SSN College of Engineering
UCS1511 - Network Lab
Exercise 5: Domain name server using UDP
Batch : 2018 - 2022

Prasanna Kumaran D
185001110

October 3, 2020

---

1 Domain name server using UDP

1.1 Aim

To simulate the concept of Domain Name Server using UDP.

1.2 Algorithm

1.2.1 Server

1. Create a socket descriptor with socket() system call and use AF_INET as domain and SOCK_STREAM for domain and communication type, store the socket descriptor in sockfd.
2. 2. If sockfd is a negative number,
   (a) print socket creation failed, terminate program.
3. Assign family, address and port to the server socketadd_in object. Set the family as AF_INET to access IPv4 protocols, and INADDR_ANY for address to accept connections from any client.
4. bind the socket to the server sockadd_in object
5. If bind is non zero,
   (a) Print bind creation failed and terminate.
6. Read a request from the client using the **recvfrom** system call

- if the domain is found in the DNS table then send the corresponding IP address of the domain to the requesting client.
- if the domain is not found add the domain name with its corresponding IP to the DNS table

7. Return the IP address of the domain requested to the client using **sendto** system call
8. Modify the table contents if required, an invalid IP or pre-existing IP address is notified to the user and table should **NOT** be updated

1.2.2 Client

1. Create a socket descriptor using socket()
   with AF_INET(IPv4 domain), SOCK_STREAM(connection type)
2. if socket < 0
   (a) Print socket creation failed and terminate program
3. Create a sockaddr_in object for the client and set up family, address and port number.
4. While the client is requesting for IP address loop
   - Request IP from the server using **sendto** system call
   - Receive the IP from server using the **recvfrom** system call and display
5. Close the socket connection using close() and terminate program

1.3 Program

1.3.1 Server

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<fcntl.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
typedef struct dns_table{
        char domain_name[50];
        char ip_address[16];
        struct dns_table* next;
}dns;
```

```
dns* create_Node(){
        dns* head = (dns*)malloc(sizeof(dns));
        head->next = NULL;
        return head;
        }

void add_node(dns* head, char domain_name[50], char ip_address[16]){
        dns* newLink = (dns*)malloc(sizeof(dns));
        strcpy(newLink->domain_name, domain_name);
        strcpy(newLink->ip_address, ip_address);
        newLink->next = NULL;
        newLink->next = head->next;
        head->next = newLink;
}
void printTable(dns * head)
{
        dns* temp = head->next;
        printf("Domain \t IP \n");
        while(temp != NULL){
                printf("%s \t %s\n", temp->domain_name, temp->ip_address);
                temp = temp->next;
        }
}
int main(int argc,char **argv)
{
int len;
char ip_add[16];
int sockfd, newfd,n;
int get_req_n;
char dom[50];
char buffer[50];
char ret_ip[16];
int found;
struct sockaddr_in servaddr,cliaddr;
dns* table = create_Node();
sockfd=socket(AF_INET,SOCK_DGRAM,0);
if(sockfd < 0)
        perror("cannot create socket");
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=INADDR_ANY;
servaddr.sin_port=htons(8083);
if(bind(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
        perror("Bind error");
```

```
len = sizeof(cliaddr);
while(1){
found = 0;
get_req_n = recvfrom(sockfd, buffer, sizeof(buffer), MSG_WAITALL,
                                (struct sockaddr *)&cliaddr, &len);
if(get_req_n != 0){
        printf("DNS request for %s\n", buffer);
        for(dns* temp = table; temp!= NULL && found == 0;
                                        temp = temp->next){
                if (strcmp(temp->domain_name, buffer) == 0){
                        printf("Domain found!. Sending IP
                                        information to the client...\n");
                        printf("Domain name : %s\n", buffer);
                        found = 1;
                        strcpy(ret_ip, temp->ip_address);
                        }
                }
        if (found == 0){
                printf("Domain not found!. Adding the domain to the
                                        DNS table...\n");
                scanf("%s", ip_add);
                add_node(table, buffer, ip_add);
                printf("Table updated!\n");
                strcpy(ret_ip, ip_add);
        }
        int send_req = sendto(sockfd, ret_ip, sizeof(ret_ip),
            MSG_WAITALL,(struct sockaddr*)&cliaddr, sizeof(cliaddr));
        printf("Request cleared\n");
        }
printf("Do you want to modify the table ? (Y/n)\n");
char choice;
scanf(" %c", &choice);
if (choice == 'Y'){
        printf("Enter domain name :");
        scanf("%s", dom);
        for (dns* temp = table; temp!= NULL; temp = temp->next){
        if (strcmp(temp->domain_name, dom)==0){
                printf("Old IP-Address : %s\n", temp->ip_address);
                printf("Enter new IP address :");
                scanf("%s", ip_add);
                int flag = 0;
        for (dns* temp_ip = table; temp_ip != NULL; temp_ip = temp_ip->next)
        {       if (strcmp(temp_ip->ip_address, ip_add) == 0){
                        printf("Invalid IP. Already exists. \n");
                        flag = 1;
```

```
                        printf("Table not updated...\n\n");
                        break;
                        }
        }
                if (flag == 0){
                strcpy(temp->ip_address, ip_add);
                printf("Table Updated...\n\n");
                printf("Domain : %s\nIP : %s", temp->domain_name,
                                                temp->ip_address);
                        }
                }
        }
}
printTable(table);
}
close(sockfd);
 return 0;
}
```

1.3.2 Client

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<string.h>
int main(int argc, char **argv)
{
    unsigned int len;
    int sockfd, n = 0;
    struct sockaddr_in servaddr, cliaddr;
    char buff[50];
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if(sockfd < 0)
        perror("Cannot create socket!\n");
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(8083);
    len = sizeof(servaddr);
    while(1) {
        printf("Domain Name: ");
        scanf("%s", buff);
```

```
        n = sendto(sockfd, buff, sizeof(buff), MSG_DONTWAIT,
            (struct sockaddr*)&servaddr, sizeof(servaddr));
        n = recvfrom(sockfd, buff, sizeof(buff), MSG_WAITALL,
            (struct sockaddr*)&servaddr, &len);
        printf("The IP address is %s\n", buff);
    }
    close(sockfd);
    return 0;
}
```

1.4 Output

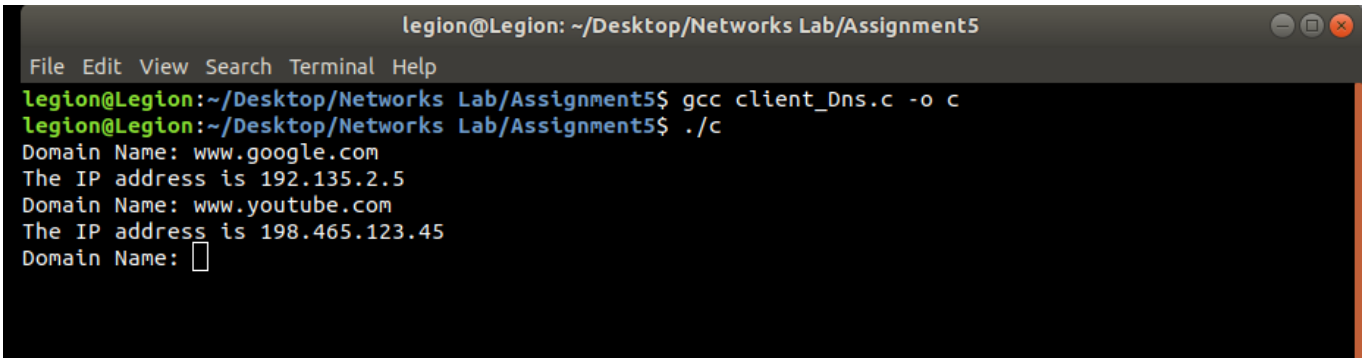

Figure 1: Client side - client 1

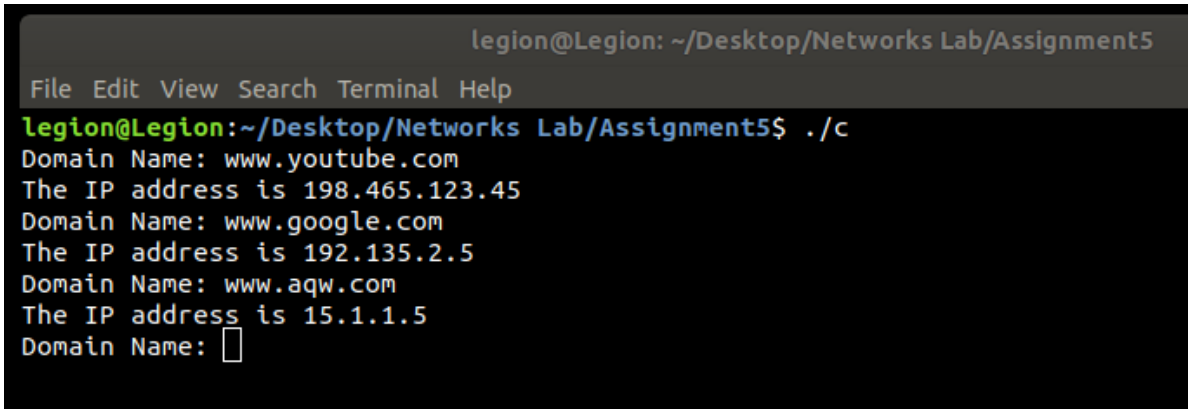

Figure 2: Client side - client 2

```
File  Edit  View  Search  Terminal  Help
legion@Legion:~/Desktop/Networks Lab/Assignment5$ gcc server_Dns.c -o s
legion@Legion:~/Desktop/Networks Lab/Assignment5$ ./s
DNS request for www.google.com
Domain not found!. Adding the domain to the DNS table...
192.135.2.5
Table updated!
Request cleared
Do you want to modify the table ? (Y/n)
n
Domain   IP
www.google.com   192.135.2.5
DNS request for www.youtube.com
Domain not found!. Adding the domain to the DNS table...
198.465.123.45
Table updated!
Request cleared
Do you want to modify the table ? (Y/n)
n
Domain   IP
www.youtube.com          198.465.123.45
www.google.com   192.135.2.5
DNS request for www.youtube.com
Domain found!. Sending IP information to the client...
Domain name : www.youtube.com
Request cleared
Do you want to modify the table ? (Y/n)
n
Domain   IP
www.youtube.com          198.465.123.45
www.google.com   192.135.2.5
DNS request for www.google.com
Domain found!. Sending IP information to the client...
Domain name : www.google.com
Request cleared
Do you want to modify the table ? (Y/n)
Y
Enter domain name :www.google.com
Old IP-Address : 192.135.2.5
Enter new IP address :137.568.45.6
Table Updated...

Domain : www.google.com
IP : 137.568.45.6Domain          IP
www.youtube.com          198.465.123.45
www.google.com   137.568.45.6
DNS request for www.aqw.com
Domain not found!. Adding the domain to the DNS table...
15.1.1.5
Table updated!
Request cleared
Do you want to modify the table ? (Y/n)
Y
Enter domain name :www.google.com
Old IP-Address : 137.568.45.6
Enter new IP address :198.465.123.45
Invalid IP. Already exists.
Table not updated...
```

Figure 3: Server side

1.5 Learning Outcomes

- Learnt how to simulate domain name server using UDP
- Learnt about the basic syntax and system calls used in socket programming
- Learnt how to handle errors in socket programming
- Learnt how to handle exceptions when table update was required
- Learnt how to establish and use a connection-less protocol