EX6 – ADDRESS RESOLUTION PROTOCOL

- S. Vishakan CSE – C 18 5001 196

Server Program:

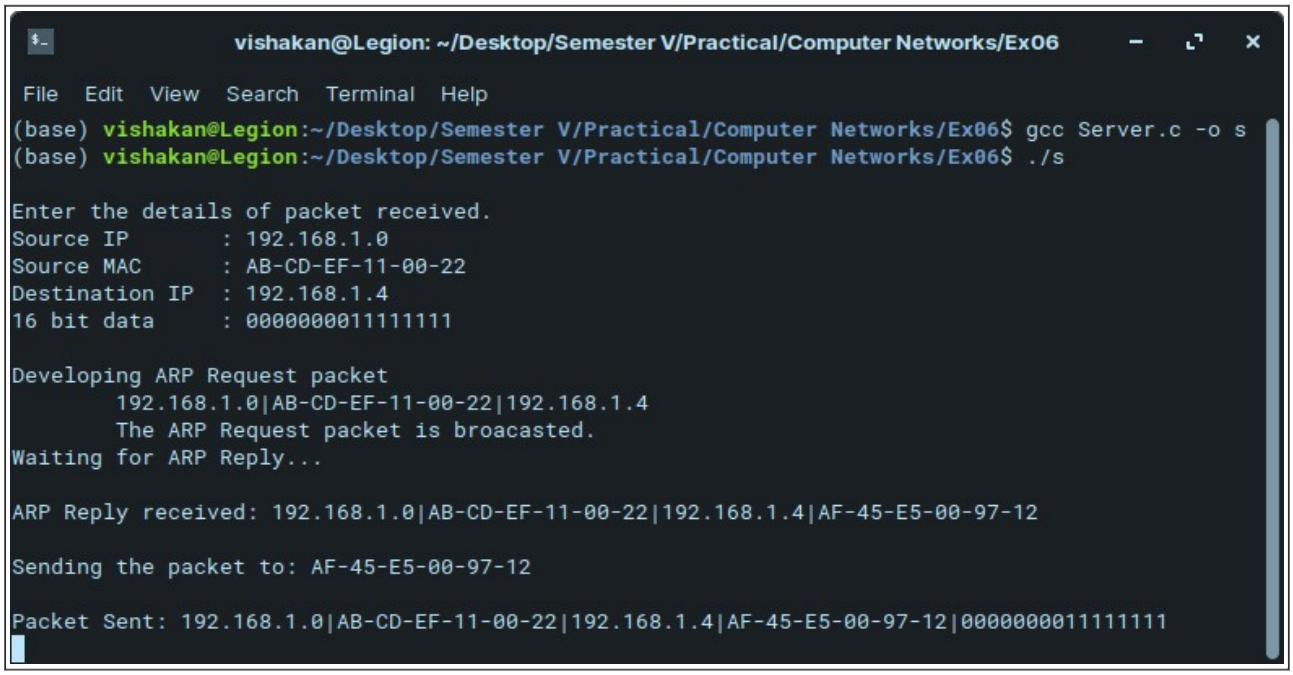
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <sys/socket.h>

struct ARP_PACKET{
        char SRC_IP[100];
        char DEST_IP[100];
        char SRC_MAC[100];
        char DEST_MAC[100];
        char DATA[100];
        char PKT[600];
};

typedef struct ARP_PACKET arp;

arp createARPPacket(void);

int main(int argc, char **argv){
        struct sockaddr_in server, client;
        char buffer[1024];
        int client_sockets[10], max, fd, sockfd, newfd, ping;
        int k, i, len, count;
        fd_set newfds;
        arp packet;

        packet = createARPPacket();
        printf("\nDeveloping ARP Request packet\n");
        printf("\t%s\n", packet.PKT);
        printf("\tThe ARP Request packet is broacasted.\n");
        printf("Waiting for ARP Reply...\n");

        for(i = 0; i < 10; i++){
                client_sockets[i] = 0;
        }

        sockfd = socket(AF_INET, SOCK_STREAM, 0);

        if(sockfd < 0){
                perror("Unable to open socket.\n");
```

```
}

bzero(&server, sizeof(server));

server.sin_family = AF_INET;
server.sin_addr.s_addr = INADDR_ANY;
server.sin_port = htons(7228);

if(bind(sockfd, (struct sockaddr*)&server, sizeof(server)) < 0){
        perror("Bind error occurred.\n");
}

listen(sockfd, 10);

len = sizeof(client);

while(1){
        FD_ZERO(&newfds); //Clears socket set.
        FD_SET(sockfd, &newfds); //Add sockfd to socket set.
        max = sockfd;

        for(i = 0; i < 10; i++){
                fd = client_sockets[i];
                if(fd > 0){
                        FD_SET(fd, &newfds);
                }

                if(fd > max){ //Store the max valued FD.
                        max = fd;
                }
        }

        //Wait indefinitely till any client pings.
        ping = select(max+1, &newfds, NULL, NULL, NULL);

        if(ping < 0){
                perror("Select error occurred.\n");
        }

        //if sockfd change => new connection request.
        if(FD_ISSET(sockfd, &newfds)){
                newfd = accept(sockfd, (struct sockaddr*)&client, &len);
                if(newfd < 0){
                        perror("Unable to accept the new connection.\n");
                }

                strcpy(buffer, packet.PKT);
                send(newfd, buffer, sizeof(buffer), 0);

                //Add the new client on an empty slot.
```

```
                        for(i = 0; i < 10; i++){
                                if(client_sockets[i] == 0){
                                        client_sockets[i] = newfd;
                                        break;
                                }
                        }
                }

                //Broadcast on all established connections
                for(i = 0; i < 10; i++){
                        fd = client_sockets[i];
                        bzero(buffer, sizeof(buffer));

                        //Check for change in FD
                        if(FD_ISSET(fd, &newfds)){
                                recv(fd, buffer, sizeof(buffer), 0);

                                //Check ARP response
                                if(buffer[0]){
                                        printf("\nARP Reply received: %s\n", buffer);
                                        count = 0;
                                        k = 0;
                                        for(i = 0; buffer[i]; i++){
                                                if(count == 3){
                                                        packet.DEST_MAC[k++] = buffer[i];
                                                }
                                                if(buffer[i] == '|'){
                                                        count++;
                                                }
                                        }
                                        packet.DEST_MAC[k] = '\0';

                                        printf("\nSending the packet to: %s\n",
                                                packet.DEST_MAC);
                                        bzero(buffer, sizeof(buffer));

                                        strcpy(buffer, packet.PKT);
                                        strcat(buffer, "|");
                                        strcat(buffer, packet.DEST_MAC);
                                        strcat(buffer, "|");
                                        strcat(buffer, packet.DATA);
                                        send(newfd, buffer, sizeof(buffer), 0);
                                        printf("\nPacket Sent: %s\n", buffer);
                                }
                        }
                }

        }

        return 0;
}
```

```
arp createARPPacket(void){
        arp packet;

        printf("\nEnter the details of packet received.\n");
        printf("Source IP\t: ");
        scanf(" %s", packet.SRC_IP);
        printf("Source MAC\t: ");
        scanf(" %s", packet.SRC_MAC);
        printf("Destination IP\t: ");
        scanf(" %s", packet.DEST_IP);
        printf("16 bit data\t: ");
        scanf(" %s", packet.DATA);

        strcpy(packet.PKT, packet.SRC_IP);
        strcat(packet.PKT, "|");
        strcat(packet.PKT, packet.SRC_MAC);
        strcat(packet.PKT, "|");
        strcat(packet.PKT, packet.DEST_IP);

        return packet;
}
```

Output:

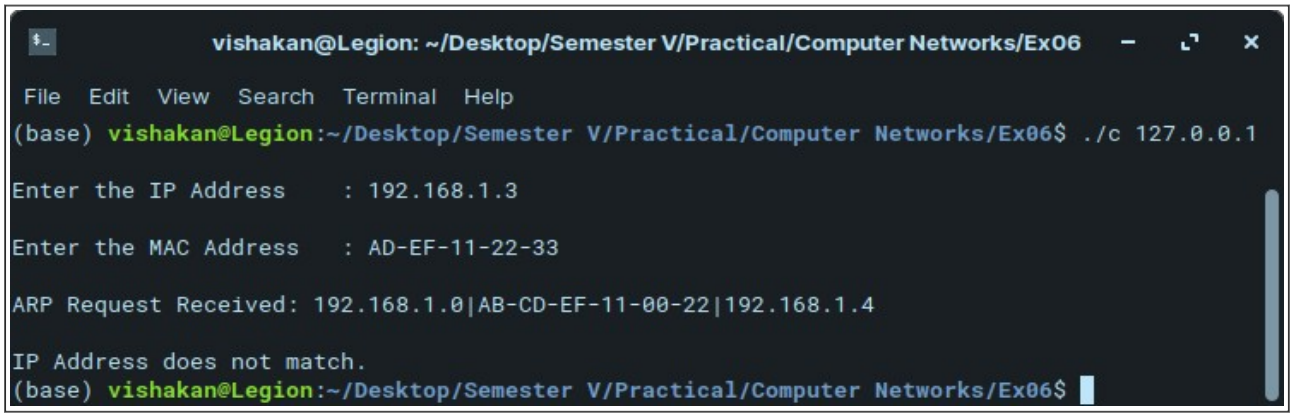


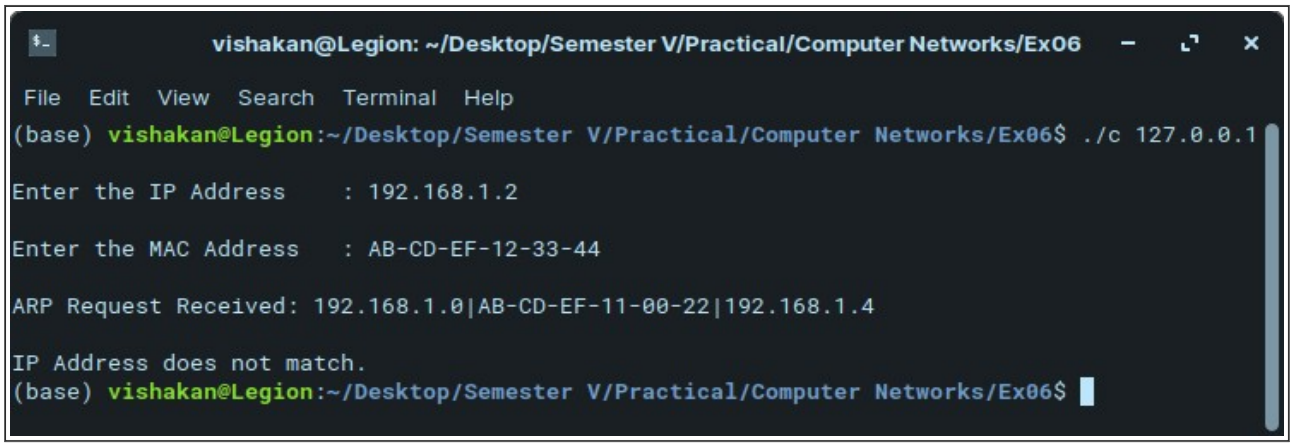
Client Program:

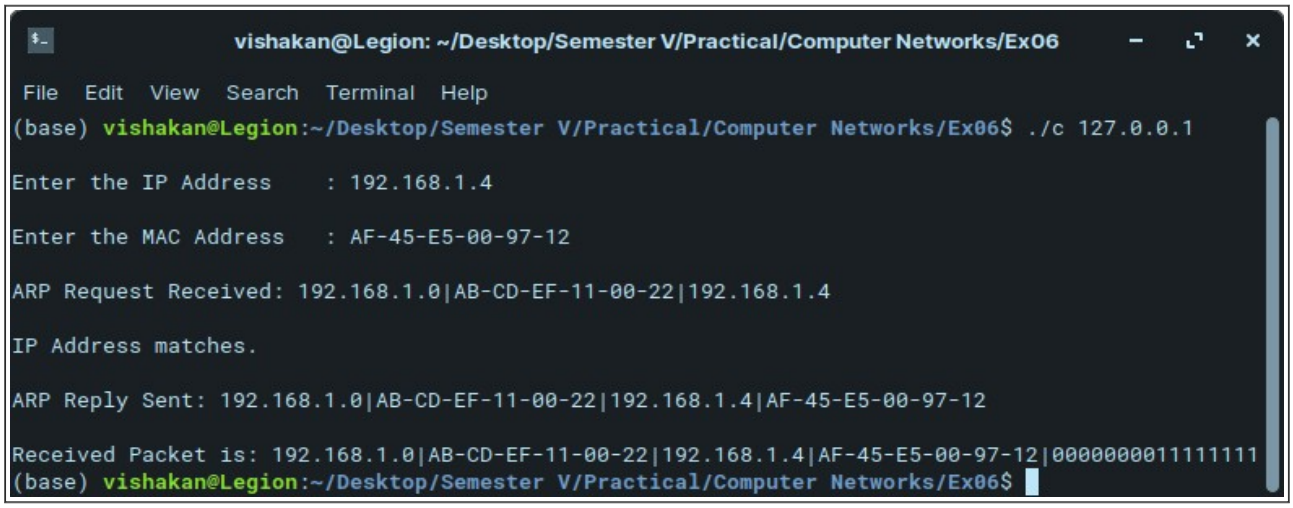
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <sys/socket.h>

struct ARP_PACKET{
        char SRC_IP[100];
        char DEST_IP[100];
        char SRC_MAC[100];
        char DEST_MAC[100];
        char DATA[16];
        char PKT[600];
};

typedef struct ARP_PACKET arp;

int main(int argc, char **argv){
        struct sockaddr_in server, client;
        char buffer[1024];
        int sockfd, newfd;
        int len, i, count, k;
        arp packet;

        printf("\nEnter the IP Address\t: ");
        scanf("%s", packet.DEST_IP);
        printf("\nEnter the MAC Address\t: ");
        scanf("%s", packet.DEST_MAC);

        sockfd = socket(AF_INET, SOCK_STREAM, 0);

        if(sockfd < 0){
                perror("Unable to open socket.\n");
        }

        bzero(&server, sizeof(server));

        server.sin_family = AF_INET;
        server.sin_addr.s_addr = inet_addr(argv[1]);
        server.sin_port = htons(7228);

        connect(sockfd, (struct sockaddr*)&server, sizeof(server));
        len = sizeof(client);

        bzero(buffer, sizeof(buffer));
        recv(sockfd, buffer, sizeof(buffer), 0);
        printf("\nARP Request Received: %s\n", buffer);
```

```
	count = 0;
	k = 0;
	for(i = 0; buffer[i]; i++){
		if(count == 2){
			packet.SRC_IP[k++] = buffer[i];
		}
		if(buffer[i] == '|'){
			count++;
		}
	}

	packet.SRC_IP[k] = '\0';

	if(strcmp(packet.SRC_IP, packet.DEST_IP) == 0){
		printf("\nIP Address matches.\n");
		strcat(buffer, "|");
		strcat(buffer, packet.DEST_MAC);
		send(sockfd, buffer, sizeof(buffer), 0);
		printf("\nARP Reply Sent: %s\n", buffer);

		bzero(buffer, sizeof(buffer));
		recv(sockfd, buffer, sizeof(buffer), 0);
		printf("\nReceived Packet is: %s\n", buffer);
	}

	else{
		printf("\nIP Address does not match.\n");
	}

	close(sockfd);

	return 0;
}
```

Output:


vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex06
File Edit View Search Terminal Help
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex06$ ./c 127.0.0.1
Enter the IP Address : 192.168.1.3
Enter the MAC Address : AD-EF-11-22-33
ARP Request Received: 192.168.1.0|AB-CD-EF-11-00-22|192.168.1.4
IP Address does not match.
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex06$


vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex06
File Edit View Search Terminal Help
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex06$ ./c 127.0.0.1
Enter the IP Address : 192.168.1.2
Enter the MAC Address : AB-CD-EF-12-33-44
ARP Request Received: 192.168.1.0|AB-CD-EF-11-00-22|192.168.1.4
IP Address does not match.
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex06$


vishakan@Legion: ~/Desktop/Semester V/Practical/Computer Networks/Ex06
File Edit View Search Terminal Help
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex06$ ./c 127.0.0.1
Enter the IP Address : 192.168.1.4
Enter the MAC Address : AF-45-E5-00-97-12
ARP Request Received: 192.168.1.0|AB-CD-EF-11-00-22|192.168.1.4
IP Address matches.
ARP Reply Sent: 192.168.1.0|AB-CD-EF-11-00-22|192.168.1.4|AF-45-E5-00-97-12
Received Packet is: 192.168.1.0|AB-CD-EF-11-00-22|192.168.1.4|AF-45-E5-00-97-12|0000000011111111
(base) vishakan@Legion:~/Desktop/Semester V/Practical/Computer Networks/Ex06$