Madhurya Mozumder

RA1911028010036

CSE-CC

I2

**ARP IMPLEMENTATION USING UDP**

**GIVEN REQUIREMENTS:**

There is a single host. The IP address of any Client in the network is given as input and the

corresponding hardware address is got as the output.

**TECHNICAL OBJECTIVE:**

Address Resolution Protocol (ARP) is implemented through this program. The IP address

of any Client is given as the input. The ARP cache is looked up for the corresponding hardware address.

This is returned as the output. Before compiling that Client is pinged.

**METHODOLOGY:**

Include the necessary header files.

Create a socket using socket function with family AF\_INET, type as SOCK\_DGRAM.

Declare structures arpreq ( as NULL structure, if required) and sockaddr\_in.

Initialize server address to 0 using the bzero function.

Assign the sin\_family to AF\_INET and sin\_addr using inet\_aton().

Using the object of arpreq structure assign the name of the Network Device to the data

member arp\_dev like, arp\_dev=”eth0”.

Ping the required Client.

Using the ioctl() we get the ARP cache entry for the given IP address.

The output of the ioctl() function is stored in the sa\_data[0] datamember of the arp\_ha

structure which is in turn a data member of structure arpreq.

Print the hardware address of the given IP address on the output console.

**CODE:**

#include<sys/types.h>

#include<sys/socket.h>

#include<net/if\_arp.h>

#include<sys/ioctl.h>

#include<stdio.h>

#include<string.h>

#include<unistd.h>

#include<math.h>

#include<complex.h>

#include<arpa/inet.h>

#include<netinet/in.h>

#include<netinet/if\_ether.h>

#include<net/ethernet.h>

#include<stdlib.h>

int main()

{

struct sockaddr\_in sin={0};

struct arpreq myarp={{0}};

unsigned char \*ptr;

int sd;

sin.sin\_family=AF\_INET;

printf("Enter IP address: ");

char ip[20];

scanf("%s", ip);

if(inet\_pton(AF\_INET,ip,&sin.sin\_addr)==0)

{

printf("IP address Entered '%s' is not valid \n",ip);

exit(0);

}

memcpy(&myarp.arp\_pa,&sin,sizeof(myarp.arp\_pa));

strcpy(myarp.arp\_dev,"echo");

sd=socket(AF\_INET,SOCK\_DGRAM,0);

printf("\nSend ARP request\n");

if(ioctl(sd,SIOCGARP,&myarp)==1)

{

printf("No Entry in ARP cache for '%s'\n",ip);

exit(0);

}

ptr=&myarp.arp\_pa.sa\_data[0];

printf("Received ARP Reply\n");

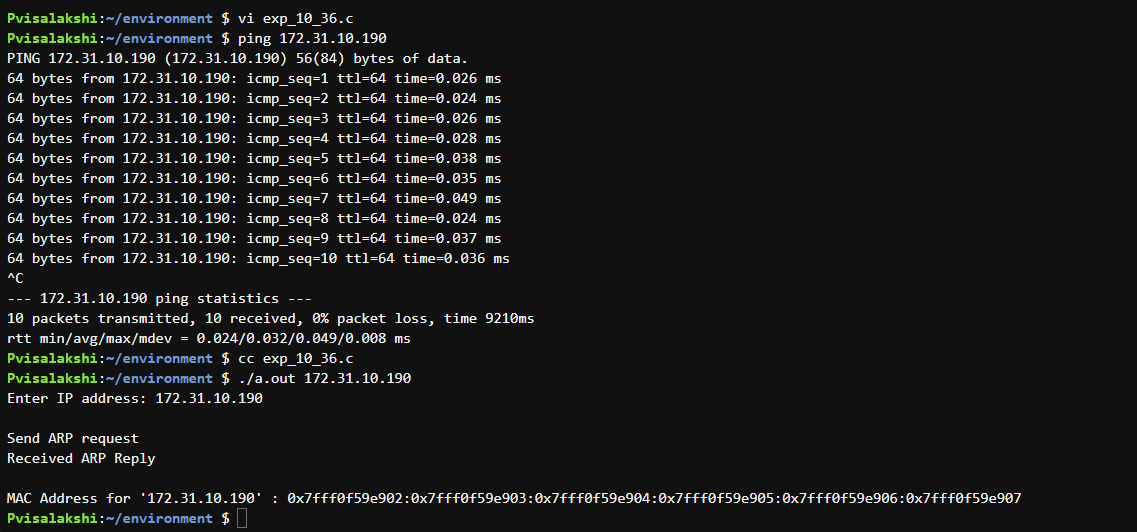
printf("\nMAC Address for '%s' : ",ip);

printf("%p:%p:%p:%p:%p:%p\n",ptr,(ptr+1),(ptr+2),(ptr+3),(ptr+4),(ptr+5));

return 0;

}

**OUTPUT :**

****

**INFERENCE:**

Thus the ARP implementation is developed to gets the MAC address of the remote machine’s IP

address from ARP cache and prints it.