UCS1511 - COMPUTER NETWORKS

Simulation of RARP

REG NO : 205001085 **EX.NO : 8**
NAME : SABARIVASAN
DATE : 01.10.22

OBJECTIVE :

To be proficient in developing an application to simulate the functionality of RARP protocol using socket programming in C

CODE :

CLIENT :

```c
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include<string.h>
#include<ctype.h>
#include <stdio.h>
#include <sys/ioctl.h>
#include <netinet/in.h>
#include <net/if.h>
#include <unistd.h>
#include <arpa/inet.h>
char * MAC()
{
    struct ifreq s;
    int fd = socket(PF_INET, SOCK_DGRAM, IPPROTO_IP);
    char temp[100],*mac=(char*)malloc(sizeof(char)*500);
    strcpy(mac,"");
    strcpy(s.ifr_name, "eth0");
    if (0 == ioctl(fd, SIOCGIFHWADDR, &s)) {
```

```
        int i;
        for (i = 0; i < 6; ++i)
        {
            sprintf(temp,"%02x%s", (unsigned char)
s.ifr_addr.sa_data[i],(i<5?":":""));
            strcat(mac,temp);
        }
    }
    return mac;
}

char * IP()
{
    int n;
    char *ip=(char*)malloc(sizeof(char)*500);
    struct ifreq ifr;
    char array[] = "eth0";
    n = socket(AF_INET, SOCK_DGRAM, 0);
    //Type of address to retrieve - IPv4 IP address
    ifr.ifr_addr.sa_family = AF_INET;
    //Copy the interface name in the ifreq structure
    strncpy(ifr.ifr_name , array , IFNAMSIZ - 1);
    ioctl(n, SIOCGIFADDR, &ifr);
    close(n);
    //display result
    sprintf(ip,"%s", inet_ntoa(( (struct sockaddr_in *)&ifr.ifr_addr
)->sin_addr) );
    return ip;
}

int main(int argc, char const* argv[])
{
    int port=atoi(argv[1]);
    int sockD = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in servAddr;
    servAddr.sin_family = AF_INET;
    servAddr.sin_port= htons(port);
    servAddr.sin_addr.s_addr = inet_addr("172.29.136.138");
    char *ip=IP();
    char *mac=MAC();
```

```
    printf("\nMy MAC:%s\n",mac);
    printf("\nMy IP:%s\n",ip);
    int connectStatus= connect(sockD, (struct
sockaddr*)&servAddr,sizeof(servAddr));
    if (connectStatus == -1)
        printf("Error...\n");
    else
    {
        char strData[255];
        recv(sockD, strData, sizeof(strData),0);

        printf("\nRARP request received!\n");

        if(!strcmp(strData,mac))
            {
            strcpy(strData,ip);
            printf("MAC matched!\nIP sent!\n");
            }
        else
        {
            printf("\nMAC NOT MATCHED! MAC = %s received\n",strData);
            strcpy(strData,"False");
        }
        send(sockD, strData, sizeof(strData), 0);
        if(!strcmp(strData,mac))
        {
            recv(sockD, strData, sizeof(strData),0);
            printf("Ack: %s\n",strData);
        }
    }

    return 0;
 }
```

SERVER :

```
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <string.h>
#include <ctype.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <netinet/in.h>
#include <net/if.h>
#include <unistd.h>
#include <arpa/inet.h>
char *MAC()
{
    struct ifreq s;
    int fd = socket(PF_INET, SOCK_DGRAM, IPPROTO_IP);
    char temp[100], *mac = (char *)malloc(sizeof(char) * 500);
    strcpy(mac, "");
    strcpy(s.ifr_name, "eth0");
    if (0 == ioctl(fd, SIOCGIFHWADDR, &s))
    {
        int i;
        for (i = 0; i < 6; ++i)
        {
            sprintf(temp, "%02x%s", (unsigned char)s.ifr_addr.sa_data[i],
(i < 5 ? ":" : ""));
            strcat(mac, temp);
        }
    }
    return mac;
}

char *IP()
{
```

```
    char *ip = (char *)malloc(sizeof(char) * 500);
    struct ifreq ifr;
    char array[] = "eth0";
    int n = socket(AF_INET, SOCK_DGRAM, 0);
    ifr.ifr_addr.sa_family = AF_INET;
    strncpy(ifr.ifr_name, array, IFNAMSIZ - 1);
    ioctl(n, SIOCGIFADDR, &ifr);
    close(n);
    // display result
    sprintf(ip, "%s", inet_ntoa(((struct sockaddr_in
*)&ifr.ifr_addr)->sin_addr));
    return ip;
}
int main(int argc, char const *argv[])
{
    int port = atoi(argv[1]);
    int servSockD = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in servAddr;
    servAddr.sin_family = AF_INET;
    servAddr.sin_port = htons(port);
    servAddr.sin_addr.s_addr = INADDR_ANY;
    bind(servSockD, (struct sockaddr *)&servAddr, sizeof(servAddr));
    listen(servSockD, 3);
    char strData[30];
    char *ip = IP();
    char *mac = MAC();
    printf("\nMy MAC:%s", mac);
    printf("\nMy IP:%s\n", ip);
    printf("Enter Destination MAC: ");
    scanf("%s", strData);
    // strcpy(strData,"40:a8:f0:5c:e7:d4");
    printf("\nRARP request broadcasted..\n\nWaiting for Reply..\n\n");

    // number of clients need for connection ... ( noc )
    int noc = 1;
    int clientsocket[noc];
    for (int i = 0; i < noc; i++)
        clientsocket[i] = accept(servSockD, NULL, NULL);
    for (int i = 0; i < noc; i++)
        send(clientsocket[i], strData, sizeof(strData), 0);
```

```
    int flag = 0;
    for (int i = 0; i < noc; i++)
    {
        recv(clientsocket[i], strData, sizeof(strData), 0);
        if (strcmp(strData, "False"))
        {
            printf("IP of the accepted client %s\n", strData);
            strcpy(strData, "Received");
            send(clientsocket[i], strData, sizeof(strData), 0);
            printf("\nAcknowledgement Sent!\n");
            flag = 1;
            break;
        }
    }
    if (!flag)
        printf("\nRARP requeted failed\n");
    return 0;
}
```

OUTPUT :