What Happens When We Type google.com and Press Enter?

Aayush Adhikari

10th *June*, 2024

Introduction

- This presentation explains the step-by-step process that occurs when we type "google.com" and press enter.
- We will cover DNS resolution, establishing a connection, HTTP request and response, server processing, and webpage rendering.

Introduction Description

- ▶ Understanding these processes is crucial for web development and networking.
- ▶ We will use analogies and diagrams to simplify the technical concepts.
- ▶ By the end of this presentation, we will have a clear understanding of the entire process.
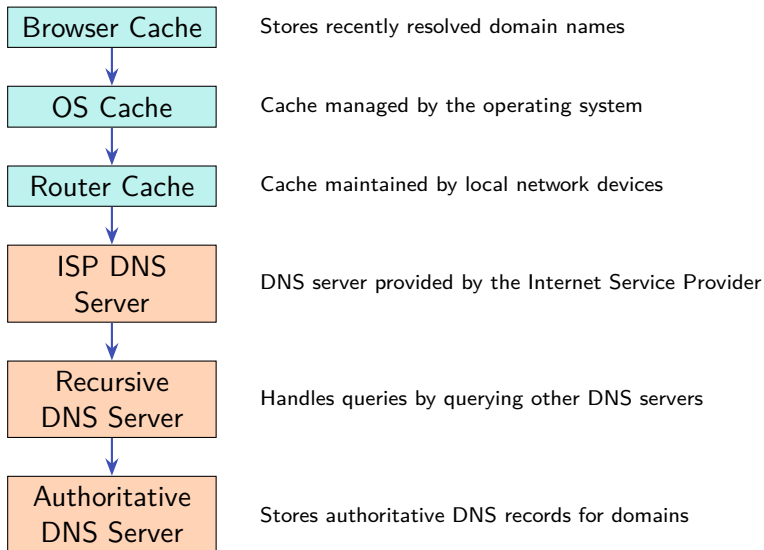
DNS Resolution

- ▶ DNS (Domain Name System) translates domain names into IP addresses.
- ▶ Steps involved in DNS resolution include checking browser cache, OS cache, router cache, ISP's DNS server, recursive DNS servers, and authoritative DNS servers.

DNS Resolution Description

- DNS acts like a phone book for the internet, mapping human-readable domain names to machine-readable IP addresses.
- When we type "google.com", the browser first checks its own cache to see if it already knows the IP address.
- If not, it queries the OS cache, then the router cache, and so on until it reaches an authoritative DNS server.

DNS Resolution Process

Browser Cache	Stores recently resolved domain names
↓	
OS Cache	Cache managed by the operating system
↓	
Router Cache	Cache maintained by local network devices
↓	
ISP DNS Server	DNS server provided by the Internet Service Provider
↓	
Recursive DNS Server	Handles queries by querying other DNS servers
↓	
Authoritative DNS Server	Stores authoritative DNS records for domains

Establishing a Connection

- ▶ Once the IP address is obtained, the browser establishes a connection using TCP/IP.
- ▶ This involves a three-way handshake:
  - ▶ SYN: Client sends a synchronization packet to the server.
  - ▶ SYN-ACK: Server acknowledges with a synchronization-acknowledgment packet.
  - ▶ ACK: Client sends an acknowledgment packet back to the server.
- ▶ For secure connections, the TLS/SSL handshake is also performed.

Establishing a Connection Description

- TCP/IP ensures reliable communication between the client and server.
- The three-way handshake sets up a reliable connection:
    - SYN: "Hello, I'd like to connect."
    - SYN-ACK: "Hello, I acknowledge wer request."
    - ACK: "Great, let's start communicating."
- For HTTPS, additional steps include exchanging encryption keys to secure the communication.

TCP Three-Way Handshake

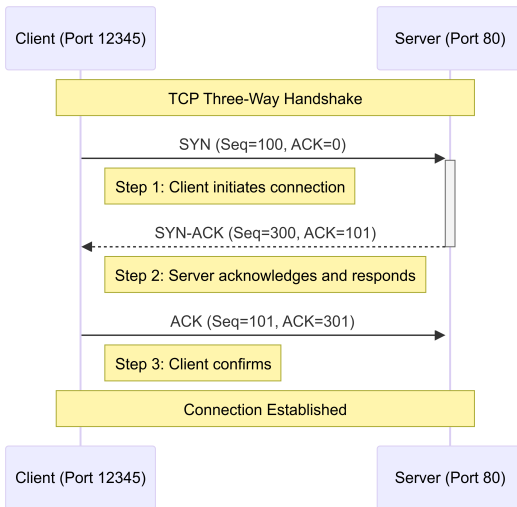| Client	Step	Server
Port: 12345		Port: 80
Sends SYN Seq=x	1	
	2	Sends SYN-ACK Seq=y, ACK=x+1
Sends ACK Seq=x+1, ACK=y+1	3	
Connection Established		

Three-Way Handshake Diagram

Figure: Three-Way Handshake

HTTP Request

- ▶ After the connection is established, the browser sends an HTTP request to the server.
- ▶ Structure of an HTTP request:
  - ▶ Request line (e.g., GET / HTTP/1.1).
  - ▶ Headers (e.g., Host, User-Agent).
  - ▶ Body (for POST requests).
- ▶ Common HTTP methods: GET, POST, PUT, DELETE.

HTTP Request Description

- ▶ The HTTP request is like asking a librarian for a book:
  - ▶ Request line: "I want to get the homepage (GET /)."
  - ▶ Headers: Additional information about the request, like who is asking (User-Agent).
  - ▶ Body: The actual data sent to the server (for POST requests).
- ▶ GET requests fetch data, while POST requests send data to the server.

HTTP Request Example

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

Server Processing

- The server receives the request and processes it.
- Server processing involves:
    - Interpreting the request.
    - Retrieving data from databases or other sources.
    - Generating a response.

Server Processing Description

- The server acts like a chef in a restaurant:
  - Interpreting the request: Understanding what dish (data) we want.
  - Retrieving data: Gathering the ingredients (data) from the kitchen (database).
  - Generating a response: Preparing and serving the dish (response) back to we.
- The server uses backend logic and databases to create the appropriate response.

HTTP Response

- ▶ The server sends an HTTP response back to the client.
- ▶ Structure of an HTTP response:
    - ▶ Status line (e.g., HTTP/1.1 200 OK).
    - ▶ Headers (e.g., Content-Type, Content-Length).
    - ▶ Body (the actual content).
- ▶ Common status codes:
    - ▶ 200 OK: The request was successful.
    - ▶ 404 Not Found: The requested resource was not found.
    - ▶ 500 Internal Server Error: A server error occurred.

HTTP Response Description

▶ The HTTP response is like the librarian giving we the book:
  ▶ Status line: Indicates whether the request was successful (200 OK) or if there was an error (404 Not Found).
  ▶ Headers: Additional information about the response, such as the type of content being sent (Content-Type) and its length (Content-Length).
  ▶ Body: The actual data or content requested (e.g., HTML of the webpage).

HTTP Response Example

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1234

<html>
<head>
<title>Google</title>
</head>
<body>
<h1>Welcome to Google</h1>
</body>
</html>
```

Rendering the Webpage

- The browser processes the received HTML, CSS, and JavaScript to render the webpage.
- The rendering process involves:
  - Parsing HTML to construct the DOM (Document Object Model).
  - Applying CSS to style the elements.
 - Executing JavaScript to add interactivity.

Rendering the Webpage Description

- The browser acts like an artist creating a painting:
  - Parsing HTML: Sketching the basic outline (structure) of the painting.
  - Applying CSS: Adding colors and styles to the sketch.
  - Executing JavaScript: Adding moving parts or interactivity to the painting.
- This process ensures that the webpage appears as intended by the web designers.

Rendering Process
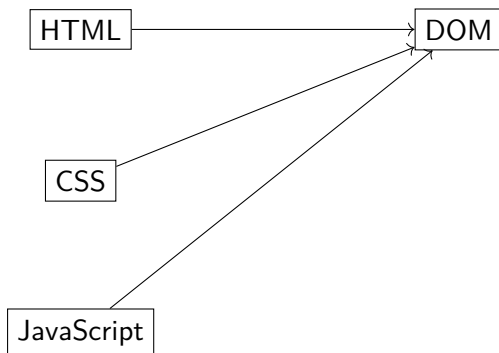
Sequence Diagram
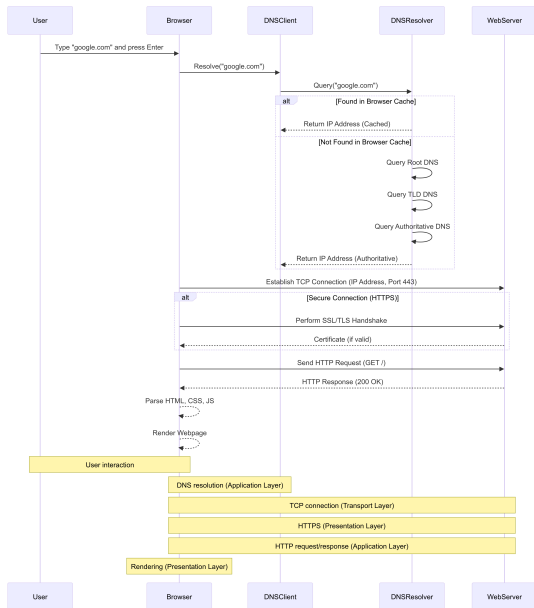
Conclusion

- We have covered the complete process from typing a URL to displaying a webpage.
- Each step is crucial for ensuring that web content is delivered accurately and efficiently.
- Understanding this process helps in diagnosing issues and optimizing web performance.

Conclusion

- Recap of the steps:
  - DNS resolution to find the IP address.
  - Establishing a connection using TCP/IP.
  - Sending an HTTP request to the server.
  - Server processing the request and sending back an HTTP response.
  - Browser rendering the webpage using HTML, CSS, and JavaScript.
- This process is repeated millions of times every day, making the internet an integral part of our lives.