Domain Name System (DNS)

Ashish Pratap Singh

3 min read

magine trying to remember the IP address of every website you want to visit—like having to memorize a phone number for every friend instead of just calling by name.

hankfully, the Internet has a solution: the **Domain Name ystem (DNS)**.

hink of DNS as the phone book of the Internet, translating easy-to-remember domain names (like `www.example.com`) into machine-friendly IP addresses (like `93.184.216.34`).

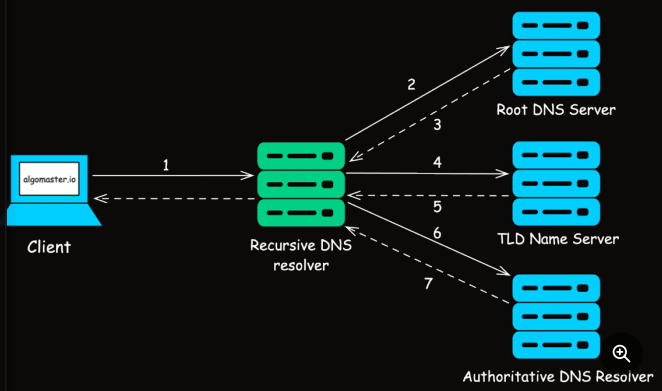
he **Domain Name System (DNS)** is a distributed, hierarchical system designed to translate human-readable domain names into IP addresses, which are necessary for locating and communicating with servers on the Internet.

nstead of typing `93.184.216.34` into your browser, you imply enter , and DNS takes care of finding the corresponding IP address.

1. How DNS Works

When you enter a domain name in your browser, a series of steps occur to translate that name into an IP address.

Here's a simplified breakdown of the process:



1. **User Request:** You type into your browser.
2. **DNS Resolver (Recursive Resolver):** Your computer sends the request to a DNS resolver provided by your Internet Service Provider (ISP) or a public DNS service (like Google DNS or Cloudflare).
3. **Root Name Server Query:** The resolver first queries a root name server, which doesn't know the exact IP but

can direct the resolver to the appropriate Top-Level Domain (TLD) server.

4. **TLD Name Server Query:** The resolver then contacts the TLD server (for `.com` in our example) to get further guidance.
5. **Authoritative Name Server Query:** Finally, the resolver queries the authoritative name server for to obtain the precise IP address.
6. **Response to User:** The IP address is returned to your browser, and you're directed to the website.

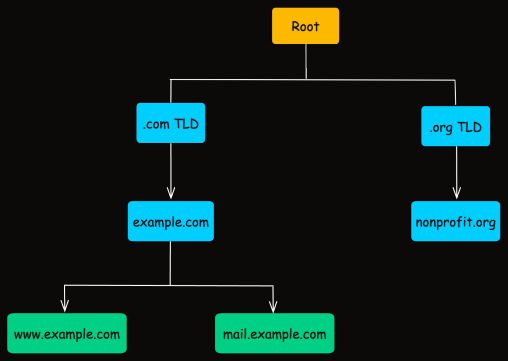
Recursive vs. Iterative Queries

- **Recursive Query:** The resolver takes full responsibility for retrieving the requested IP address, making multiple queries on behalf of the client until it gets an answer.
- **Iterative Query:** The resolver may receive partial answers and is directed to the next server in the hierarchy, with each server providing the best answer it can.

---

2. The Hierarchical Structure of DNS

DNS is designed as a hierarchical system, much like a tree:



- **Root Level:** At the top are the root servers, which know where to find the TLD servers.
- **Top-Level Domains (TLDs):** These are the next layer, like `.com`, `.org`, `.net`, etc.
- **Second-Level Domains:** These are the domain names you register (e.g., `example` in `example.com`).
- **Subdomains:** These can be added to further organize the domain (e.g., `www` or `blog`).

This hierarchy allows DNS to scale efficiently and handle a massive number of queries simultaneously.

---

3. DNS Components and

Their Roles

Root Name Servers

- **Role:** The starting point for DNS resolution. They direct queries to the appropriate TLD servers.
- **Quantity:** There are 13 logical root servers (labeled A through M), operated by various organizations worldwide.

Top-Level Domain (TLD) Servers

- **Role:** These servers manage the next layer of the domain hierarchy. For instance, the `.com` TLD server directs queries for domains ending in `.com`.
- **Example:** When resolving , the root server points to the `.com` TLD server.

Authoritative Name Servers

- **Role:** These servers contain the definitive information about a domain, including its IP address and DNS records (such as A, AAAA, MX, etc.).
- **Importance:** They provide the final answer in the DNS resolution process.

DNS Resolvers

- **Role:** Resolvers (or recursive resolvers) are the intermediaries that handle DNS queries from clients. They communicate with root, TLD, and authoritative servers to retrieve the required information.
- **Caching:** They cache responses to speed up subsequent queries.

---

4. Caching in DNS

Caching is a crucial aspect of DNS that improves performance by storing previously resolved queries for a predetermined period (Time-To-Live, or TTL). This means that if another user requests the same domain, the resolver can return the cached IP address quickly without going through the entire resolution process again.

Benefits of DNS Caching:

- **Reduced Latency:** Faster responses for frequently accessed domains.
- **Lower Load on Servers:** Fewer queries are sent upstream, reducing the load on root and TLD servers.
- **Improved Scalability:** Caching allows DNS infrastructure to handle a higher volume of queries efficiently.

5. Conclusion

The Domain Name System (DNS) is an essential component of the Internet, acting as a distributed phone book that maps human-friendly domain names to machine-friendly IP addresses. Its hierarchical, distributed architecture ensures scalability, high availability, and resilience, even in the face of high query volumes or network failures.

From root servers to TLDs, authoritative name servers, and resolvers, each component of DNS plays a crucial role in ensuring that users can quickly and reliably access websites and online services.

With features like caching and security extensions such as DNSSEC, the DNS ecosystem is designed to meet the demanding needs of the modern Internet.