Assignment Group 10

Problem Definition:

Write a program for DNS lookup. Given an IP address input, it should return URL and vice versa. (Use JAVA/PYTHON)

1. Prerequisite:

1. Application Layer: Roles, Protocols
2. Java Programming Syntax

2. Learning Objectives:

- Students will able to understand working of DNS Protocol

3. Theory

DNS

Domain Name System (DNS) is the default name resolution service used in a Microsoft Windows Server 2003 network. DNS is part of the Windows Server 2003 TCP/IP protocol suite and all TCP/IP network connections are, by default, configured with the IP address of at least one DNS server in order to perform name resolution on the network.

DNS Architecture

DNS architecture is a hierarchical distributed database and an associated set of protocols that define:

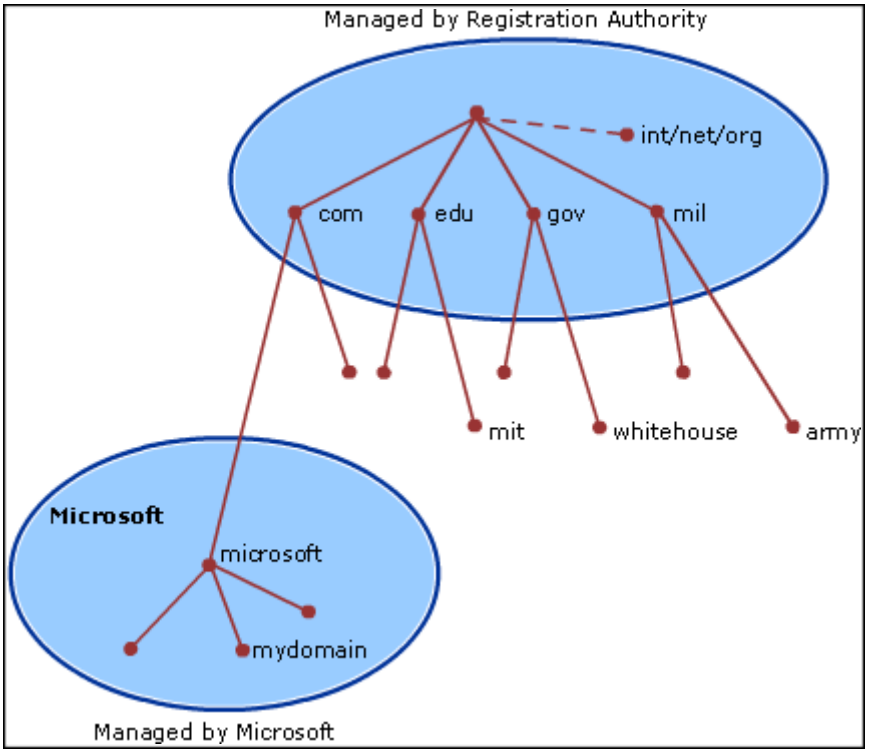
- A mechanism for querying and updating the database.
- A mechanism for replicating the information in the database among servers.
- A schema of the database.

DNS Domain Names

The Domain Name System is implemented as a hierarchical and distributed database containing various types of data, including host names and domain names. The names in a DNS database form a hierarchical tree structure called the domain namespace. Domain names consist of individual labels separated by dots, for example: mydomain.microsoft.com.

A Fully Qualified Domain Name (FQDN) uniquely identifies the hosts position within the DNS hierarchical tree by specifying a list of names separated by dots in the path from the referenced host to the root. The next figure shows an example of a DNS tree with a host called mydomain within the microsoft.com. domain. The FQDN for the host would be mydomain.microsoft.com.

DNS Domain Name Hierarchy



Types of DNS Domain Names

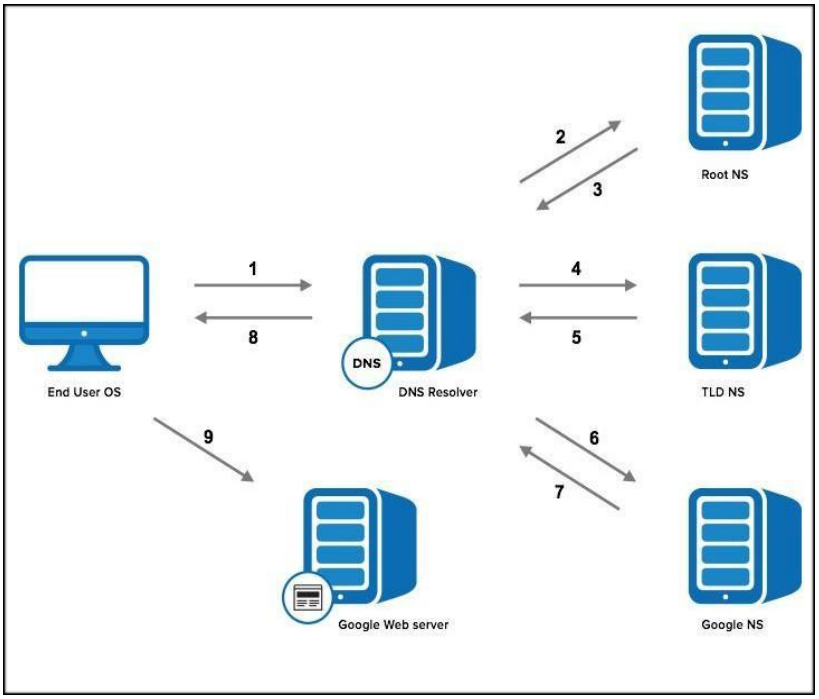
Name Type	Description	Example
Root domain	This is the top of the tree, representing an unnamed level; it is sometimes shown as two empty quotation marks (""), indicating a null value. When used in a DNS domain name, it is stated by a trailing period (.) to designate that the name is located at the root or highest level of the domain hierarchy. In this instance, the DNS domain name is considered to be complete and points to an exact location in the tree of names.	A single period (.) or a period used at the end of a name, such as "example.microsoft.com."

	Names stated this way are called fully qualified domain names (FQDNs).	
Top level domain	A name used to indicate a country/region or the type of organization using a name.	".com", which indicates a name registered to a business for commercial use on the Internet.
Second level domain	Variable-length names registered to an individual or organization for use on the Internet. These names are always based upon an appropriate top-level domain, depending on the type of organization or geographic location where a name is used.	"microsoft.com.", which is the second-level domain name registered to Microsoft by the Internet DNS domain name registrar.
Subdomain	Additional names that an organization can create that are derived from the registered second-level domain name. These include names added to grow the DNS tree of names in an organization and divide it into departments or geographic locations.	"example.microsoft.com.", which is a fictitious subdomain assigned by Microsoft for use in documentation example names.
Host or resource name	Names that represent a leaf in the DNS tree of names and identify a specific resource. Typically, the leftmost label of a DNS domain name identifies a specific computer on the network. For example, if a name at this level is used in a host (A) RR, it is used to look up the IP address of computer based on its host name.	"host-a.example.microsoft.com.", where the first label ("host-a") is the DNS host name for a specific computer on the network.

Working of DNS Lookup

DNS is what translates your familiar domain name (www.google.com) into an IP address your browser can use (173.194.33.174).

Before the page and any resource on the page is loaded, the DNS must be resolved so the browser can establish a TCP connection to make the HTTP request. In addition, for every external resource referenced by a URL, the DNS resolution must complete the same steps (per unique domain) before the request is made over HTTP. The DNS Resolution process starts when the user types a URL address on the browser and hits Enter. At this point, the browser asks the operating system for a specific page, in this case google.com.



Step 1: OS Recursive Query to DNS Resolver

Since the operating system doesn't know where "www.google.com" is, it queries a DNS resolver. The query the OS sends to the DNS Resolver has a special flag that tells it is a "recursive query." This means that the resolver must complete the recursion and the response must be either an IP address or an error.

Step 2: DNS Resolver Iterative Query to the Root Server

The resolver starts by querying one of the root DNS servers for the IP of "www.google.com." This query does not have the recursive flag and therefore is an "iterative query," meaning its response must be an address, the location of an authoritative name server, or an error. The root is represented in the hidden trailing "." at the end of the domain name. Typing this extra "." is not necessary as your browser automatically adds it.

Step 3: Root Server Response

These root servers hold the locations of all of the top level domains (TLDs) such as .com, .de, .io, and newer generic TLDs such as .camera.

The root doesn't have the IP info for "www.google.com," but it knows that .com might know, so it returns the location of the .com servers. The root responds with a list of the 13 locations of the .com gTLD servers, listed as NS or "name server" records.

Step 4: DNS Resolver Iterative Query to the TLD Server

Next the resolver queries one of the .com name servers for the location of google.com. Like the Root Servers, each of the TLDs have 4-13 clustered name servers existing in many locations. There are two types of TLDs: country codes (ccTLDs) run by government

organizations, and generic (gTLDs). Every gTLD has a different commercial entity responsible for running these servers. In this case, we will be using the gTLD servers controlled by Verisign, who run the .com, .net, .edu, and .gov among gTLDs.

Step 5: TLD Server Response

Each TLD server holds a list of all of the authoritative name servers for each domain in the TLD. For example, each of the 13 .com gTLD servers has a list with all of the name servers for every single .com domain. The .com gTLD server does not have the IP addresses for google.com, but it knows the location of google.com's name servers. The .com gTLD server responds with a list of all of google.com's NS records. In this case Google has four name servers, "ns1.google.com" to "ns4.google.com."

Step 6: DNS Resolver Iterative Query to the Google.com NS

Finally, the DNS resolver queries one of Google's name server for the IP of "www.google.com."

Step 7: Google.com NS Response

This time the queried Name Server knows the IPs and responds with an A or AAAA address record (depending on the query type) for IPv4 and IPv6, respectively.

Step 8: DNS Resolver Response to OS

At this point the resolver has finished the recursion process and is able to respond to the end user's operating system with an IP address.

Step 9: Browser Starts TCP Handshake

At this point the operating system, now in possession of www.google.com's IP address, provides the IP to the Application (browser), which initiates the TCP connection to start loading the page.

Conclusion:

Hence we have studied working of DNS protocol.