Application ID – 4 : Multi-Stage DNS Resolving System

CS 558: Computer Systems Lab

Assignment #3 on Socket Programming

Group 16: 1) Samiksha Agrawal(204101069)

2) Kunal Wanikar(204101070)

3) Kaushal Dewangan(204101071)

4) Bidyanath Jha(204101072)

/----------

Assignment Details

1. Request message is of two types.(Type1 & Type 2)

a. Type 1 Message field specifies the Domain Name and requests for the IP address corresponding to the Domain name.

b. Type 2 Message field specifies IP address and requests for the Domain name corresponding to the IP address.

2. Response can be of two types again (Type3 & Type4)

a. Type 3 Message field contains Domain Name/IP address.

b. Type 4: Message field contains error message “entry not found in the database”.

3. Proxy server caches DNS Server response.

4. FIFO scheme is used for updating the cache by the proxy server .

/----------

How to run?

In this Application,Three programs has been implemented, namely, client.c, proxy.c and dns.c

Assume all the three- programs runs on the local host, i.e. 127.0.0.1

In order to compile the 3 programs, open 3 seperate terminal windows and run the below commands in each window. This will create 3-

Executable files namely, dns, proxy, and client.

```
$ gcc -o dns dns.c

$ gcc -o proxy proxy.c

$ gcc -o client client.c
```

The dns.c requires DNS server port number as an input, running proxy.c requires the Proxy server's port as the input while running DNS client, client.c, we are required to input IP address and the port of the proxy server.

Let's say the DNS Server is running on the port 8080 while the proxy server is serving its service at the port number 9999.

To serve these services at the port specified, we need to run the following 3-commands, in the respective windows:

```
$ ./dns 8080

$ ./proxy 9999

$ ./client 127.0.0.1 9999
```

This command will binds the port number 8080 to the dns server, and 9999 to the proxy server, and we have to explicitely tell the client program, the IP address(127.0.0.1) and the port number(9999) of the proxy server.

/----------

EXAMPLES:

Example 1 to run the code on the client side:

After running ./client 127.0.0.1 9999, the client will ask for type of message and hostname/IP. Suppose we need to find the IP address of kunal.com from then follow the below steps.

step1) Open terminal and run gcc -o dns dns.c

step2) ./dns 8080

step3) Open new terminal and run gcc -o proxy proxy.c

step4) ./proxy 9999

step5) Open new terminal and run gcc -o client client.c

step6)./client 127.0.0.1 9999

This will output the following

127.0.0.1 9999

Socket Creation Successful

Establishing connection with the server Successful

Requesting input: request and type of request

step7) type1 kunal.com

Request Message to server: 0kunal.com

Response from server: 1.2.3.4

The requested data: 1.2.3.4

Example 2) Suppose we need to find the Hostname of 1.2.3.4 from then follow the below steps.

step1) Open terminal and run gcc -o dns dns.c

step2) ./dns 8080

step3) Open new terminal and run gcc -o proxy proxy.c

step4) ./proxy 9999

step5) Open new terminal and run gcc -o client client.c

step6)./client 127.0.0.1 9999

This will output the following:

127.0.0.1 9999

Socket Creation Successful

Establishing connection with the server Successful

Requesting input: request and type of request

step7) type2 1.2.3.4

Request Message to server: 1.2.3.4

Response from server: kunal.com

The requested data: kunal.com

If the requested hostname/IP is not in the cache then connection of proxy server to dns server is made by entering the line "127.0.0.1 8080" in the terminal where proxy server is currently running.

Working:

1) Run the dns server by using the command above (gcc -o dns dns.c for compilation & ./dns 8080 for running the program) This will set the port number of DNS Server to 8080 and dns server will be connected successfully.

2) Run the proxy server by using the above command (gcc -o proxy proxy.c for compilation & ./proxy 9999 for running the program) This will set the port number of proxy server to 9999 and proxy server will be connected successfully.

3) Run the client by using the above command (gcc -o client client.c for compilation & ./client 127.0.0.1 9999 for running the program) This will make the connection between the client and the proxy server.

4) After running the above program, client asks for type of message and the hostname/IP depending on the type of message. The user must input type1/type2 and after a space write the specified IPaddress/Hostname.

5) The program will then search for the request in the proxy cache, if fount then it will respond according to the type if message specified otherwise it will search in the DNS server database.

/----------

Working in the backend:

Now, all the server initial setup is over.The client will request for either the IP address or for the Domain Name.In order to request for IP address a request message of type-1 is used, while for requesting for the Domain name, a request message of type of 2 is used.

If the proxy cache contains an entry corresponding to the requested domain name/IP address, the proxy server program sends back the data to the client programin form of a response message.If the proxy server does not contain the requested IPaddress/domain name then the proxy server sends a request to the DNS server to which the DNS server responds with a response message to proxy server will be of Type-3 in this case.

If the proxy cache doesn't contain an entry corresponding to the requested domain name/IP address, then the proxy server will forward the request message to the DNS-server which is located at the Address 127.0.0.1 and serving at the port 8080. After establishing a connection with the DNS server, the proxy server will now forward the request to the DNS server, which will check if there is an entry corresponding to the request. If there doesn't exist any entry, the DNS server sends a message to the proxy server saying that it can't service the request, which the proxy server will forward to the client program.In this case a response message of Type-4 is send back to the proxy server, which will eventually forward it to the client.

If a corresponding entry exists, the DNS server will send the required data to the proxy server, which will now update its proxy cache using the FIFO algorithm, and then it forwards the data to the client program.

We have implemented the proxy cache and the DNS database as text programs with the names proxyCache and dnsCache.txt

/----------