

+------------------------+

| User types a URL |

| (e.g., www.example.com)|

+------------+-----------+

|

v

+------------------------+

| Local DNS resolver |

| Cache check |

+------------+-----------+

|

+------------|-----------+

| Cache hit? | Cache miss|

+------------+-----------+

|

+------------|-----------+

|

v

+------------------------+

| Recursive DNS Server |

| (e.g., ISP's DNS) |

+------------+-----------+

|

+------------|-----------+

| Root DNS | |

| Servers | |

+------------+-----------+

|

+------------|-----------+

| TLD DNS | |

| Servers | |

+------------+-----------+

|

+------------|-----------+

| Authoritative DNS |

| Servers for the domain |

+------------+-----------+

|

v

+------------------------+

| Authoritative DNS |

| Servers provide IP |

| address for the domain |

+------------+-----------+

|

v

+------------------------+

| Recursive DNS Server |

| Receives IP address |

+------------+-----------+

|

v

+------------------------+

| Local DNS resolver |

| Provides IP address |

+------------+-----------+

|

v

+------------------------+

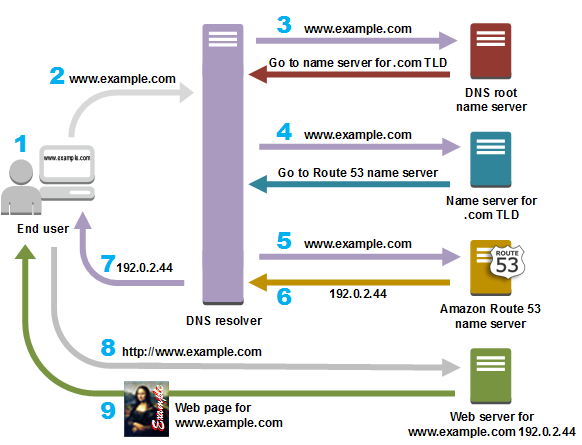
| User's device |

| Connects to the IP |

| address of the |

| requested domain |

|  |
| --- |
|  |



The Domain Name System (DNS) is a crucial component of the internet that translates human-readable domain names into IP addresses. This translation is necessary because computers communicate with each other using IP addresses, which are numerical labels assigned to each device connected to a network. Here's a step-by-step explanation of how DNS works:

User Input: A user enters a domain name (e.g., [www.example.com](http://www.example.com/)) into a web browser or another application.

1. Local DNS Resolver (Cache Check): The user's device checks its local DNS resolver cache to see if it already has the IP address associated with the requested domain. If the information is found in the cache and is still valid, the process skips the next steps and proceeds directly to connecting to the IP address.
2. Recursive DNS Server: If the local DNS resolver cache does not contain the required information, the user's device queries a recursive DNS server. This server is typically provided by the user's Internet Service Provider (ISP) or another designated DNS service.
3. Root DNS Servers: If the recursive DNS server doesn't have the necessary information, it contacts the root DNS servers. The root DNS servers are a crucial part of the DNS hierarchy and provide information about Top-Level Domains (TLDs).

TLD DNS Servers: The root DNS servers direct the recursive DNS server to the appropriate Top-Level Domain (TLD) server. TLDs are the last part of domain names, such as .com, .org, or .net.

1. Authoritative DNS Servers: The TLD server provides information about the authoritative DNS servers responsible for the specific domain. The authoritative servers have the most up-to-date information about the domain's IP address.

IP Address Resolution: The recursive DNS server queries the authoritative DNS servers for the actual IP address associated with the requested domain.

1. Response to Recursive DNS Server: The authoritative DNS servers respond to the recursive DNS server with the IP address.

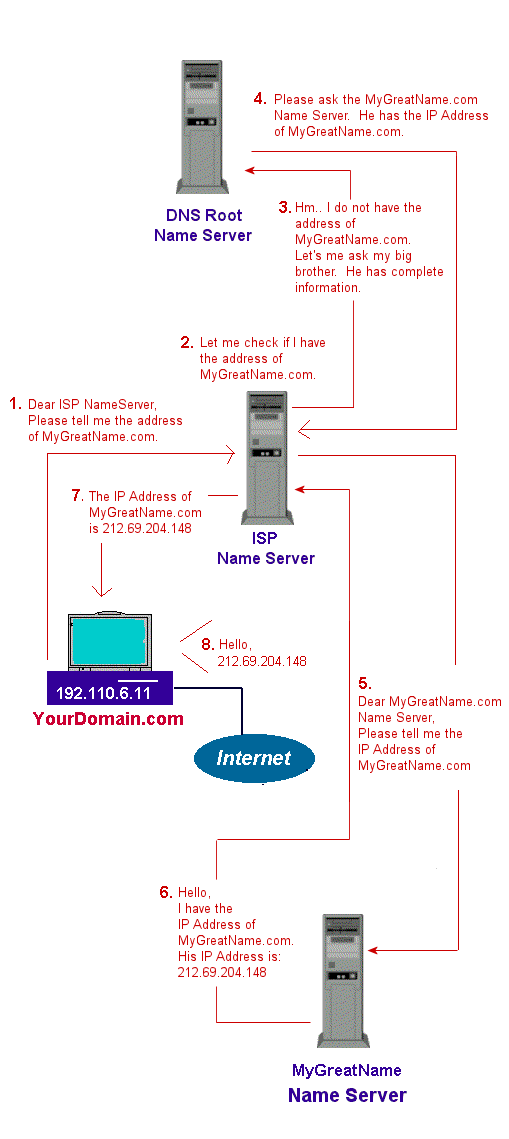
Cache Update: The recursive DNS server caches the IP address information for future use, reducing the need to repeat the lookup process for the same domain.

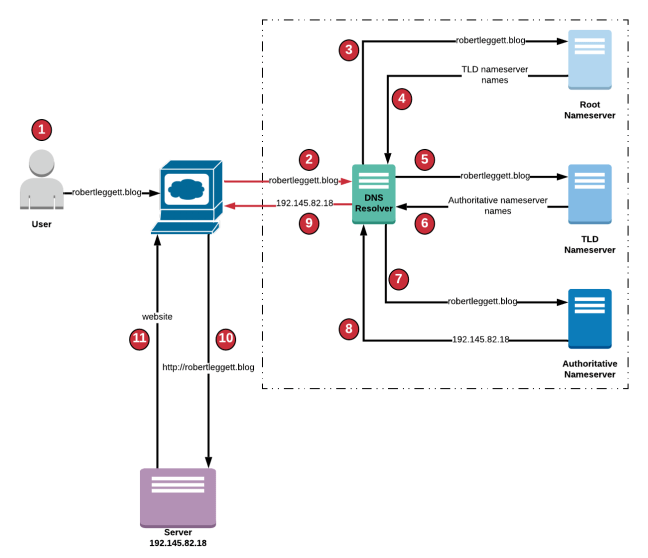
1. Response to Local DNS Resolver: The recursive DNS server provides the IP address to the user's device through the local DNS resolver.
2. User Access: Now armed with the IP address, the user's device can connect to the web server hosting the requested content.

Throughout this process, various caching mechanisms help improve efficiency by reducing the need to repeatedly traverse the entire DNS hierarchy for frequently accessed domains. The distributed nature of DNS and its hierarchical structure contribute to the scalability and reliability of the system.

Top of Form

w**orking system of DNS:**





DNS Sequence Flow

1. The user enters robertleggett.blog in the address bar of the browser.
2. The request for robertleggett.blog is forwarded to a DNS resolver.
3. The DNS resolver forwards the request for robertleggett.blog to a Root Nameserver.
4. The Root Nameserver for . domain responds to the request with the names of the TLD Nameservers.
5. The DNS resolver then forwards the request for robertleggett.blog to an TLD Nameserver that is associated to .blog domain.
6. The TLD Nameserver for .blog domain responds to the request with the names of the Authoritative Nameservers that are associated with the robertleggett.blog.
7. The DNS resolver selects an Authoritative Nameserver and forwards the request for robertleggett.blog.
8. The Authoritative Nameserver looks in the robertleggett.blog hosted zone for the record, it then gets the associated IP address for a server, 192.145.82.18, and returns the IP address to the DNS resolver.
9. The DNS resolver now has the IP address that the browser needs. The DNS resolver will cache the IP address for robertleggett.blog for an amount of time that is specified for a quicker retrieval the next time robertleggett.blog is requested.
10. The browser sends a request for robertleggett.blog to the IP address that it got from the DNS resolver.
11. The server or other resource at 192.145.82.18 returns the website for robertleggett.blog to the browser which displays the page.