

**EX.NO:1** Learn to use commands like `tcpdump`, `netstat`, `ifconfig`, `nslookup` and `tracert`.  
Capture ping and `tracert` PDUs using a network protocol analyzer and examine.

**AIM:** To Learn to use commands like `tcpdump`, `netstat`, `ifconfig`, `nslookup` and `tracert` ping.

### PRE LAB DISCUSSION:

#### **Tcpdump:**

The `tcpdump` utility allows you to capture packets that flow within your network to assist in network troubleshooting. The following are several examples of using `tcpdump` with different options. Traffic is captured based on a specified filter.

#### **Netstat**

`Netstat` is a common command line TCP/IP networking available in most versions of Windows, Linux, UNIX and other operating systems.

`Netstat` provides information and statistics about protocols in use and current TCP/IP network connections.

#### **ipconfig**

`ipconfig` is a console application designed to run from the Windows command prompt. This utility allows you to get the IP address information of a Windows computer.

From the command prompt, type `ipconfig` to run the utility with default options. The output of the default command contains the IP address, network mask, and gateway for all physical and virtual network adapter.

#### **nslookup**

The `nslookup` (which stands for *name server lookup*) command is a network utility program used to obtain information about internet servers. It finds name server information for domains by querying the Domain Name System.

#### **Trace route:**

`Traceroute` is a network diagnostic tool used to track the pathway taken by a packet on an IP network from source to destination. `Traceroute` also records the time taken for each hop the packet makes during its route to the destination

### Commands:

#### Tcpdump:

##### **Display traffic between 2 hosts:**

To display all traffic between two hosts (represented by variables `host1` and `host2`): `# tcpdump host host1 and host2`

##### **Display traffic from a source or destination host only:**

To display traffic from only a source (`src`) or destination (`dst`) host:

`# tcpdump src host`

`# tcpdump dst host`

##### **Display traffic for a specific protocol**

Provide the protocol as an argument to display only traffic for a specific protocol, for example `tcp`, `udp`, `icmp`, `arp`

# tcpdump protocol

For example to display traffic only for the tcp traffic :

# tcpdump tcp

**Filtering based on source or destination port**

To filter based on a source or destination port:

# tcpdump src port ftp

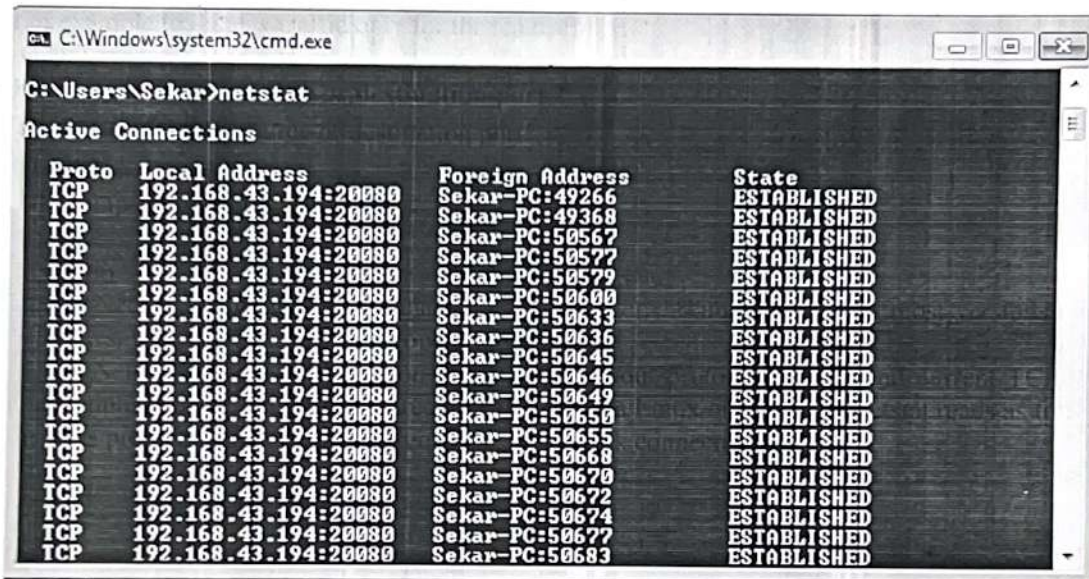
# tcpdump dst port http

## 2. Netstat

Netstat is a common command line TCP/IP networking available in most versions of Windows, Linux, UNIX and other operating systems.

Netstat provides information and statistics about protocols in use and current TCP/IP network connections. The Windows help screen (analogous to a Linux or UNIX for netstat reads as follows: displays protocol statistics and current TCP/IP network connections.

#netstat



```
C:\Windows\system32\cmd.exe
C:\Users\Sekar>netstat
Active Connections
Proto Local Address Foreign Address State
TCP 192.168.43.194:20080 Sekar-PC:49266 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:49368 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50567 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50577 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50579 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50600 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50633 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50636 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50645 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50646 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50649 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50650 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50655 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50668 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50670 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50672 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50674 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50677 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50683 ESTABLISHED
```

## 3. ipconfig

In Windows, ipconfig is a console application designed to run from the Windows command prompt. This utility allows you to get the IP address information of a Windows computer.

**Using ipconfig**

From the command prompt, type ipconfig to run the utility with default options. The output of the default command contains the IP address, network mask, and gateway for all physical and virtual network adapter.

#ipconfig



```
C:\Windows\system32\cmd.exe
C:\Users>ipconfig

Windows IP Configuration

Wireless LAN adapter Wireless Network Connection 3:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Wireless LAN adapter Wireless Network Connection 2:
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Wireless LAN adapter Wireless Network Connection:
    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : 2409:4072:616:44d0:61fd:d041:5a78:c2d0
    Temporary IPv6 Address. . . . . : 2409:4072:616:44d0:1093:b8ff:c0e:7b00
    Link-local IPv6 Address . . . . . : fe80::61fd:d041:5a78:c2d0%16
    IPv4 Address. . . . . : 192.168.43.194
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::d551:a02c:fa47:897c%16
```

#### 4. nslookup

The **nslookup** (which stands for *name server lookup*) command is a network utility program used to obtain information about internet servers. It finds name server information for domains by querying the Domain Name System.

The nslookup command is a powerful tool for diagnosing DNS problems. You know you're experiencing a DNS problem when you can access a resource by specifying its IP address but not its DNS name.

#nslookup

#### 5. Trace route:

Traceroute uses Internet Control Message Protocol (ICMP) echo packets with variable time to live (TTL) values. The response time of each hop is calculated. To guarantee accuracy, each hop is queried multiple times (usually three times) to better measure the response of that particular hop.

Traceroute is a network diagnostic tool used to track the pathway taken by a packet on an IP network from source to destination. Traceroute also records the time taken for each hop the packet makes during its route to the destination. Traceroute uses Internet Control Message Protocol (ICMP) echo packets with variable time to live (TTL) values.

The response time of each hop is calculated. To guarantee accuracy, each hop is queried multiple times (usually three times) to better measure the response of that particular hop. Traceroute sends packets with TTL values that gradually increase from packet to packet, starting with TTL value of one. Routers decrement TTL values of packets by one when routing and discard packets whose TTL value has reached zero, returning the ICMP error message ICMP Time Exceeded.

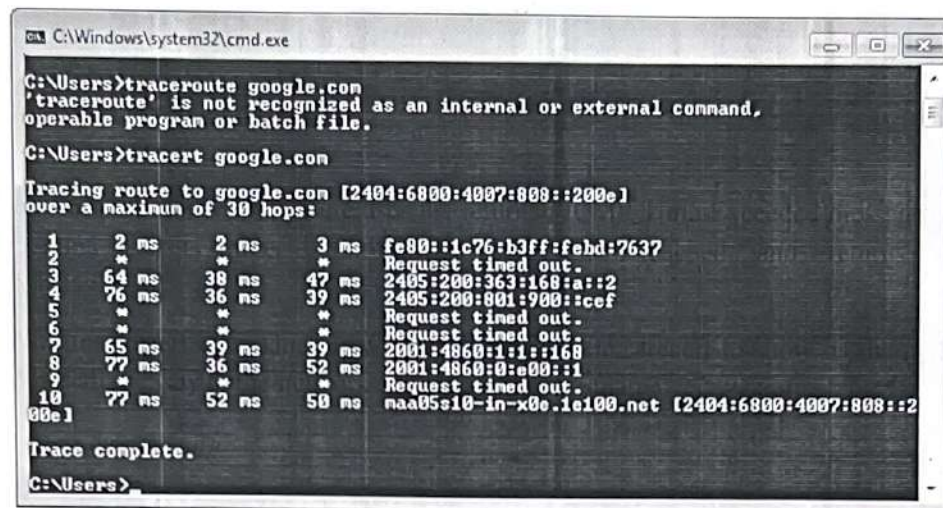
For the first set of packets, the first router receives the packet, decrements the TTL value and drops the packet because it then has TTL value zero. The router sends an ICMP Time Exceeded message back to the source. The next set of packets are given a TTL value of two, so the first router forwards the packets, but the second router drops them and replies with ICMP Time Exceeded.

Proceeding in this way, traceroute uses the returned ICMP Time Exceeded messages to build a list of routers that packets traverse, until the destination is reached and returns an ICMP Echo Reply message.

With the tracert command shown above, we're asking tracert to show us the path from the local computer all the way to the network device with the hostname

www.google.com.

#tracert google.com



```
C:\Windows\system32\cmd.exe
C:\Users>tracert google.com
'tracert' is not recognized as an internal or external command,
operable program or batch file.
C:\Users>tracert google.com
Tracing route to google.com [2404:6800:4007:808::200e]
over a maximum of 30 hops:
  0  2 ms    2 ms    3 ms  fe80::1c76:b3ff:febd:7637
  1  *        *        *      Request timed out.
  2  64 ms   38 ms   47 ms  2405:200:363:168:a::2
  3  76 ms   36 ms   39 ms  2405:200:801:900::cef
  4  *        *        *      Request timed out.
  5  *        *        *      Request timed out.
  6  65 ms   39 ms   39 ms  2001:4860:1::1:168
  7  77 ms   36 ms   52 ms  2001:4860:0:e00::1
  8  *        *        *      Request timed out.
  9  77 ms   52 ms   50 ms  naa05s10-in-x0e.1e100.net [2404:6800:4007:808::200e]
Trace complete.
C:\Users>
```

## 6. Ping:

The ping command sends an echo request to a host available on the network. Using this command, you can check if your remote host is responding well or not. Tracking and isolating hardware and software problems. Determining the status of the network and various foreign hosts. The ping command is usually used as a simple way to verify that a computer can communicate over the network with another computer or network device. The ping command operates by sending Internet Control Message Protocol (ICMP) Echo Request messages to the destination computer and waiting for a response

# ping 172.16.6.2



```
C:\Windows\system32\cmd.exe
er). and has no effect on the type of service field in the IP Head
-r count Record route for count hops (IPv4-only).
-s count Timestamp for count hops (IPv4-only).
-j host-list Loose source route along host-list (IPv4-only).
-k host-list Strict source route along host-list (IPv4-only).
-v timeout Timeout in milliseconds to wait for each reply.
-R Use routing header to test reverse route also (IPv6-only).
-S srcaddr Source address to use.
-4 Force using IPv4.
-6 Force using IPv6.

C:\Users>ping 172.16.6.2
Pinging 172.16.6.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.16.6.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users>
```

#### VIVA(Pre & Post Lab) QUESTIONS:

1. Define network
2. Define network topology
3. What is OSI Layers.
4. What is the use of netstat command?
5. What is nslookup command?
6. What is the purpose of traceroute command?
7. What is ping command.

#### RESULT:

Thus the various networks commands like tcpdump, netstat, ifconfig, nslookup and traceroute ping are executed successfully.

Ex.No: 2      Write a HTTP web client program to download a web page using TCP sockets

**AIM:**

To write a java program for socket for HTTP for web page upload and download .

**PRE LAB DISCUSSION:**

- HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.
- For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page.
- The other main standard that controls how the World Wide Web works is HTML, which covers how Web pages are formatted and displayed. HTTP functions as a request-response protocol in the client-server computing model.
- A web browser, for example, may be the client and an application running on a computer hosting a website may be the server.
- The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client.
- The response contains completion status information about the request and may also contain requested content in its message body.

**ALGORITHM:**

**Client:**

1. Start.
2. Create socket and establish the connection with the server.
3. Read the image to be uploaded from the disk
4. Send the image read to the server
5. Terminate the connection
6. Stop.

**Server:**

1. Start
2. Create socket, bind IP address and port number with the created socket and make server a listening server.
3. Accept the connection request from the client
4. Receive the image sent by the client.
5. Display the image.
6. Close the connection.
7. Stop.

## PROGRAM

### Client

```
import javax.swing.*;
import java.net.*;
import java.awt.image.*;
import javax.imageio.*;
import java.io.*;
import java.awt.image.BufferedImage; import
java.io.ByteArrayOutputStream; import
java.io.File;
import java.io.IOException; import
javax.imageio.ImageIO;
public class Client
{
    public static void main(String args[]) throws Exception
    {
        Socket soc;
        BufferedImage img = null;
        soc=new
        Socket("localhost",4000);
        System.out.println("Client is running.
        ");
        try {
            System.out.println("Reading image from disk. ");
            img = ImageIO.read(new File("digital_image_processing.jpg"));
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            ImageIO.write(img, "jpg", baos);
            baos.flush();
            byte[] bytes = baos.toByteArray(); baos.close();
            System.out.println("Sending image to server.");
            OutputStream out = soc.getOutputStream();
            DataOutputStream dos = new DataOutputStream(out);
            dos.writeInt(bytes.length);
            dos.write(bytes, 0, bytes.length);
            System.out.println("Image sent to server. ");
            dos.close();
            out.close();
        }
        catch (Exception e)
        {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```



```

        soc.close();
    }
    soc.close();
}

```

### Server

```

import java.net.*;
import java.io.*;
import java.awt.image.*;
import javax.imageio.*;
import javax.swing.*;
class Server
{
    public static void main(String args[]) throws Exception
    {
        ServerSocket server=null;
        Socket socket;
        server=new ServerSocket(4000);
        System.out.println("Server Waiting for image");
        socket=server.accept(); System.out.println("Client connected.");
        InputStream in = socket.getInputStream();
        DataInputStream dis = new DataInputStream(in);
        int len = dis.readInt();
        System.out.println("Image Size: " + len/1024 + "KB"); byte[] data = new byte[len];
        dis.readFully(data);
        dis.close();
        in.close();
        InputStream ian = new ByteArrayInputStream(data);
        BufferedImage bImage = ImageIO.read(ian);
        JFrame f= new JFrame("Server");
        ImageIcon icon = new ImageIcon(bImage);
        JLabel l = new JLabel();
        l.setIcon(icon);
        f.add(l);
        f.pack();
        f.setVisible(true);
    }
}

```

**OUTPUT:**



When you run the client code, following output screen would appear on client side:

```
Server Waiting for image  
Client connected.  
Image Size: 29KB
```

**VIVA(Pre & Post Lab) QUESTIONS:**

1. What is URL.
2. Why is http required?
3. What id http client?
4. what is WWW?
5. Compare HTTP and FTP.
6. Define Socket.
7. What do you mean by active web page?
8. What are the four Main properties of HTTP?

**RESULT:**

Thus the socket program for HTTP for web page upload and download was developed and executed successfully.