CB8581 - Networks

Laboratory

Practical Examination DEC/2020

Class: CSE - III yr

Sec: A

Sem: 5th

Regno: 310618104047

Date: 28/12/2020

Simulation of DNS using UDP sockets

Aim: To write a code in java to simulate DNS (Domain name system).

Algorithm:

Server:

1. A UDP socket is created
2. An internet socket address structure is filled in with address and servers well known port.
3. Listen function is called to convert socket into listening socket
4. Server calls in the call to accept. Waiting for client connection to complete

5. when the connection is established, the server reads the domain name from client using readn method.
6. It then finds the corresponding address and using sendto(). Then send back to client.
7. Server closes connection socket.

Client :

1. UDP socket is created.
2. Internet socket address is filled with server's IP address and the corresponding port number.
3. Connect function establishes connection with the server.
4. The client reads the domain name for standard input using fgets function. Then writes it to server.
5. It then reads back server reply . Then outputs to the standard output.

dnsserver.java

```
import java.io.IOException;
import java.net;

Public class dnsserver {

        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 = { "poeticengineer.com", "gmail.com",
                           "google.com", "facebook.com"};
        String [] ip = { "172.28.251.59", "172.217.11.15",
                         "172.217.11.14", "31.13.17.36"} ;
        System.out.println("Press Ctrl+c to quit");
        while (true){
          DatagramSocket serversocket = new DatagramSocket
                                        (1362);
```

```
byte[] senddata = new byte[1021];
byte[] recievedata = new byte[1021];
Datagram recvpack = new DatagramPacket(recievedata,
                                       recievedata.length);
serversocket.recieve(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)];
    capsent = ip[indexOf(hosts, sen)];
else
    capsent = ip[indexOf "Host not found";
senddata = capsent.getBytes();
DatagramPacket pack = new DatagramPacket(senddata,
senddata.length, ipaddress, port);
serversocket.send(pack);
serversocket.close();
}
}
```

dnsclient.java:

```
import java.io;
import java.net;
Public class dnsclient{
    Public static void main(String args[]) throws Exception{
        BufferedReader br = new BufferedReader(new
        InputStreamReader (System.in));
        DatagramSocket clientsocket = new DatagramSocket();
        InetAddress ipaddress;
        if(args.length == 0)
            ipaddress = InetAddress.getlocalHost();
        else
            ip address = InetAddress.getByName(args[0]);
        byte[] senddata = new byte[1024];
        byte[] recievedata = new byte[1024];
        int portaddr = 1362;
        System.out.println("Enter the hostname ");
        String sentence = br.readline();
        senddata = sentence.getBytes();
        DatagramPacket pack = new DatagramPacket(senddata,
                senddata.length, ipaddress portaddr);
```

```
clientsocket.recieve(recvpack);
String modified = new String(recvpack.getData());
System.out.println("Ipaddress: " + modified
clientsocket.close();
}
}
```

Output:

Client:

Enter the hostname: google.com
IP Address: 172.217.11.14

Server:

Press Ctrl+C to quit
Request for host google.com

Result: Thus, a program for DNS simulation using UDP has been executed successfully.

CLIENT :

SERVER :