CS-342 Assignment 2

Group 4: Aditya Patidar, 200101009

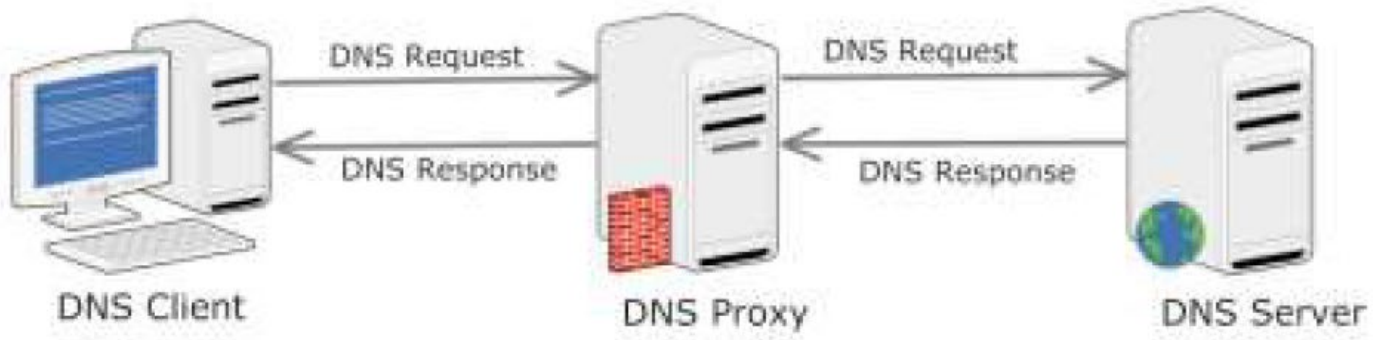
Advaita Mallik, 200101010

Akshat Mittal, 200101011

Aman Soni, 200101012

Application #4: Multi-stage DNS Resolving System using Client-Server socket programming

Implemented a 2 stage DNS resolver system as shown below:



Request Message Format:

Request_Type	Message

- Type 1: Message field contains Domain Name and requests for corresponding IP address.
- Type 2: Message field contains IP address and requests for the corresponding Domain Name.

Response Message Format:

Response_Type	Message

- Type 3: Message field contains Domain Name/IP address.
- Type 4: Message field contains error message “Entry not found in the database”.

Commands To Create Executables:

- gcc server.c -o s
- gcc -pthread proxy.c -o p
- gcc client.c -o c

Note: **-pthread** is used for POSIX semaphores and shared memory for concurrency.

Commands To Run server, proxy and client:

1. ./s <DNS_Server_Port>
2. ./p <Proxy_Server_Port> <DNS_Server_IP> <DNS_Server_Port>
3. ./c <Proxy_Server_IP> <Proxy_Server_Port>

These commands must be run in this order only since client requires proxy server running to connect.

For example:

1. ./s 8124
2. ./p 8075 127.0.0.1 8124
3. ./c 127.0.0.1 8075

Notes And Assumptions:

- Each message transferred has a maximum size of 256 bytes.
- The first byte of every message transferred is the request/response type.
- All Hostname-IP mappings must be stored in “database.txt”.
- After responding to a request by client, the connection between client and proxy server is closed.
- To send another request, run the client again.