

DNS (Domain Name System) - Complete Study Notes

What is DNS?

Domain Name System (DNS) is a system used to map domain names with IP addresses. It acts as a translator between human-readable domain names and machine-readable IP addresses.

Why Do We Use DNS? - Two Main Reasons

Reason 1: Human Memory and Ease of Use

The Problem:

- In real life, we open lots of websites and web applications daily
- We enter domain names like Google, Facebook, YouTube - not IP addresses
- Users always recall websites by domain names, not by IP addresses

Why Domain Names Are Easier:

- It's easier to remember pictures and names than numbers
- Our mind can easily capture names and pictures
- Capturing numbers is difficult for human memory

Real Example: Instead of telling friends to open an IP address like `123.456.789.012` (which is the IP address of cricinfo.com), we simply say "open cricinfo.com"

Phonebook Analogy:

- DNS works like a phonebook
- In a phonebook, we never search for someone by their phone number
- We remember a person's name and search alphabetically in the phonebook
- By the person's name, we find their phone number
- Similarly, by domain name, DNS finds the IP address

Reason 2: IP Addresses Are Dynamic (Not Static)

The Scenario: Let's say you create a website called `xyz.com`:

1. Initial Setup:

- You create a web application `xyz.com`
- Host it on Amazon Web Services (AWS)
- Purchase database and get hosting
- Hosting gives you an IP address

- Everyone searches for your website by the name `xyz.com`, not by IP address

2. When You Need to Change Hosting:

- After 1-2 years, you or Amazon decide to shift to a different IP address
- Maybe you find Amazon expensive and want to switch to another provider like GoDaddy
- When you shift to the new web application, your IP address changes
- **But your domain name `xyz.com` remains the same**
- Everyone still remembers you by your name `xyz.com`, not by your IP address

3. What Happens Behind the Scenes:

- IP addresses keep changing from the backend
- The domain name database keeps updating the IP mappings
- **The domain name should remain the same**
- If you change your website name from `xyz.com` to `abc.com`, you'd have to restart your business because everybody knows you by `xyz.com`

How DNS Resolution Works - The Process

The Setup

When you open your browser and type `cricinfo.com`, here's what happens:

Step 1: Initial Request

- You enter the domain name `cricinfo.com` in your browser
- The domain name needs to be mapped with an IP address
- This is not your responsibility - there's a hierarchical architecture to handle this

The DNS Hierarchical Architecture

Component 1: Resolver

- Acts as a middleman
- Its job is to resolve the domain name into an IP address

Component 2: Root Server

- **Important Fact:** There are exactly **13 root servers** all over the world
- Your request goes to the root server first
- Root server gives an IP address and location of the next level to the resolver

Component 3: Generic/Country Server Level

- After root server, the request moves to generic server level or country server level

- **Key Point:** Resolution happens from **right to left** (resolves from the backend)

For **cricinfo.com**:

- **.com** = Commercial (non-profitable organization)
- **.in** = India
- **.mil** = Military
- **.edu** = Education
- Different domains for different purposes

Component 4: Authoritative Servers

- These servers store all IP addresses and domain names
- They have the actual mapping: **cricinfo** → IP address
- **Important:** A domain can have multiple IP addresses

Example of Multiple IP Addresses:

- Large organizations like Google, YouTube, Facebook cannot work with a single server
- Facebook alone has approximately **70,000 servers**
- Since there are so many servers, there can be multiple IP addresses
- Authoritative server chooses one of the IP addresses for your request

The Complete DNS Resolution Flow

1. **User → Resolver:** "I want cricinfo.com"
2. **Resolver → Root Server:** "Where can I find .com domains?"
3. **Root Server → Resolver:** "Go to this commercial server"
4. **Resolver → Commercial Server:** "Where can I find cricinfo?"
5. **Commercial Server → Resolver:** "Go to this authoritative server"
6. **Resolver → Authoritative Server:** "What's the IP for cricinfo?"
7. **Authoritative Server → Resolver:** "Here's the IP address"
8. **Resolver → User's Browser:** "Here's the IP address for cricinfo.com"

Step 2: After Getting the IP Address

- Once you have the IP address, let's say cricinfo is hosted in a European country
- Your request is sent to Europe
- You can now check data, match scores, etc.
- The data comes back from the European server to your browser

Traffic Distribution and Load Balancing

Why Hierarchical Architecture?

- **Distribution of Traffic:** Instead of all traffic going to one place, it's distributed
- **Load Balancing:** .com servers are not single servers - they are multiple servers
- This makes load balancing easier and more efficient

DNS Caching - Performance Optimization

The Caching Concept

- **ISP (Internet Service Provider)** is responsible for caching
- ISP always keeps frequently accessed domain-IP mappings in cache memory

How Caching Works:

1. **First Time:** When you search for a website for the first time, the request goes through the complete DNS resolution process (root server → commercial server → authoritative server)
2. **Subsequent Times:** The ISP stores the domain-IP mapping in cache
3. **Result:** Next time your request doesn't take as much time and gets resolved quickly

Benefits of DNS Caching:

- Faster website loading
- Reduced load on DNS servers
- Better user experience

DNS Resolution Modes (Advanced Topics)

Note: The video mentions two modes that will be covered in detail in future videos:

1. **Recursive Mode**
2. **Iterative Mode**

These modes determine how the DNS resolver communicates with different DNS servers during the resolution process.

Key Takeaways

1. **DNS Purpose:** Maps human-readable domain names to machine-readable IP addresses
2. **Human-Friendly:** We remember names better than numbers
3. **Dynamic IP Support:** Domain names stay constant even when IP addresses change
4. **Hierarchical System:** Root servers → Generic/Country servers → Authoritative servers

5. **Caching:** ISPs cache frequently used mappings for faster resolution

6. **Load Distribution:** Multiple servers handle requests efficiently

7. **Global Infrastructure:** 13 root servers serve the entire world

Real-World Applications

- **Website Access:** Every time you visit a website
 - **Email Delivery:** Email servers use DNS to find mail servers
 - **CDN Services:** Content delivery networks use DNS for optimal server selection
 - **Load Balancing:** Large organizations use DNS for distributing traffic across multiple servers
-

Note: This covers the basic DNS concepts. Advanced topics like recursive vs iterative modes and detailed caching mechanisms will be covered in subsequent studies.