Ex.No: 4

Simulation of DNS using UDP Sockets

AIM

To write a java program for DNS application

PRE LAB DISCUSSION:

- The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities.
- The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. The following diagram shows the domain name space hierarchy.
- Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.
- Types of Name Servers
- Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server
- Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.
- Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.
- DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.
- The main function of DNS is to translate domain names into IP Addresses, which computers can understand. It also provides a list of mail servers which accept Emails for each domain name. Each domain name in DNS will nominate a set of name servers to be authoritative for its DNS records.

ALGORITHM

Server

1. Start
2. Create UDP datagram socket
3. Create a table that maps host name and IP address
4. Receive the host name from the client
5. Retrieve the client's IP address from the received datagram
6. Get the IP address mapped for the host name from the table.
7. Display the host name and corresponding IP address

8. Send the IP address for the requested host name to the client
9. Stop.

Client

1. Start
2. Create UDP datagram socket.
3. Get the host name from the client
4. Send the host name to the server
5. Wait for the reply from the server
6. Receive the reply datagram and read the IP address for the requested host name
7. Display the IP address.
8. Stop.

PROGRAM

DNS Server

```
java import java.io.*;
import java.net.*;
public class udpdnsserver
{
        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

```
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 ericinfo.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 : ericinfo.com
IP Address: 80.168.92.140
>java udpdnsclient
Enter the hostname : youtube.com
IP Address: Host Not Found
```

VIVA (Pre & Post Lab) QUESTIONS:

1. What layer is DNS?
2. What is pupose of network layer?
3. What is logical address
4. What type of transport protocol is used for DNS.
5. What is difference between IP address and DNS?
6. What is a DNS and how does it work?
7. Why do we need a Domain Name System?
8. What role does the DNS Resolver play in the DNS System.
9. What is DNS and its types?.
10. List the Two types of DNS message

RESULT:

Thus the java application program using UDP Sockets to implement DNS was developed and executed successfully