Week : 10

AIM:

To write a program to get the MAC address from the system using Address

Resolution Protocol. (ARP Protocol)

ALGORITHM:

STEP 1: Start

STEP 2: Declare the variables and structure for the socket

STEP 3: Specify the family, protocol, IP address and port number

STEP 4: Create a socket using socket() function

STEP 5: Call memcpy() and strcpy functions

STEP 6: Display the MAC address

STEP 7: Stop

SOURCE 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(int argc,char \*argv[])

{

struct sockaddr\_in sin={0};

struct arpreq myarp={{0}};

unsigned char \*ptr;

int sd;

sin.sin\_family=AF\_INET;

if(inet\_aton(argv[1],&sin.sin\_addr)==0)

{

printf("IP address Entered '%s' is not valid \n",argv[1]);

exit(0);

}

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

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

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

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

{

printf("No Entry in ATP cache for '%s'\n",argv[1]);

exit(0);

}

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

printf("\nMAC Address for '%s' : ",argv[1]);

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

printf("\n\t\t\t\t%x:%x:%x:%x:%x:%x\n", myarp.arp\_ha.sa\_data[0],

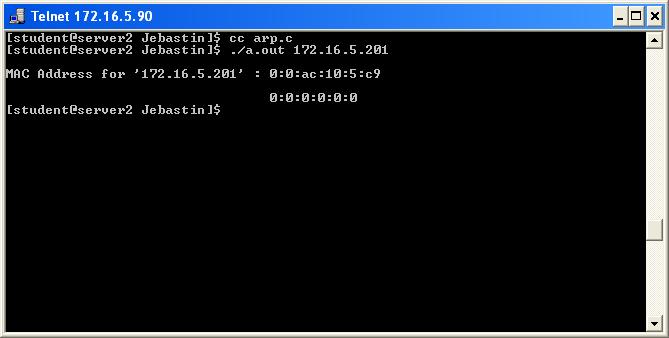
myarp.arp\_ha.sa\_data[1], myarp.arp\_ha.sa\_data[2],

myarp.arp\_ha.sa\_data[3], myarp.arp\_ha.sa\_data[4], myarp.arp\_ha.sa\_data[5]);

return 0;

}

OUTPUT:



Week : 11

Routing Protocols Using Border Gateway Protocol(BGP) in C

<https://forgetcode.com/c/1478-routing-protocols-using-border-gateway-protocol-bgp>

**Additional experiments**

**Aim: Connect the computers in Local Area Network.**

**Procedure: On the host computer**

On the host computer, follow these steps to share the Internet connection:

1. Log on to the host computer as Administrator or as Owner.

2. Click Start, and then click Control Panel.

3. Click Network and Internet Connections.

4. Click Network Connections.

5. Right-click the connection that you use to connect to the Internet.

For example, if you

connect to the Internet by using a modem, right-click the connection that you want under

Dial-up / other network available.

6. Click Properties.

7. Click the Advanced tab.

8. Under Internet Connection Sharing, select the Allow other network users to connectthrough this computer's Internet connection check box.

9. If you are sharing a dial-up Internet connection, select the Establish a dial-up connectionwhenever a computer on my network attempts to access the Internet check box if youwant to permit your computer to automatically connect to the Internet.

10. Click OK. You receive the following message:

When Internet Connection Sharing is enabled, your LAN adapter will be set to use IP address

192.168.0.1. Your computer may lose connectivity with other computers on your network. Ifthese other computers have static IP addresses, it is a good idea to set them to obtain their IPaddresses automatically. Are you sure you want to enable Internet Connection Sharing?

11. Click Yes.

The connection to the Internet is shared to other computers on the local area network (LAN).

The network adapter that is connected to the LAN is configured with a static IP address of192.168.0.1 and a subnet mask of 255.255.255.0

**On the client computer**

To connect to the Internet by using the shared connection, you must confirm the LAN adapterIP configuration, and then configure the client computer. To confirm the LAN adapter IPconfiguration, follow these steps:

1. Log on to the client computer as Administrator or as Owner.

2. Click Start, and then click Control Panel.

3. Click Network and Internet Connections.

4. Click Network Connections.

5. Right-click Local Area Connection and then click Properties.

6. Click the General tab, click Internet Protocol (TCP/IP) in the connection uses thefollowing items list, and then click Properties.

7. In the Internet Protocol (TCP/IP) Properties dialog box, click Obtain an IP addressautomatically (if it is not already selected), and then click OK.

Note: You can also assign a unique static IP address in the range of 192.168.0.2 to192.168.0.254. For example, you can assign the following static IP address, subnet mask, anddefault gateway:

8. IP Address 192.168.31.202

9. Subnet mask 255.255.255.0

10. Default gateway 192.168.31.1

11. In the Local Area Connection Properties dialog box, click OK.

12. Quit Control Panel

#### AIM:

Writeaprogramforcongestioncontrol usingLeakybucket algorithm.

#### Theory

The congesting control algorithms are basically divided into two groups: open loop and closed loop.Open loop solutions attempt to solve the problem by good design, in essence, to make sure it does notoccurinthefirstplace.Oncethesystemisupandrunning,midcoursecorrectionsarenotmade.Openloopalgorithmsarefurtherdividedintoones thatactatsource versusones thatactat thedestination.

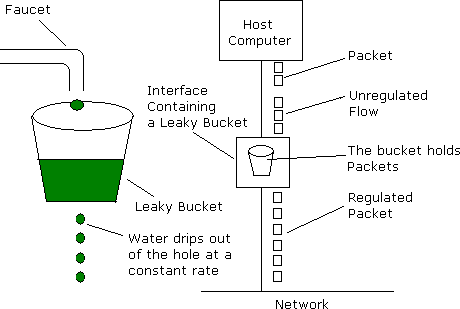
Incontrast,closedloopsolutionsarebasedontheconceptofafeedbackloopifthereisanycongestion.Closed loop algorithmsarealsodividedinto twosubcategories:explicitfeedback andimplicit feedback. In explicit feedback algorithms, packets are sent back from the point of congestion towarn the source. In implicit algorithm, the source deduces the existence of congestion by making localobservation,suchasthetime needed for acknowledgmenttocome back.

The presence of congestion means that the load is (temporarily) greater than the resources (in part ofthe system) can handle. For subnets that use virtual circuits internally, these methods can be used at thenetworklayer.

Another open loop method to help manage congestion is forcing the packet to be transmitted at a morepredictable rate. This approach to congestion management is widely used in ATM networks and is called**trafficshaping**.

The other method is the leaky bucket algorithm. Each host is connected to the network by an interfacecontainingaleakybucket,thatis,afiniteinternalqueue.Ifapacketarrivesatthequeuewhenitisfull,the packet is discarded. In other words, if one or more process are already queued, the new packet isunceremoniously discarded. Thisarrangementcan be builtinto the hardware interface orsimulate d bythe host operating system. In fact it is nothing other than a single server queuing system with constantservicetime.

The host is allowed to putonepacket per clock tick onto the network. This mechanism turnsanuneven flow of packet from the user process inside the host into an even flow of packet onto the network,smoothingoutbursts and greatlyreducingthe chancesof congestion.



#### Program

#include<iostream.h>#include<dos.h>#include<stdlib.h>#definebucketSize512

voidbktInput(inta,intb) {if(a>bucketSize)

cout<<"\n\t\tBucket overflow";else{

delay(500);while(a>b){

cout<<"\n\t\t"<<b<<" bytes outputted.";a-=b;

delay(500);

}

if (a>0) cout<<"\n\t\tLast "<<a<<" bytes sent\t";cout<<"\n\t\tBucketoutputsuccessful";

}

}

voidmain(){

intop, pktSize;randomize();

cout<<"Enter output rate : "; cin>>op;for(inti=1;i<=5;i++){

delay(random(1000));pktSize=random(1000);

cout<<"\nPacket no "<<i<<"\tPacket size = "<<pktSize;bktInput(pktSize,op);

}

}

#### SampleOutput

Enteroutputrate:100

Packetno0Packetsize=3

Bucket output successfulLast3bytes sent

Packetno1Packetsize=33

Bucket output successfulLast33bytessent

Packetno2Packetsize=117

Bucketoutputsuccessful

100 bytes outputted.Last17bytessent

Packetno3Packetsize=95

Bucket output successfulLast95bytessent

Packetno4Packetsize=949

Bucketoverflow