Exercise 4 : Simulation of DNS Using Datagram Packets

Aim: To Simulate DNS using Datagram Packets.

Algorithm:

Server:

1. Run Process .exec command nslookup to get dns ip address from cmd
2. Create a datagram socket and bind it to a port
3. Create a datagram packet to receive client request
4. Read the domain name from client to be resolved.
5. Lookup the host array for the domain name
6. If found then retrieve corresponding address
7. Create a datagram packet and send ip address to client
8. Repeat steps 3-7 to resolve further requests from clients
9. Close the server socket

Client:

1. Create a datagram socket
2. Get domain name from user
3. Create a datagram packet and send domain name to the server
4. Create a datagram packet to receive server message
5. Read server's response
6. If ip address is found then display it else display "Domain does not exist"
7. Close the client socket

Code:

UdpDnsServer.java:

```
package networkslab;
import java.net.*;
import java.util.Scanner;
import java.io.*;
public class UdpDnsServer {
@SuppressWarnings("deprecation")
public static void main(String [] args) {
```

```
try {
        String command="nslookup ";
        String send1= "";
        String op="";

        while(true) {
                DatagramSocket s=new DatagramSocket(8210);
                byte[] send=new byte[1024];
                byte [] recv=new byte[1024];
                DatagramPacket rec=new DatagramPacket(recv,
recv.length,InetAddress.getLocalHost(),8210);
                s.receive(rec);
                String s12=new String (rec.getData());
                command+=s12;
                System.out.println(command);
                InetAddress a= rec.getAddress();
                int port=rec.getPort();

                Process p=Runtime.getRuntime().exec(command.trim());
                Scanner r=new Scanner(p.getInputStream());
                //DatagramPacket send=new DatagramPacket(recv, port, a, port)
                while(r.hasNext()) {
                        op+=r.next();
                        op+="\n";
                }

                System.out.println(op);
                send1+=op;
                send=send1.getBytes();
                send1="";
                DatagramPacket sendd=new DatagramPacket(send,
send.length,InetAddress.getLocalHost(),rec.getPort());
```

```
                    s.send(sendd);
                    s.close();
                }
                /*
                command+="www.google.com";
                Process p=Runtime.getRuntime().exec(command);
                Scanner r=new Scanner(p.getInputStream());
                while(r.hasNext()) {
                    op+=r.next();
                    op+="\n";

                }
                System.out.println(op);
                */
                } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
}
}
```

UdpDnsClient.java

```
package networkslab;
import java.net.*;
import java.io.*;
public class UdpDnsClient {
public static void main(String args[]) {
        BufferedReader r=new BufferedReader(new InputStreamReader(System.in));
        try {
                DatagramSocket s=new DatagramSocket();
                byte[] send=new byte[1024];
                byte [] recv=new byte[1024];
```

```
                System.out.println("Enter host Name:");
                String input=r.readLine();
                send=input.getBytes();
                DatagramPacket p=new DatagramPacket(send, send.length,
InetAddress.getLocalHost(), 8210);

                DatagramPacket q=new
DatagramPacket(recv,recv.length,InetAddress.getLocalHost(),8210);

                s.send(p);
                s.receive(q);
                String ip=new String(q.getData());
                System.out.println("IP Address: "+ip);

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

Output: