPS**:
  - **What Happens**: When you visit a secure website like Facebook, your browser uses HTTPS (HyperText Transfer Protocol Secure) instead of plain HTTP. HTTPS ensures that all data sent between your browser and the website is encrypted.
  - **Encryption with SSL/TLS**: HTTPS uses SSL (Secure Sockets Layer) or its successor TLS (Transport Layer Security) to encrypt the data. This means that anyone intercepting the data cannot read it because it appears as gibberish without the proper decryption keys.
- **Why It Matters**: Encryption ensures that sensitive information, such as your login credentials and personal data, is protected from eavesdroppers and hackers while being transmitted over the internet.

Step 2: Request Message

- **Encrypted HTTP GET Request**:
  - **What Happens**: Your browser prepares an HTTP GET request, which is a type of message that asks the server to send back the contents of a specific web page (in this case, the Facebook homepage). This GET request is encrypted using SSL/TLS.
  - **Sending the Request**: The browser sends this encrypted HTTP GET request to the IP address of Facebook's server.
- **Why It Matters**: By sending an encrypted request, your browser ensures that the communication remains private and secure, protecting the data from being tampered with or read by unauthorized parties.

Step 1: Server Processing

- **Processing the Request**:
  - **What Happens**: When Facebook's server receives your encrypted HTTP GET request, it decrypts the request using its SSL/TLS keys. The server

then processes the request to fetch the necessary data needed to display the Facebook homepage.

    - **Fetching Data**: This includes:
        - **HTML**: The structure and content of the web page.
        - **CSS**: The styling information (colors, fonts, layout).
        - **JavaScript**: Scripts that add interactivity and functionality to the page.
        - **Images**: Any pictures, icons, or graphics needed for the page.
- **Why It Matters**: The server must gather all these resources to build and display the web page correctly on your browser.

Step 2: Response Message

- **Sending the Response**:
    - **What Happens**: Once the server has collected all the necessary resources, it creates an HTTP response message. This message contains the requested resources (HTML, CSS, JavaScript, images, etc.). The server then encrypts the response using SSL/TLS to ensure secure communication back to your browser.
    - **Encryption**: Just like the request, the response is encrypted to protect the data as it travels over the internet, ensuring privacy and security.
- **Why It Matters**: Encrypting the response ensures that the data remains secure and private while being transmitted back to your browser. This prevents eavesdropping and tampering by unauthorized parties.

Data Transmission and Reception

Step 1: Transport Layer (Layer 4)

- **Breaking Down the HTTP Response**:
    - **What Happens**: The HTTP response from Facebook's server is too large to be sent all at once. It's broken down into smaller packets.
    - **TCP (Transmission Control Protocol)**: TCP is responsible for ensuring that these packets are delivered reliably and in the correct order. It assigns

sequence numbers to the packets so they can be reassembled correctly on the receiving end.

    - **Error Handling**: If any packets are lost or corrupted during transmission, TCP handles retransmissions, ensuring that all data arrives intact.

- **Why It Matters**: TCP ensures that the entire HTTP response is delivered accurately and completely, even if the data is split across multiple packets.

Step 2: Network Layer (Layer 3)

- **Adding IP Addresses**:
    - **What Happens**: Each packet is given a header that includes the source IP address (Facebook's server) and the destination IP address (your computer). This allows the packets to be routed correctly across the internet.
    - **Routing**: The packets are sent through a series of routers and other network devices that use these IP addresses to determine the best path to your computer.
- **Why It Matters**: IP addresses ensure that the packets are routed from Facebook's server to your computer across the complex network of the internet.

Step 3: Data Link Layer (Layer 2) and Physical Layer (Layer 1)

- **Frame Creation**:
    - **What Happens**: At each hop along the way (e.g., from one router to another), the packets are encapsulated into frames, which include source and destination MAC addresses. These addresses are specific to each network interface on the local network.
    - **Error-Checking**: Frames also include error-checking information to detect and correct any errors that occur during transmission.
- **Transmission Over Physical Medium**:
    - **What Happens**: Frames are converted into physical signals appropriate for the transmission medium:

        - **Ethernet**: Frames are converted into electrical signals that travel over copper cables.
        - **Fiber Optic**: Frames are converted into light pulses that travel through fiber optic cables.
        - **Wi-Fi**: Frames are converted into radio waves that travel through the air.
    - **Transmission and Reception**: These signals are sent over the physical medium, received by the next device in the network path, and converted back into frames and then packets.
- **Why It Matters**: The Data Link and Physical Layers handle the actual physical transmission of data, ensuring that packets travel across the network medium and reach their destination.

Rendering the Web Page

Step 1: Browser Processing

- **Receiving the HTTP Response**:
    - **What Happens**: Your browser receives the encrypted HTTP response from Facebook's server. This response contains the HTML, CSS, JavaScript, and other resources needed to display the Facebook homepage.
    - **Decrypting the Response**: The browser decrypts the response using SSL/TLS to access the contents securely.
- **Parsing the Resources**:
    - **HTML**: The browser parses the HTML to understand the structure and content of the web page. This includes elements like headings, paragraphs, images, and links.
    - **CSS**: The browser processes the CSS to apply styles to the HTML elements, defining how they should look (colors, fonts, layout, etc.).
    - **JavaScript**: The browser executes any JavaScript code to add interactivity and functionality to the web page (e.g., loading new content without refreshing the page).
- **Rendering the Page**:

- **What Happens**: The browser combines the parsed HTML, CSS, and JavaScript to render the Facebook homepage on your screen. This process involves constructing the Document Object Model (DOM) and applying styles and scripts to create the final visual representation.
- **Why It Matters**: This step transforms the raw data from the server into a visually appealing and interactive web page that you can see and interact with.

Step 2: User Interaction

- **Interacting with the Web Page**:
  - **What Happens**: Once the Facebook homepage is rendered, you can interact with it in various ways, such as liking a post, posting a comment, or navigating to another page.
  - **Generating HTTP Requests**: Each interaction (like clicking a "like" button or submitting a comment) generates further HTTP requests to Facebook's server. These requests follow the same process as before: they are sent to the server, processed, and the server's responses are rendered by the browser.
- **Why It Matters**: User interactions are essential for dynamic web experiences, allowing you to engage with content and perform actions that update the web page in real time.