ACADEMIC YEAR: 2022-23

DEPARTMENT of COMPUTER ENGINEERING DEPARTMENT

CLASS: T.E. **SEMESTER: I**

SUBJECT: CNSL

ASSINGMENT NO.	C–1
TITLE	DNS lookup
PROBLEM STATEMENT /DEFINITION	Write a program for DNS lookup. Given an IP address as input, it should return URL & vice–versa
OBJECTIVE	DNS Objective: I. mechanism for administrators to assign host names to their own systems without creating duplicate names & to make that host name information globally available to other administrators without relying in single access point II. mechanism has to support information about systems that use various protocols with different types of addresses
OUTCOME	o To understand that Domain Name Servers are essentially the 'address book' of the Internet & store information to help Internet systems route requests & replies. o To explain how DNS hierarchy supports scaling on the Internet
S/W PACKAGES AND HARDWARE APPARATUS USED	IDE, compiler, Internet PC with the configuration as Pentium IV 1.7 GHz. 128M.B RAM, 40 GB HDD, 15''Color Monitor, Keyboard, Mouse. Operating Systems (64-Bit) 64-BIT Fedora 20 or latest 64-BIT Update of Equivalent Open source OS Programming Tools (64-Bit) GCC/G++
REFERENCES	'Data Communications & Networking', 5th edition, B. Forouzan, Mc Graw Hill
STEPS	Understand the concept through simulation Write algorithm & flow chart Implementation Debugging Testing Troubleshooting
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Algorithm 8. Test cases 10. Conclusion/Analysis

Prerequisites:

Concepts related Theory:

DNS = Internet’s Directory Service

Two ways to identify host:

i. by host name (people prefer mnemonic hostname identifier)
ii. by IP address (routers prefer fixed length , hierarchically structured IP address)

The main task of Internet’s domain name system is providing a directory service (DNS : name → IP address)

DNS is

a) distributed database implemented in a hierarchy of DNS servers
b) an application layer protocol that allows hosts to query the distributed database

Host table = simple ASCII file that contains a list of network system addresses & their equivalent host names

Ex: 127.0.0.1 local host

Today DNS has replaced host table universally

DNS Objective:

I. mechanism for administrators to assign host names to their own systems without creating duplicate names & to make that host name information globally available to other administrators without relying in single access point
II. mechanism has to support information about systems that use various protocols with different types of addresses

<u>DNS designed by Mockapetris consists of 3 basic elements:</u>

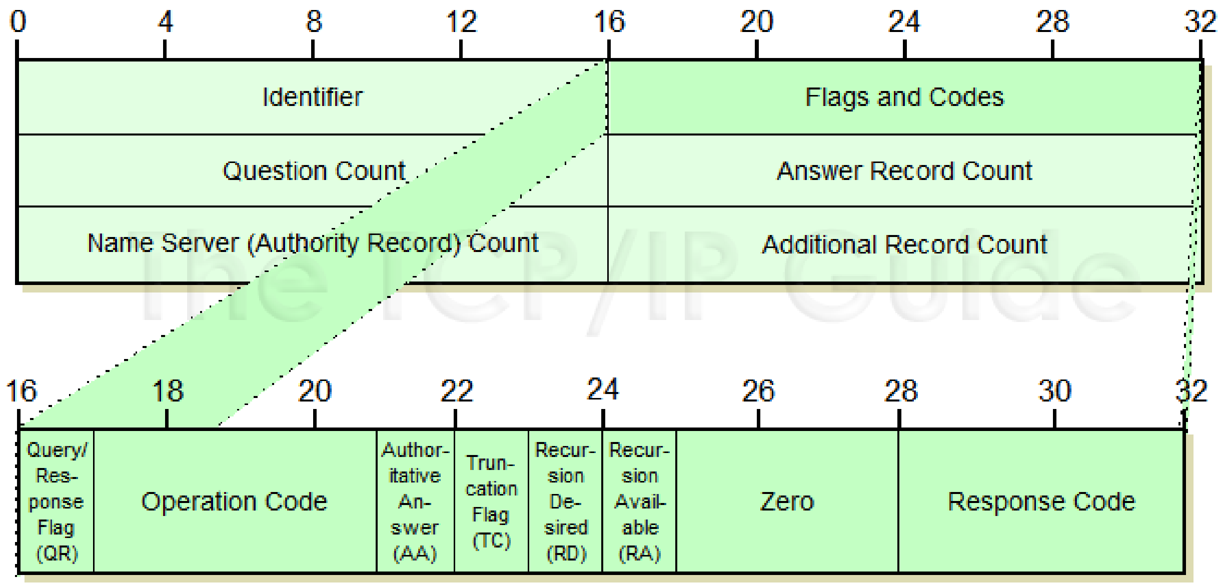
i. hierarchical name space that divides host system database into discrete elements called domains
ii. Domain name servers that contain information about the host & subdomains within a given domain
iii. Resolvers that generate requests for information from domain name servers (software that perform the translation is name resolver / resolver software)

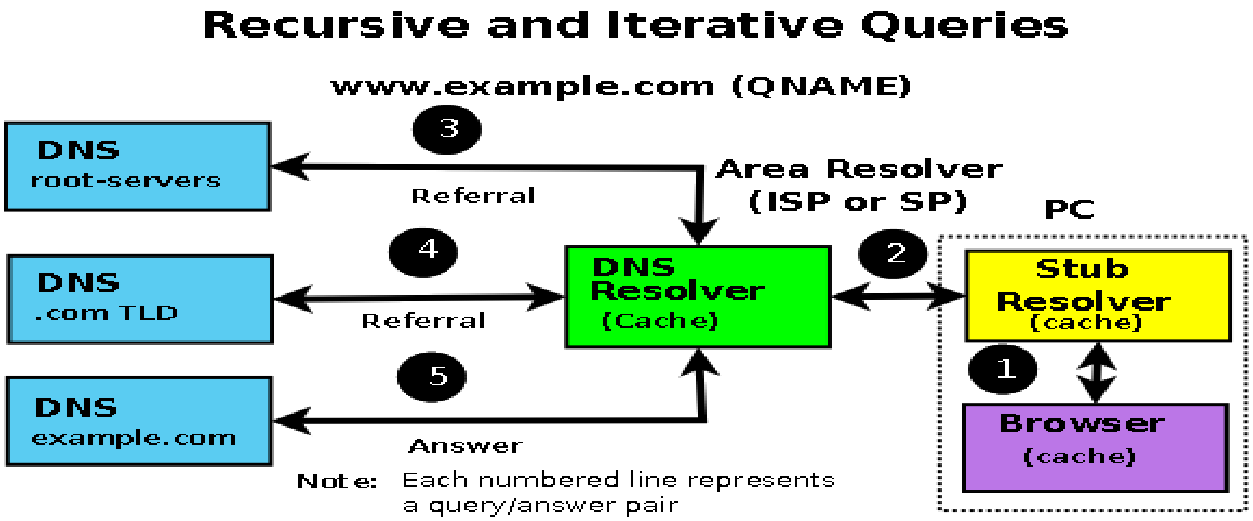
Client can choose either UDP or TCP when communicating with DNS server , most resolver use UDP because it requires less overhead. Well known port nos. for DNS name servers are UDP port 53 & TCP port 53 . It supports both.

- com = commercial organization
- edu = 4 years degree Educational Institute
- gov = Government organization
- mil = Military group
- net = Major network support center
- org = non – commercial / other organization
- arpa = Temporary ARPA domain
- int = International organization
- country code = A Country

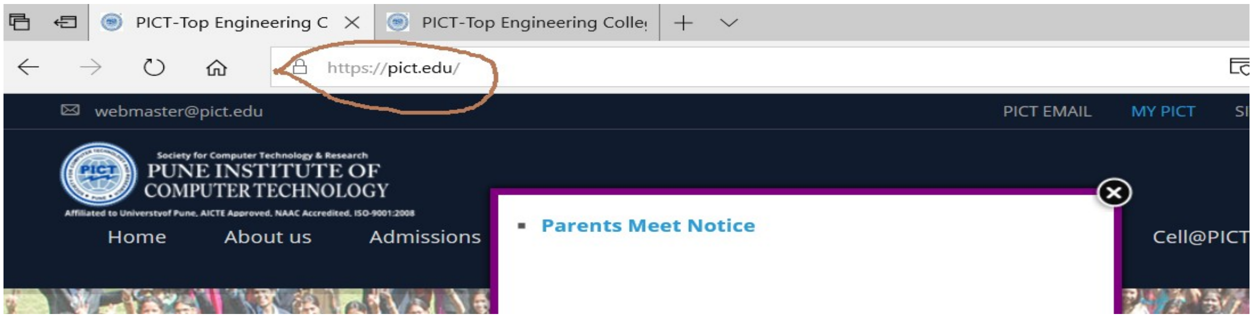
In Iterative resolution, if a client sends a request to a name server that does not have information the client needs, the server returns a pointer to a different name server & then client sends a new request to server. The first concatenated DNS server refers the client to some other server in the hierarchy ; servers use iterative query (referral)

In Recursive resolution, if a client sends a request to a server that doesn’t have the requested information, that server takes on the responsibility for sending requests to other servers to find necessary records then returns them to client . The first concatenated DNS server performs the lookup on behalf of client ; resolvers (clients) use recursive query





Item (2) is a Recursive Query - one question gives one complete answer
Items (3), (4) and (5) are Iterative queries which may return either a Referral or an answer


PICT-Top Engineering C
PICT-Top Engineering Colle
https://pict.edu/
webmaster@pict.edu
PICT EMAIL
MY PICT
Society for Computer Technology & Research
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
Affiliated to Universtyof Pune, AICTE Approved, NAAC Accredited, ISO-9001:2008
Home
About us
Admissions
Parents Meet Notice
Cell@PICT

Algorithm:

Server:

1) A TCP socket is created.
2) An Internet socket address structure is filled in with the wildcard address (INADDR_ANY) and the server‟s well-known port (PORT).
3) The socket is converted into a listening socket by calling the listen function.
4) The server blocks in the call to accept, waiting for the client connection to complete.
5) When the connection is established, the server reads the domain name from the client using readn.
6) It then finds out the corresponding address using gethostbyname() and sends it back to the client using writen.
7) Finally, the server closes the connected socket.

Client:

1) A TCP socket is created.
2) An Internet socket address structure is filled in with the server‟s IP address and the same port number.
3) The connect function establishes the connection with the server.
4) The client reads the domain name from the standard input using fgets, writes it to the server using writen.
5) It then reads back the server reply (IP address) using readline and outputs the same to the standard output using fputs.

<u>Nslookup:</u>

The nslookup command can be used in Windows and Unix to find various details relating to the Domain Name System (DNS) like IP addresses of a particular computer. nslookup comes with a number of subcommands to help us to get more information from the specific dns servers.

Example:
server NAME (where NAME is the name or ip address of the dns server we wish to query). It is not always possible to query a specific dns server as often dns queries are blocked to prevent denial of service attacks.

Using subcommands:

```
Z:\>nslookup 204.228.150.3
Server: eec.ac.in
Address: 10.1.1.2
Name: www.computerhope.com
```

```
Address: 204.228.150.3
```

The first two lines are information about the server delivering the answer to the nslookup requested by the user. The next two lines tell the user the name and IP address of the machine being looked up.

OUTPUT:

```
SERVER
$ cc DNSServer.c
$ ./a.out
SERVER...
Server Started
www.google.com
HOST : www.google.com --> IP ADDRESS --> 74.125.71.105
www.eec.ac.in
HOST : www.eec.ac.in --> IP ADDRESS --> 173.193.3.233
CLIENT
$ cc DNSClient.c
$ ./a.out 127.0.0.1
Enter the domain namewww.google.com
The host address is 74.125.71.105
$ ./a.out 127.0.0.1
Enter the domain namewww.eec.ac.in
The host address is 173.193.3.233
$
```

Conclusion:

Review Questions:

Qu.1) Why not centralize DNS ?
Qu. 2) Difference between Zone & Domain
Qu. 3) What is the purpose of identification field in DNS message format ?