

Reg o: 192011478

Name: c. ramakrishna

CODE: CSA0734

EXPERIMENT: 22

AIM: To implement a DNS server and client in java using UDP sockets

ALGORITHM:

Server

1. Create an array of hosts and its ip address in another array
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
10. Stop

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 then display it else display "Domain does not exist"
7. Close the client socket

PROGRAM

// UDP DNS Server -- udpdnserver.java

```
import java.io.*;

import java.net.*;

public class udpdnserver

{ private static int indexOf(String[] array, String str)

{ str = str.trim(); for (int i=0; i < array.length; i++)

{ if (array[i].equals(str)) return i;

}

return -1;

}

public static void main(String arg[])throws IOException

{ String[] hosts = {"yahoo.com", "gmail.com", "cricinfo.com", "facebook.com"};

String[] ip = {"68.180.206.184", "209.85.148.19", "80.168.92.140", "69.63.189.16"};

System.out.println("Press Ctrl + C to Quit");

while (true)

{ DatagramSocket serversocket=new DatagramSocket(1362);

byte[] senddata = new byte[1021];

byte[] receivedata = new byte[1021];

DatagramPacket recvpack = new DatagramPacket(receivedata, receivedata.length);

serversocket.receive(recvpack); String sen = new String(recvpack.getData());

InetAddress ipaddress = recvpack.getAddress();

int port = recvpack.getPort();

String capsent; System.out.println("Request for host " + sen);

if(indexOf (hosts, sen) != -1)
```

```

capsent = ip[indexOf (hosts, sen)];

else capsent = "Host Not Found";

senddata = capsent.getBytes();

DatagramPacket pack = new DatagramPacket(senddata, senddata.length,ipaddress,port);

serversocket.send(pack);

serversocket.close(); } } }

```

//UDP DNS Client -- udpdnsclient.java

```

import java.io.*;

import java.net.*;

public class udpdnsclient { public static void main(String args[])throws IOException

{ BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

DatagramSocket clientsocket = new DatagramSocket(); InetAddress ipaddress;

if (args.length == 0)

ipaddress = InetAddress.getLocalHost();

else ipaddress = InetAddress.getByName(args[0]);

byte[] senddata = new byte[1024]; byte[] receivedata = new byte[1024];

int portaddr = 1362; System.out.print("Enter the hostname : ");

String sentence = br.readLine(); Senddata = sentence.getBytes(); DatagramPacket pack = new
DatagramPacket(senddata,senddata.length, ipaddress,portaddr);

clientsocket.send(pack);

DatagramPacket recvpack =new DatagramPacket(receivedata,receivedata.length);
clientsocket.receive(recvpack); String modified = new String(recvpack.getData());

System.out.println("IP Address: " + modified);

clientsocket.close(); }}

```

OUTPUT:

Server

```
$ javac udpdnsserver.java
```

```
$ java udpdnsserver Press Ctrl + C to Quit
```

```
Request for host yahoo.com
```

```
Request for host cricinfo.com
```

```
Request for host youtube.com
```

Client

```
$ javac udpdnsclient.java
```

```
$ java udpdnsclient
```

```
Enter the hostname : yahoo.com
```

```
IP Address: 68.180.206.184
```

```
$ java udpdnsclient
```

```
Enter the hostname : cricinfo.com
```

```
IP Address: 80.168.92.140
```

```
$ java udpdnsclient
```

```
Enter the hostname : youtube.com
```

```
IP Address: Host Not Found
```