Modern Education Society's

Wadia **College of Engineering, Pune**

NAME OF STUDENT:	**CLASS:**
SEMESTER/YEAR:	**ROLL NO:**
DATE OF PERFORMANCE:	**DATE OF SUBMISSION:**
EXAMINED BY:	**EXPERIMENT NO:**

Assignment No. 10(Group - C)

Title: DNS Lookup for specified for domain name or IP address.

Objectives:

Understand working of DNS lookup

Problem Statement:

Write a program for DNS lookup. Given an IP address as input, it should return URL and viceversa.

Outcomes:

Understand working of DNS lookup

Tools Required:

Software: jdk compiler.

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

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.

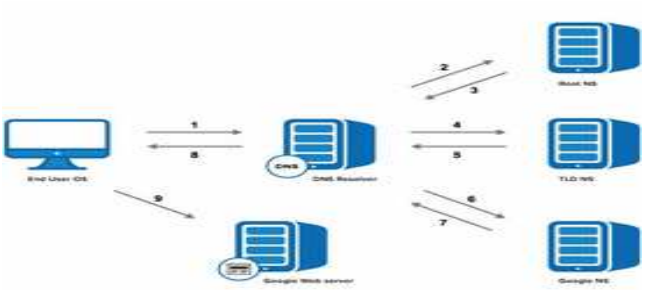
DNS Domain Names

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.

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 :

Thus we have configured RIP/OSPF/BGP using packet tracer using wireshark.

Questions:-

1. What is DNS ? What is the main purpose of DNS server?

2. What are DNS Zones?

3. What is round robin DNS?