Aim:

To simulate ARP and RARP Protocols in java.

Algorithm:

ARP:

Server:

1.Start the program

2. Accept the socket which is created by the client.

3. Server maintains the table in which IP and corresponding MAC addresses are stored.

4. Read the IP address which is send by the client.

5. Map the IP address with its MAC address and return the MAC address to client.

Client:

1. Start the program

2. Using socket connection is established between client and server.

3. Get the IP address to be converted into MAC address.

4. Send this IP address to server.

5. Server returns the MAC address to client

RARP:

Server:

1. Start the program.

2. Server maintains the table in which IP and corresponding MAC addresses are stored.

3. Read the MAC address which is send by the client.

4. Map the IP address with its MAC address and return the IP address to client

Client:

1.Start the program

2. using datagram sockets UDP function is established.

3.Get the MAC address to be converted into IP address.

4.Send this MAC address to server.

5.Server returns the IP address to client

Code:

ArpServer.java:

```
package networkslab;
import java.net.*;
import java.io.IOException;
import java.util.*;
import java.io.*;

public class ArpServer {
	@SuppressWarnings("deprecation")
	public static void main(String args[]) {

		String ip="";

		String command="arp -a ";

		String out="";
		try {


			while(true) {
				DatagramSocket s=new DatagramSocket(7080);
				byte [] send=new byte[1024];
```

```
                                byte [] recv=new byte[1024];
DatagramPacket p=new DatagramPacket(recv,recv.length,InetAddress.getLocalHost(),7080);
                                s.receive(p);
                                ip=new String(p.getData());
                                String c_out=command+ip;
                                Process p1=Runtime.getRuntime().exec(c_out.trim());
                                System.out.println(c_out);
                                Scanner r=new Scanner(p1.getInputStream());

                                while(r.hasNext()) {
                                        out+=r.next();
                                        out+="\n";
                                }
                                send=out.getBytes();
                                out="";
DatagramPacket send1=new
DatagramPacket(send,send.length,InetAddress.getLocalHost(),p.getPort());
                                s.send(send1);
                                s.close();
                        }
                } catch (IOException e) {
                        e.printStackTrace();
                }
        }
}
```

ArpClient.java:

```
package networkslab;

import java.net.*;
import java.net.SocketException;
import java.io.*;
```

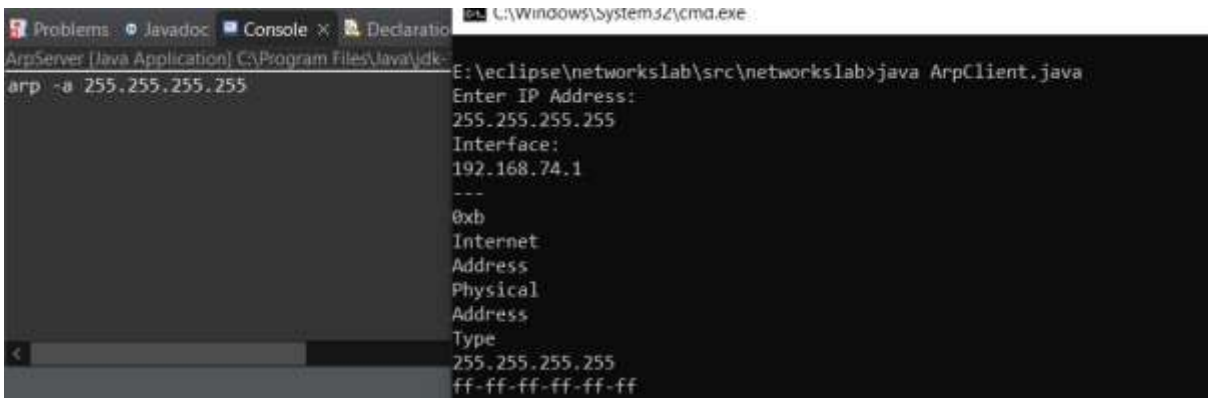
```
import java.util.*;
public class ArpClient {
	public static void main(String[] args) {
		Scanner r=new Scanner(System.in);
		try {
			DatagramSocket s1=new DatagramSocket();
			byte [] send=new byte[1024];
			byte [] recv=new  byte[1024];
			System.out.println("Enter IP Address:");
			String ip=r.next();
			send=ip.getBytes();
			DatagramPacket send1=new
DatagramPacket(send,send.length,InetAddress.getLocalHost(),7080);
			DatagramPacket recv1=new
DatagramPacket(recv,recv.length,InetAddress.getLocalHost(),7080);
			s1.send(send1);
			s1.receive(recv1);
			String s=new String(recv1.getData());
			System.out.println(s);
			s1.close();

		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

	}

}
```

OUTPUT:



RarpServer.java:

```
package networkslab;


import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;


public class RarpServer {
	public static void main(String[]args) {
		String ip[]={"165.165.80.80","165.165.79.1"};
		String mac[]={"6A:08:AA:C2","8A:BC:E3:FA"};


		String send2="";
		int flag=0;
		try {


			while(true) {
				DatagramSocket s=new DatagramSocket(8090);
				byte[] send=new byte[1024];
				byte [] recv=new byte[1024];
```

```
                                DatagramPacket recv1=new
DatagramPacket(recv,recv.length,InetAddress.getLocalHost(),8090);
                                s.receive(recv1);
                                String uin=new String(recv1.getData());
                                System.out.println(uin);
                                for(int i=0;i<ip.length;i++) {
                                        if(mac[i].equalsIgnoreCase(uin.trim())) {
                                                send2=ip[i];
                                                flag=1;
                                                break;
                                        }
                                }
                                if(flag==0) {
                                        send2="Not Found";

                                }
                                send=send2.getBytes();
                                DatagramPacket send1=new
DatagramPacket(send,send.length,InetAddress.getLocalHost(),recv1.getPort());
                                System.out.println(send2);
                                s.send(send1);
                                s.close();
                        }
                }catch(Exception e) {
                        e.printStackTrace();
                }
        }

}
```

RarpClient.java:

```
package networkslab;
import java.io.*;
import java.util.*;
import java.net.*;
public class RarpClient {
	public static void main(String[]args)
	{
		System.out.println("Enter MAC ADDRESS: ");
		Scanner s=new Scanner(System.in);
		String ip=s.next();
		try {
			DatagramSocket s1=new DatagramSocket();
			byte []send=new byte[1024];
			byte [] recv=new byte[1024];
			send=ip.getBytes();
			DatagramPacket p1=new
DatagramPacket(send,send.length,InetAddress.getLocalHost(),8090);
			s1.send(p1);
			DatagramPacket p2=new DatagramPacket(recv, recv.length,
InetAddress.getLocalHost(), 8090);
			s1.receive(p2);
			String out=new String(p2.getData());
			System.out.println(out);
		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

	}

}
```

Output:

```
E:\eclipse\networkslab\src\networkslab>javac RarpClient.java

E:\eclipse\networkslab\src\networkslab>java RarpClient.java
Enter MAC ADDRESS:
6A:08:AA:C2
165.165.80.80
```

Result:

Thus The ARP and RARP protocols were simulated in java.