Internet Domain Name System

When **DNS** was not into existence, one had to download a **Host file**containing host names and their corresponding IP address. But with increase in number of hosts of internet, the size of host file also increased. This resulted in increased traffic on downloading this file. To solve this problem the DNS system was introduced.

**Domain Name System** helps to resolve the host name to an address. It uses a hierarchical naming scheme and distributed database of IP addresses and associated names

## IP Address

IP address is a unique logical address assigned to a machine over the network. An IP address exhibits the following properties:

* IP address is the unique address assigned to each host present on Internet.
* IP address is 32 bits (4 bytes) long.
* IP address consists of two components:**network component** and **host component**.
* Each of the 4 bytes is represented by a number from 0 to 255, separated with dots. For example 137.170.4.124

IP address is 32-bit number while on the other hand domain names are easy to remember names. For example, when we enter an email address we always enter a symbolic string such as webmaster@tutorialspoint.com.

## Uniform Resource Locator (URL)

**Uniform Resource Locator (URL)** refers to a web address which uniquely identifies a document over the internet.

This document can be a web page, image, audio, video or anything else present on the web.

For example, **www.tutorialspoint.com/internet\_technology/index.html**is an URL to the index.html which is stored on tutorialspoint web server under internet\_technology directory.

### URL Types

There are two forms of URL as listed below:

1. Absolute URL
2. Relative URL

#### ABSOLUTE URL

Absolute URL is a complete address of a resource on the web. This completed address comprises of protocol used, server name, path name and file name.

For example http:// www.tutorialspoint.com / internet\_technology /index.htm. where:

* **http** is the protocol.
* **tutorialspoint.com** is the server name.
* **index.htm** is the file name.

The protocol part tells the web browser how to handle the file. Similarly we have some other protocols also that can be used to create URL are:

* FTP
* https
* Gopher
* mailto
* news

#### RELATIVE URL

Relative URL is a partial address of a webpage. Unlike absolute URL, the protocol and server part are omitted from relative URL.

Relative URLs are used for internal links i.e. to create links to file that are part of same website as the WebPages on which you are placing the link.

For example, to link an image on tutorialspoint.com/internet\_technology/internet\_referemce\_models, we can use the relative URL which can take the form like **/internet\_technologies/internet-osi\_model.jpg.**

### Difference between Absolute and Relative URL

|  |  |
| --- | --- |
| **Absolute URL** | **Relative URL** |
| Used to link web pages on different websites | Used to link web pages within the same website. |
| Difficult to manage. | Easy to Manage |
| Changes when the server name or directory name changes | Remains same even of we change the server name or directory name. |
| Take time to access | Comparatively faster to access. |

## Domain Name System Architecture

The Domain name system comprises of **Domain Names, Domain Name Space, Name Server** that have been described below:

### Domain Names

Domain Name is a symbolic string associated with an IP address. There are several domain names available; some of them are generic such as **com, edu, gov, net** etc, while some country level domain names such as **au, in, za, us**etc.

The following table shows the **Generic** Top-Level Domain names:

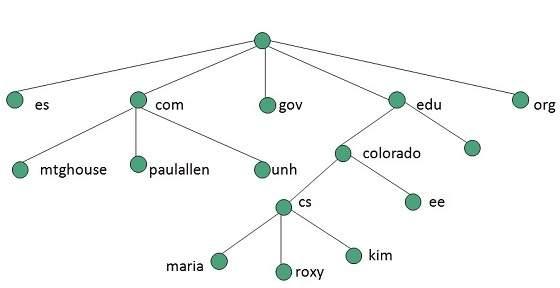
|  |
| --- |
|  |
| **Domain Name** | **Meaning** |
| Com | Commercial business |
| Edu | Education |
| Gov | U.S. government agency |
| Int | International entity |
| Mil | U.S. military |
| Net | Networking organization |
| Org | Non profit organization |

The following table shows the **Country top-level** domain names:

|  |
| --- |
|  |
| **Domain Name** | **Meaning** |
| au | Australia |
| in | India |
| cl | Chile |
| fr | France |
| us | United States |
| za | South Africa |
| uk | United Kingdom |
| jp | Japan |
| es | Spain |
| de | Germany |
| ca | Canada |
| ee | Estonia |
| hk | Hong Kong |

### Domain Name Space

The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. The following diagram shows the domain name space hierarchy:



In the above diagram each subtree represents a domain. Each domain can be partitioned into sub domains and these can be further partitioned and so on.

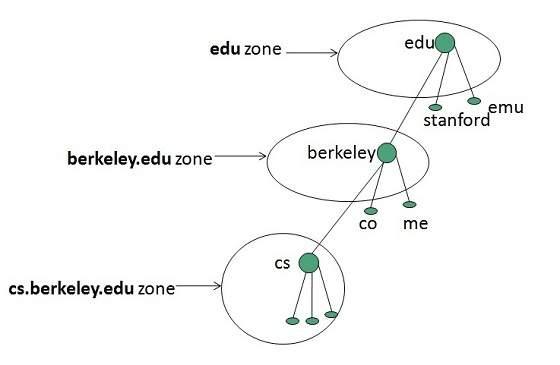
### Name Server

Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.

* Hierarchy of server is same as hierarchy of names.
* The entire name space is divided into the zones

### Zones

Zone is collection of nodes (sub domains) under the main domain. The server maintains a database called zone file for every zone.



If the domain is not further divided into sub domains then domain and zone refers to the same thing.

The information about the nodes in the sub domain is stored in the servers at the lower levels however; the original server keeps reference to these lower levels of servers.

#### TYPES OF NAME SERVERS

Following are the three categories of Name Servers that manages the entire Domain Name System:

1. Root Server
2. Primary Server
3. Secondary Server

##### ROOT SERVER

Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server

##### PRIMARY SERVERS

Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.

##### SECONDARY SERVER

Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.

## DNS Working

DNS translates the domain name into IP address automatically. Following steps will take you through the steps included in domain resolution process:

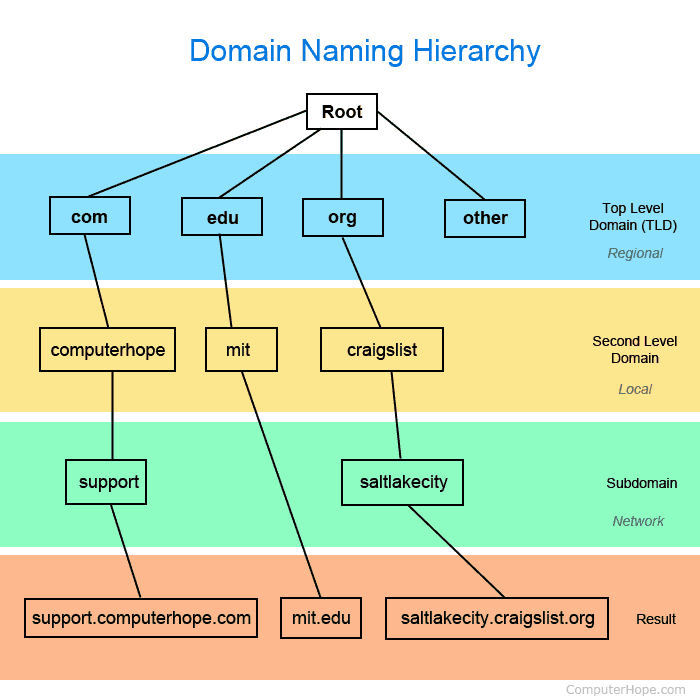
* When we type **www.tutorialspoint.com** into the browser, it asks the local DNS Server for its IP address.

Here the local DNS is at ISP end.

* When the local DNS does not find the IP address of requested domain name, it forwards the request to the root DNS server and again enquires about IP address of it.
* The root DNS server replies with delegation that **I do not know the IP address of www.tutorialspoint.com but know the IP address of DNS Server.**
* The local DNS server then asks the com DNS Server the same question.
* The **com** DNS Server replies the same that it does not know the IP address of www.tutorialspont.com but knows the address of tutorialspoint.com.
* Then the local DNS asks the tutorialspoint.com DNS server the same question.
* Then tutorialspoint.com DNS server replies with IP address of www.tutorialspoint.com.
* Now, the local DNS sends the IP address of www.tutorialspoint.com to the computer that sends the request.

# Domain namespace

Alternatively referred to as a **namespace**, a **domain namespace** is a name service provided by the [Internet](https://www.computerhope.com/jargon/i/internet.htm) for Transmission Control Protocol networks/Internet Protocol ([TCP/IP](https://www.computerhope.com/jargon/t/tcpip.htm)). DNS is broken up into domains, a logical organization of computers that exist in a larger network. Below is an example of the hierarchy of domain naming on the Internet.



In the above example, all websites are broken into regional sections based on the [top level domain](https://www.computerhope.com/jargon/t/tld.htm) (TLD). In the example of http://support.computerhope.com it has a ".com" TLD, with "computerhope" as its second level domain that is local to the .com TLD, and "support" as its [subdomain](https://www.computerhope.com/jargon/s/subdomai.htm), which is determined by its server.

### Domain name-address resolution

Mapping a domain name to a physical IP address or an IP address to a domain name is called *name-address resolution.*  
  
**Resolver**  
DNS is designed as a client-server application. A host needs to map an address to a name or a name to an address calls a DNS client called *resolver*. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.  
After the resolver recieves the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.  
  
**Mapping Domain names to Addresses**  
Most of the time, the resolver gives a domain name to the server and asks for the corresponding address. In this case, the server checks the generic domains or the country domains to find the mapping.  
  
If  the domain name is from the genric domains section, the resolver receives a domain name such as *"chal.atc.fhda.edu.".*The query is sent by the resolver to local DNS server for resolution. If the local server cannot resolve the query, it either refers the resolver to other servers or asks other servers directly.  
  
If the domain name is from the country domains section, the resolver receives a domain name such as *"ch.fhda.cu.ca.us.".*The procedure is the same.  
  
**Mapping Addresses to Names**  
A client can send an IP address to a server to be mappedto a domain name. As mentioned before, this is called a PTR query. To answer queries of this kind, DNS uses the inverse domain. However, in the request, the IP address should be reversed and two labels, *in-addr*and *arpa,*should be appended to create a domain acceptable by inverse domain section. For example, If the resolver receives the Ip address 132.34.45.121, the resolver first inverts the address and then adds the two labels before sending. The domain name sent is *"121.45.34.132.in-addr.arpa." ,*which is received by local DNS and resolved.  
  
**Recursive Resolution**  
The client (resolver) an ask for a recursive answer from a name server. This means that the resolver expects the server to supply the final answer. If the server is the authority for the domain name, it checks its database and responds. If the server is not the authority, it sends the request to yet another server. When the query is finally resolved, the response levels back until itfinally reaches the requesting client.  
  
**Iterative Resolution**  
 If the client does not ask for a recursiveanswer, the mapping can be done iteratively. If the server is an authority for the name, it sends the answer. If it is not, it returns(to the client) the IP address of the server that it thinks can resolve the query. The client is responsible for repeating the query to this second server. If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwiese, it returns the IP address of a new server to the client. Now the client must repeat the query to the third server. thi sprocess is called *iterative*  because theclient repeats the same query to multiple servers.  
  
**Domain Name Caching (Cached Version of Page)**  
Each time a server recieves a query from a name that is not in its domain, it needs to search its database for a server IP address. Reduction of this search time would increase efficiency. DNS handles this with a mechanism called *caching.*  When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client. If the same or another client asks for the same mapping, it can check its cache memory and resolve the problem. However,to inform the client that response is coming from the cache memory and not from an authitative source, the server marks the response as *unauthoritative.*  
  
Caching speeds up resolution, but it can also be problematic. If a server cache a mapping for a long time, it may send an outdated mapping to the client. To counter this, two techniques are used. First, the authoritative server always adds a piece of information to the mapping called *time-to-live*(**TTL**). It defines the time in seconds that the reciever server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server. Second, DNS requires that each server keeps a TTL counter for each mapping it caches. The cache memory must be searched periodically and those mappings with an expired TTL must be purged.  
  
That was the discussion on Name-address resolution in DNS, Iterative and recursive resolution of domain addresses. We also learned about Caching and Domain name mappings, & TTL. Hope you have enjoyed reading the article.