PROCEDURE:

There are many reasons that you may need to change and set a static IP address for your IP device, such as a managed switch, wireless router, or outdoor access point. One reason is because an installation scenario doesn't have an active network with DHCP services. Some other reasons you may need to set a static IP are because you use a dedicated web server, host server, VPN, or VoIP services.

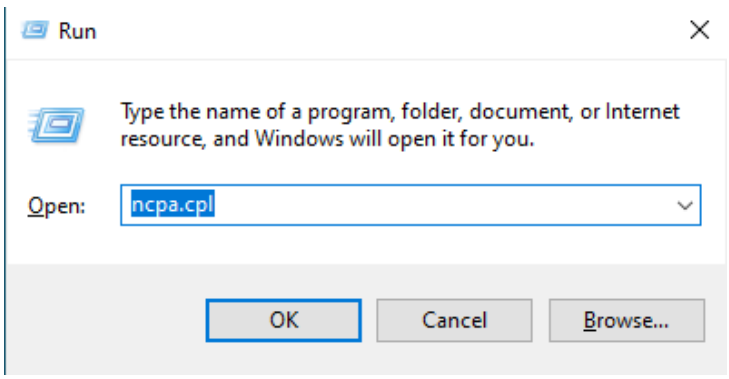
Setting static IP addresses can help to avoid network conflicts which could cause certain devices to stop working correctly. However, in most installation scenarios, users will use a regular network and will not need to use a static IP. Setting a static IP address is an advanced networking function, and a basic, fundamental knowledge of TCP/IP is needed.

In general, statically address devices outside of your DHCP pool range, which in most home networks is your router. For reference, the DHCP pool range for TREND net products is usually (but not always) 192.168.10.101 to 199.

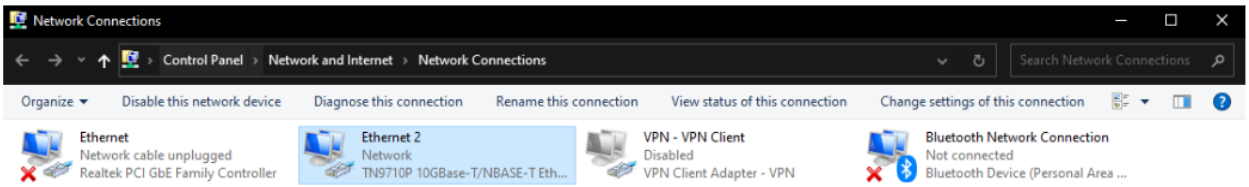
1. **Access the Control Panel**

- In the Windows search bar, type in "ncpa.cpl" and then press enter.

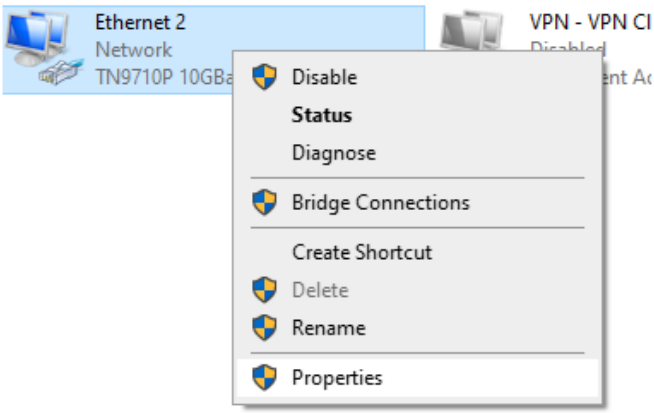
- If you are not using Windows 10, follow the steps below instead.
    - On your keyboard, press the "Windows" and "R" keys at the same time.
    - Enter "ncpa.cpl" in the window that pops up.
- Note: Network connections will display the network adapters that are currently connected to your computer.



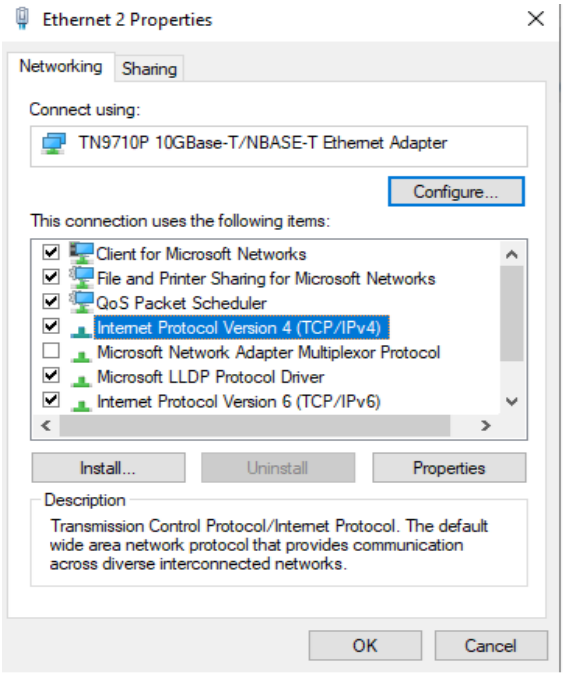
2. **Select the Network Adapter**
    - Right click on the network adapter that is currently connected to the device that you are trying to configure. Usually, it will be the adapter with the word “Ethernet” in the name.



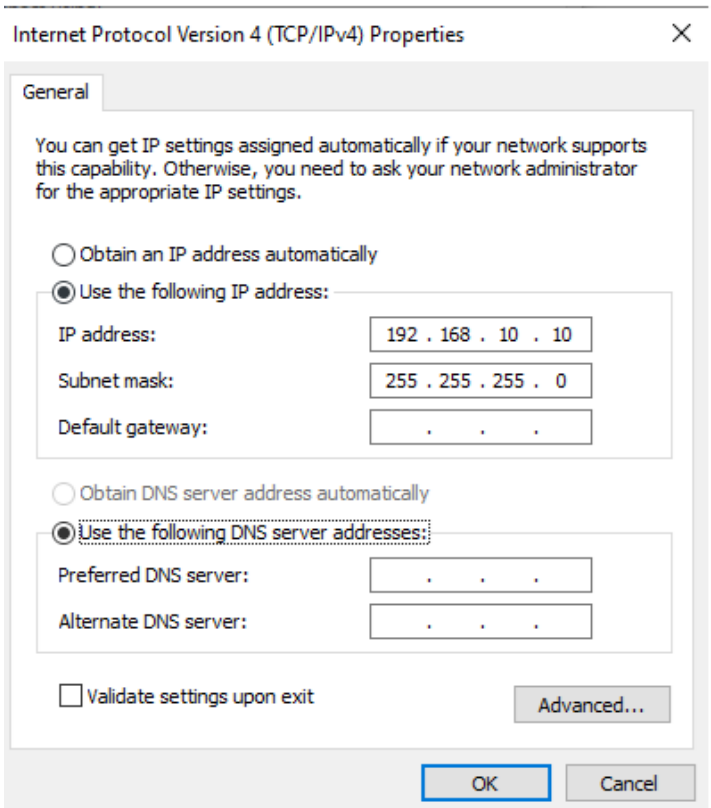
3. **Select Properties**
    - Select “Properties” from the drop-down menu.



4. **Select Internet Protocol Version 4 (TCP/IPv4)**
    - Double-click on “Internet Protocol Version 4 (TCP/IPv4)”.



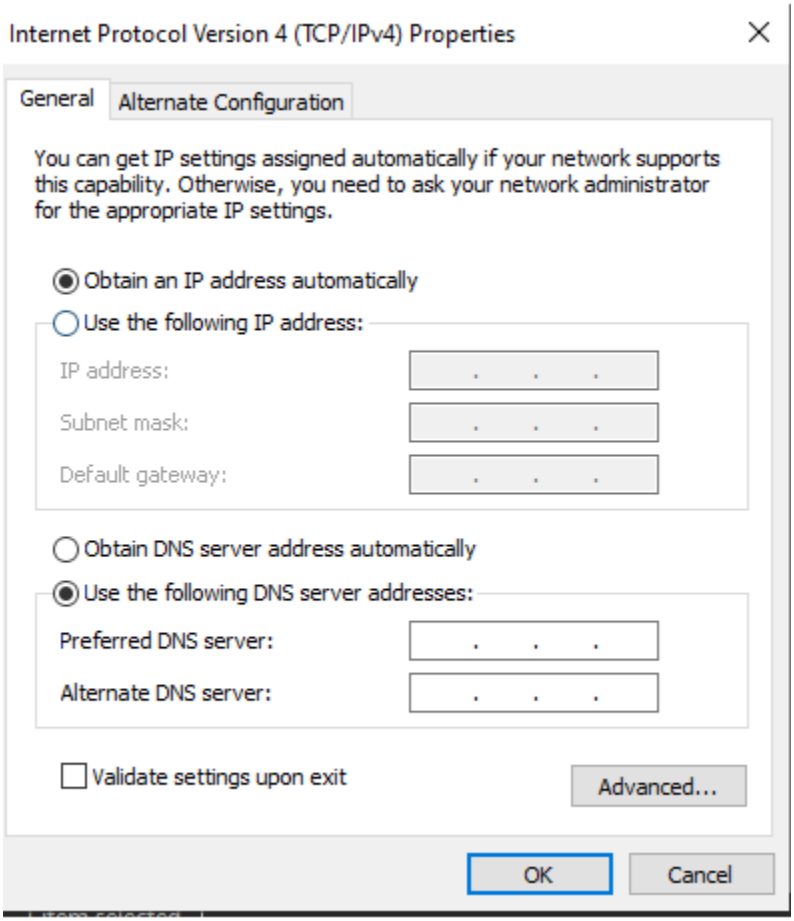
5. **Manually enter IP address and subnet mask**
    - Select “Use the following IP Address” and then input the following information in the corresponding fields:
    - IP address: Check the device that you are connected to in order to locate the IP address. The first three sets of digits should match. For this tutorial, we will use IP address 192.168.10.10.
    - Subnet mask: The subnet mask between the device that you are trying to connect to needs to be the same as your PC. For this tutorial, we will use subnet mask 255.255.255.0



6. **Save Settings**
    - Click the OK button on “Internet Protocol Version 4 (TCP/IPv4) Properties” window, and also click the OK button on “Ethernet Properties” window.
    - Note: The OK buttons must be clicked in both instances or your settings will not be saved.
7. **Revert Back to DHCP**
    - To set your computer back to DHCP, repeat steps 1-4 again. When you get to the “Internet Protocol Version 4 (TCP/IPv4) Properties” window, click “Obtain an IP

address automatically". This will allow your PC to be assigned a random IP address on your network.

- Note: The OK buttons must be clicked in both instances or your settings will not be saved.



PROGRAM:

Server:

```
import java.io.*;import java.net.*;
publicclass PingServer
{
public static void main(String a[]) throws IOException
{
Stringline1,l
ine2; int i;
System.out.println("PingS
erver"); try
{
ServerSocketsersoc=newServerSocket(9
999); Socket soc = sersoc.accept();
BufferedReadersocIn=newBufferedReader(new
InputStreamReader(soc.getInputStream()));
PrintStreamsocOut=newPrintStream(soc.getOutputStream(
)); for(i = 0; i < 4; i++)
{
line1 = socIn.readLine();
System.out.println("Pingedbyclient");
socOut.println(line1 + " reply from host:bytes=3<time<1ms TT<=128");
}
}
catch(Exception e)
{
System.out.println("Error: " + e);
}
}
}
```

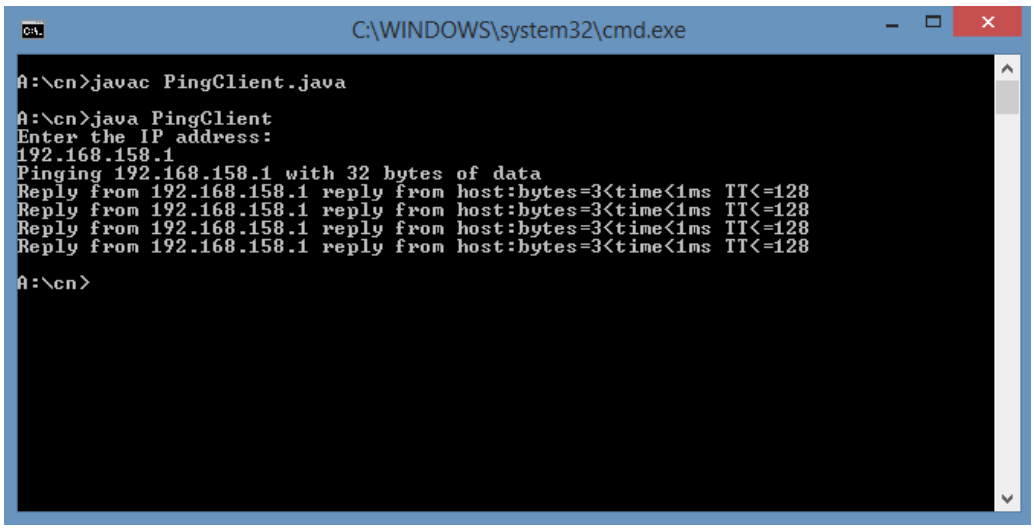
Client:

```
importjava.lang.
System; import
java.io.*;
import
java.net.*;
publicclassPin
gClient
{public static void main(String args[])
```

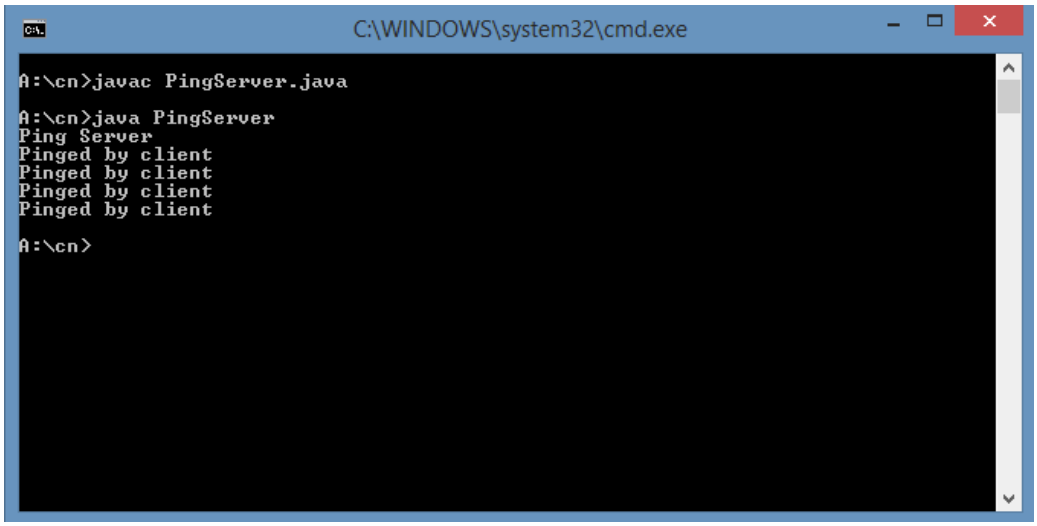
```
{
int i,J;
Stringre
moteIP;
try
{
BufferedReaderkeyIn=newBufferedReader(newInputStreamReader(System.in));
System.out.println("Enter the IP address: ");
String ip =
keyIn.readLine();
Socketsoc=newSocket(ip,9
999);
BufferedReadersocIn=newBufferedReader(new
InputStreamReader(soc.getInputStream()));
PrintStreamsocOut=newPrintStream(soc.getOutputStrea
m()); System.out.println("Pinging " + ip + " with 32
bytes of data"); for (i = 0; i < 4; i++)
{
socOut.println(ip);
remoteIP=socIn.read
Line(); if (remoteIP
!= null)
{
Thread.sleep(2000);System.out.println
("Replyfrom"+remoteIP);
}
else
{
Thread.sleep(2000);
System.out.println("Requesttimeout");
}
}
}
catch (IOException e)
{
System.out.println("Request timed out");
}
catch(InterruptedException e)
{
System.out.println("Request timed out");

}
```

OUTPUT:

Client:



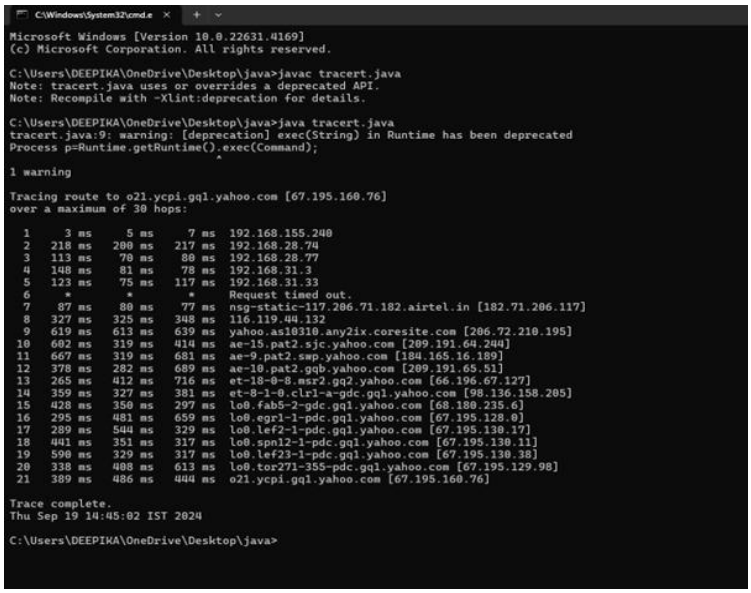
Server:



PROGRAM:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
public class tracert
{
public static void runSystemCommand(String Command)
{
try{
Process p=Runtime.getRuntime().exec(Command);
BufferedReader InputStream=new BufferedReader(new InputStreamReader(p.getInputStream()));
String s=" ";
while((s=InputStream.readLine())!=null)
{
System.out.println(s);
}
}
catch(IOException e)
{
e.printStackTrace();
}
}
public static void main(String[]args)
{
String Ip=" 67.195.160.76";
runSystemCommand("tracert" +Ip);
java.util.Date date=new java.util.Date();
System.out.println(date);
}
}
```

OUTPUT:


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.
C:\Users\DEEPIKA\OneDrive\Desktop\java>javac tracert.java
Note: tracert.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Users\DEEPIKA\OneDrive\Desktop\java>java tracert.java
tracert.java:9: warning: [deprecation] exec(String) in Runtime has been deprecated
Process p=Runtime.getRuntime().exec(Command);
^
1 warning
Tracing route to o21.ycpi.gq1.yahoo.com [67.195.160.76]
over a maximum of 30 hops:
1 3 ms 5 ms 7 ms 192.168.155.240
2 218 ms 200 ms 217 ms 192.168.28.74
3 113 ms 70 ms 80 ms 192.168.28.77
4 148 ms 81 ms 78 ms 192.168.31.3
5 123 ms 75 ms 117 ms 192.168.31.33
6 * * * Request timed out.
7 87 ms 80 ms 77 ms nsg-static-117.206.71.182.airtel.in [182.71.206.117]
8 327 ms 325 ms 348 ms 116.119.44.132
9 619 ms 613 ms 639 ms yahoo.as10310.any2ix.coresite.com [206.72.210.195]
10 602 ms 319 ms 414 ms ae-15.pat2.sjc.yahoo.com [209.191.64.244]
11 667 ms 319 ms 681 ms ae-9.pat2.smp.yahoo.com [184.165.16.189]
12 378 ms 282 ms 689 ms ae-10.pat2.gqb.yahoo.com [209.191.65.51]
13 265 ms 412 ms 716 ms et-18-0-8.msr2.gq2.yahoo.com [66.196.67.127]
14 359 ms 327 ms 381 ms et-8-1-0.clr1-a-gdc.gq1.yahoo.com [98.136.158.205]
15 428 ms 350 ms 297 ms lo0.fab5-2-gdc.gq1.yahoo.com [68.180.235.6]
16 295 ms 481 ms 659 ms lo0.egr1-1-pdc.gq1.yahoo.com [67.195.128.0]
17 289 ms 544 ms 329 ms lo0.lef2-1-pdc.gq1.yahoo.com [67.195.130.17]
18 441 ms 351 ms 317 ms lo0.spn12-1-pdc.gq1.yahoo.com [67.195.130.11]
19 590 ms 329 ms 317 ms lo0.lef23-1-pdc.gq1.yahoo.com [67.195.130.38]
20 338 ms 408 ms 613 ms lo0.tor271-355-pdc.gq1.yahoo.com [67.195.129.98]
21 389 ms 486 ms 444 ms o21.ycpi.gq1.yahoo.com [67.195.160.76]
Trace complete.
Thu Sep 19 14:45:02 IST 2024
C:\Users\DEEPIKA\OneDrive\Desktop\java>

PROCEDURE:

1. **Setting Up a Shared Folder on the Remote Computer**
   - On the remote computer (e.g., RemotePC), create a shared folder.
     - Right-click on a folder > Properties > Sharing > Share this folder.
     - Set permissions to allow access (at least Read or Read/Write access depending on your requirements).
     - Note down the path of the shared folder (e.g., \\RemotePC\SharedFolder).
2. Using the net use Command
   - Command Structure:

net use <DriveLetter>: \\<RemotePC>\<SharedFolder> /user:<username> <password>

Here:

- <DriveLetter> is the drive letter you want to assign to the mounted volume (e.g., Z:).
- <RemotePC> is the hostname or IP address of the remote computer.
- <SharedFolder> is the name of the shared folder on the remote machine.
- /user:<username> is used to specify a user account if authentication is needed.
- <password> is the password for that account.

3. Mounting the Volume (Lab Exercise)

Step 1:

- Open **Command Prompt** on the local machine as an Administrator.

Step 2:

- Enter the following command:

net use Z: \\192.168.1.10\SharedFolder /user:admin password123

- Replace 192.168.1.10 with the IP address of your remote machine.
- Replace SharedFolder with the folder name you've shared.
- Replace admin and password123 with appropriate login credentials.

Step 3:

- If the command is successful, you will see a message like

The command completed successfully.

4. Verifying the Mount

- Once the volume is mounted, it will appear as a local drive on the local machine.
    - Open **File Explorer** and check under **This PC**. The Z: drive (or whichever letter you chose) should appear, and you can access files in the remote share.

5. Disconnecting the Volume

- To unmount the mapped drive, you can use:

net use Z: /delete

- This will remove the mapping for the Z: drive.

PROCEDURE:

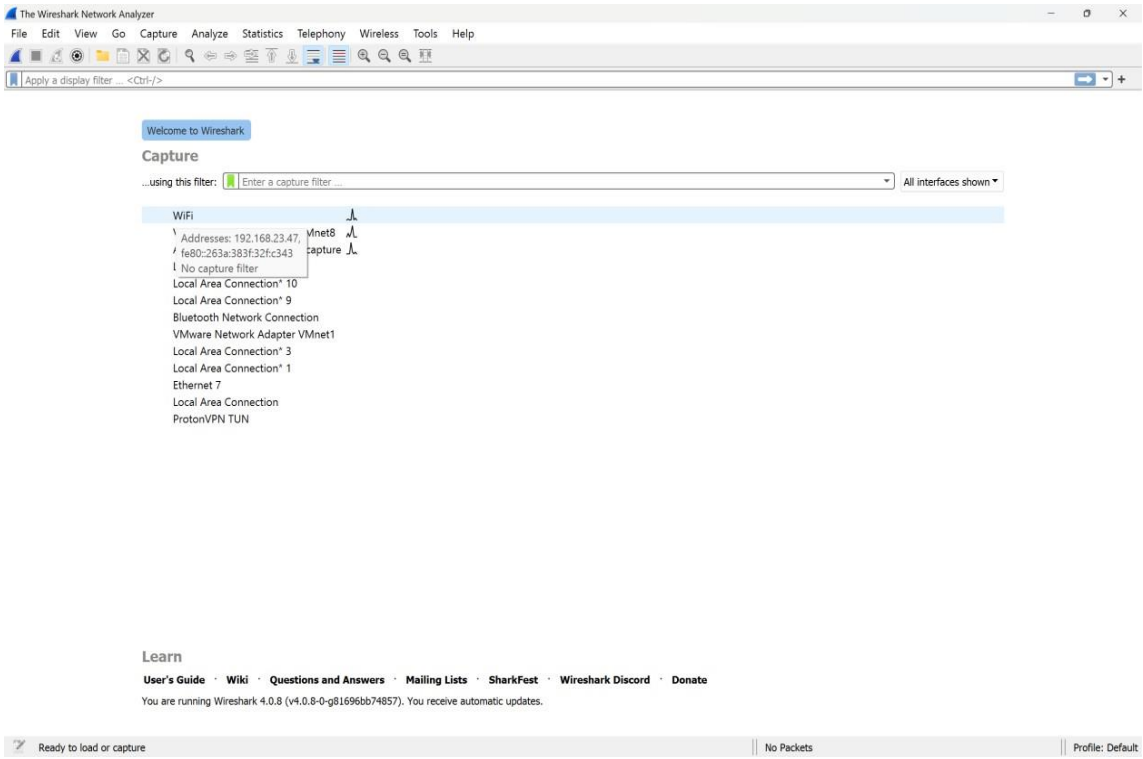
Step 1: Prepare for Packet Capture

1. **Open Wireshark:**

- Launch Wireshark on your computer.

2. **Select the Network Interface:**

- Choose the network interface that you want to monitor (e.g., Ethernet, Wi-Fi).



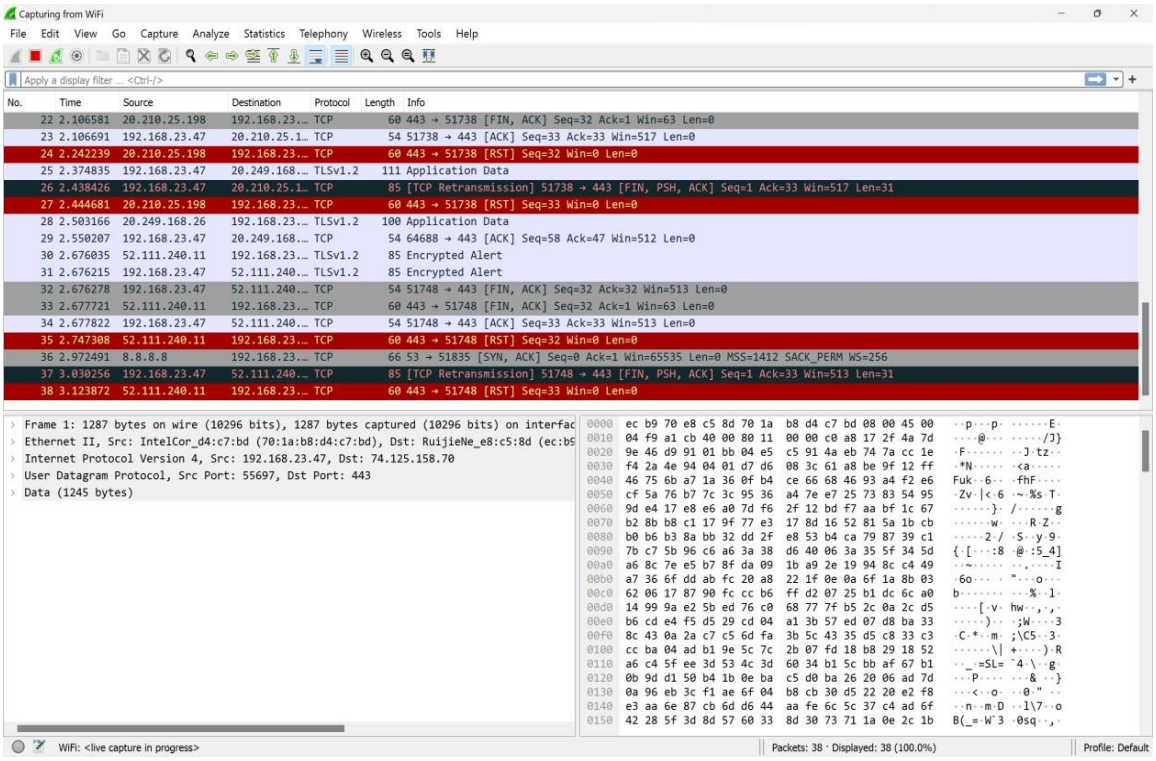
Step 2: Start Packet Capturing

1. **Begin Capturing**:

- Click the blue shark fin icon at the top left or press Ctrl + E to start capturingpackets.

2. **Monitor the Capture:**

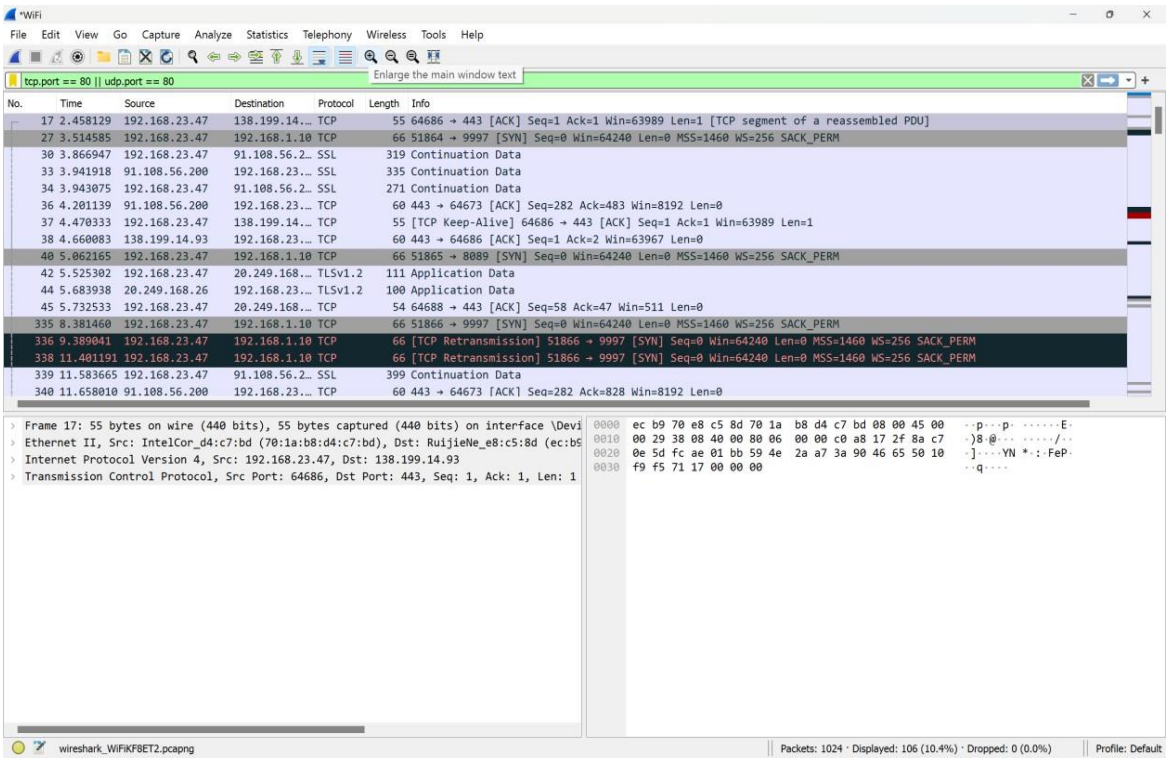
- Watch the live packet data being captured and displayed in real-time.



Step 3: Apply Capture Filters

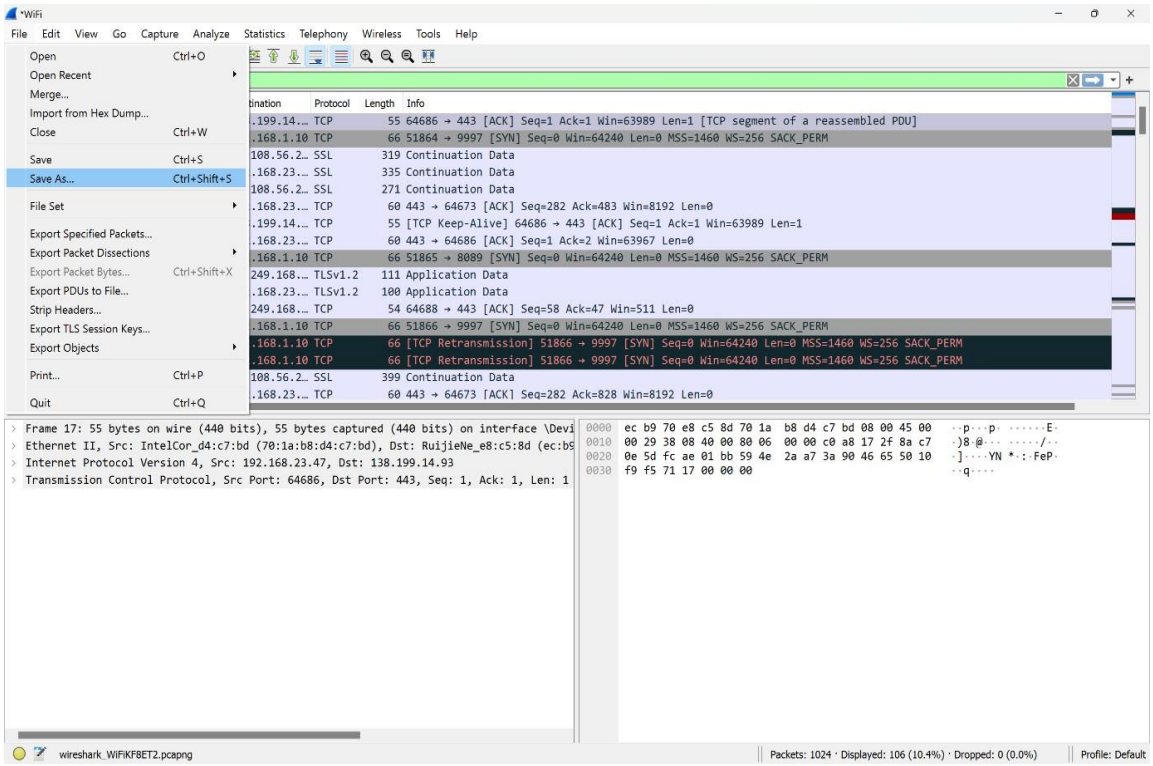
1. **Set Filters**:

- Before capturing, enter a capture filter (e.g., tcp port 80) to limit the data tospecific traffic types.



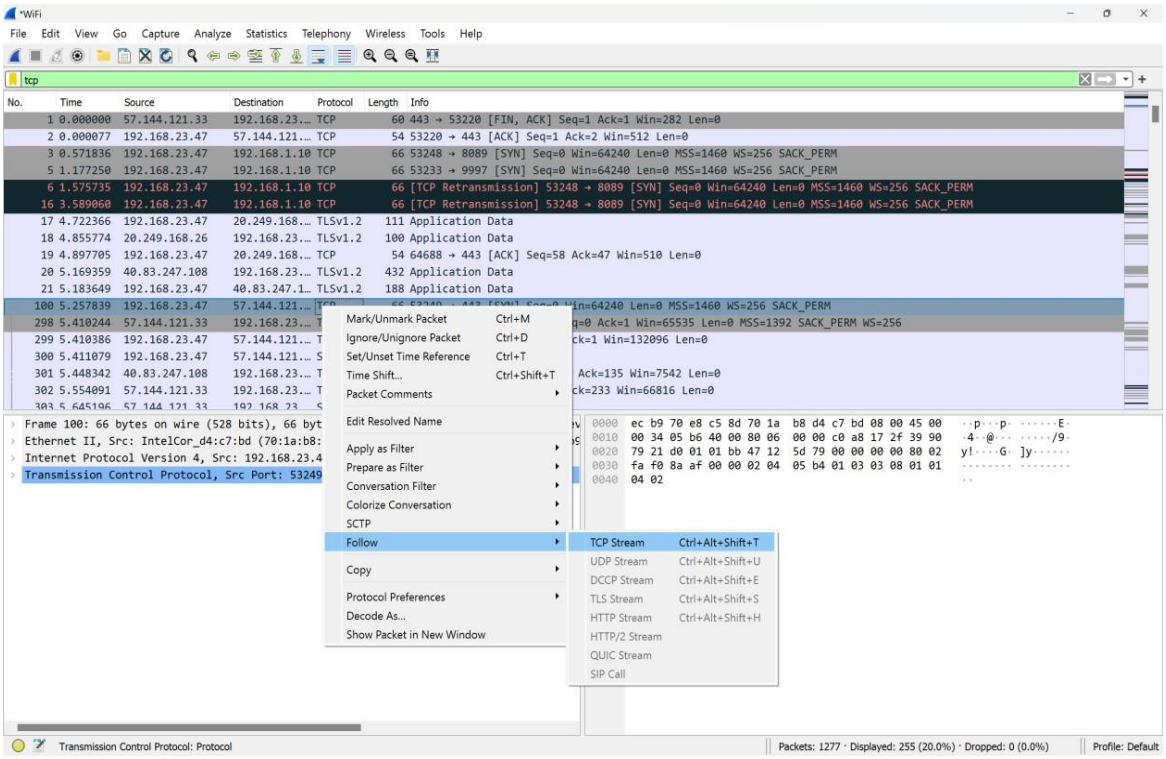
Step 4: Stop Packet Capturing

1. **Stop Capture**:
   - Click the red square icon or press Ctrl + E again to stop capturing.
2. **Save the Capture:**
   - Go to File > Save As, choose a file location, and save the captured data.



Step 5: Analyze Captured Packets

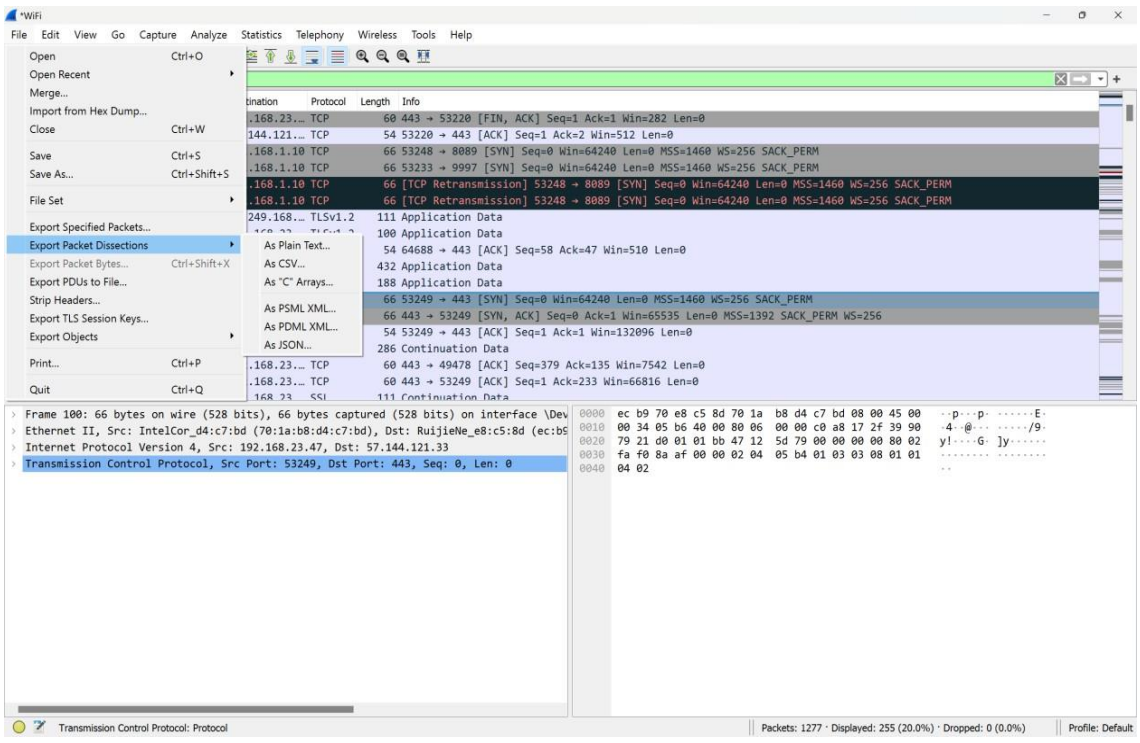
1. **Inspect Packet Details**:
   - Expand the protocol layers in the Packet Details pane to analyze specific fieldsand values.
2. **Follow a Stream:**
   - Right-click a packet and select "Follow" > "TCP Stream" to view a detailed conversation.



Step 6: Export and Document Findings

1.Export Data:

Go to File > Export Packet Dissections to save specific packets or data in adifferent format.



.**PROGRAM:**

```
import java.net.*;
import java.io.*;
public class ChatServer
{
public static void main(String args[])
{
try
{
String line;
ServerSocket ss=new ServerSocket(9000); Socket soc=ss.accept();
BufferedReader socIn = new BufferedReader(new
InputStreamReader(soc.getInputStream()));
PrintStream socOut = new PrintStream(soc.getOutputStream()); while(true)
{
line = socIn.readLine(); socOut.println(line);
}
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

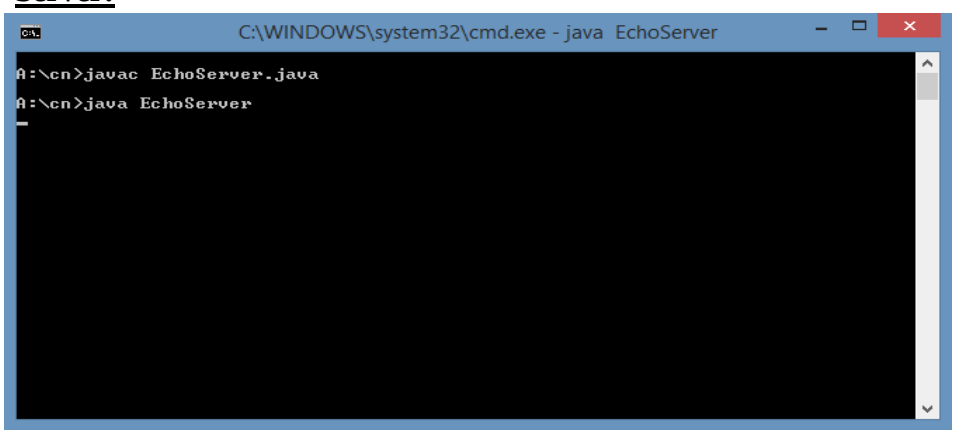
Client:

```
Importjav
a.net.*im
port
java.io.*;
public class ChatClient
{
public static void main(String arg[])
{
try
{
String line;
InetAddressip=InetAddress.getLocalHos
t(); Socket soc = new Socket(ip, 9000);
BufferedReadersocIn=newBufferedReader(newInputStreamReader(soc.getInputStream()))
; PrintStream socOut = new PrintStream(soc.getOutputStream());
BufferedReaderkeyIn=newBufferedReader(newInputStreamReader(System
.in)); while(true)
```

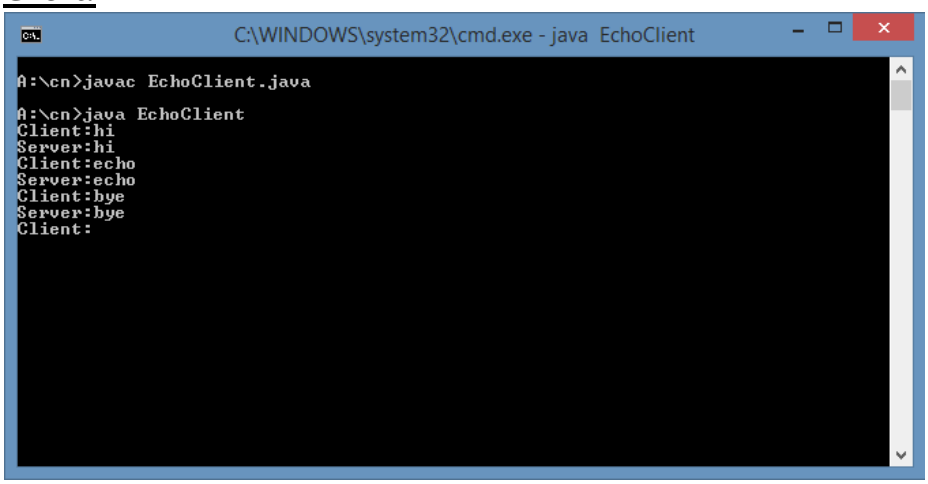
```
System.out.print("Client:");line=keyIn.readLine();socOut.println(line);
System.out.println("Server:" + socIn.readLine());
}
}
catch(IOException e)
{
System.out.println(e);
}
}
}
```

OUTPUT:

Server:



Client:



PROGRAM:

MyServer.java

```
import
java.net.*
; import
java.io.*;
class
MyClient
{
public static void main(String args[])throws IOException
{
Socket s=new Socket("localhost",3333);
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new
DataOutputStream(s.getOutputStream());
BufferedReaderbr=newBufferedReader(newInputStreamReader(Syste
m.in)); String str="",str2="";
while(!str.equals("stop"))
{
str=br.readLi
ne();
dout.writeU
TF(str);
dout.flush();
str2=din.read
UTF();
System.out.println("Server says: "+str2);
}
dout.close();
s.close();
}
}
```

MyClient

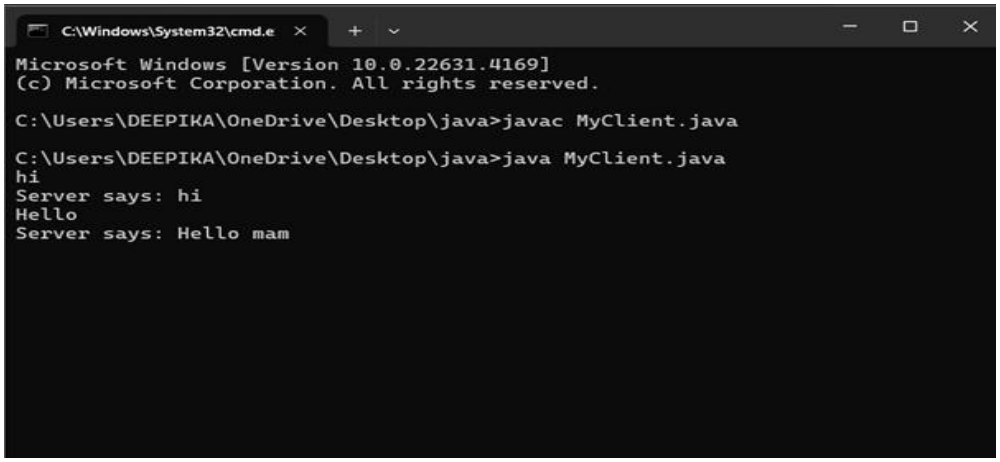
.java

```
import
java.net.*
; import
java.io.*;
class
MyServer
{
public static void main(String args[])throws IOException
```

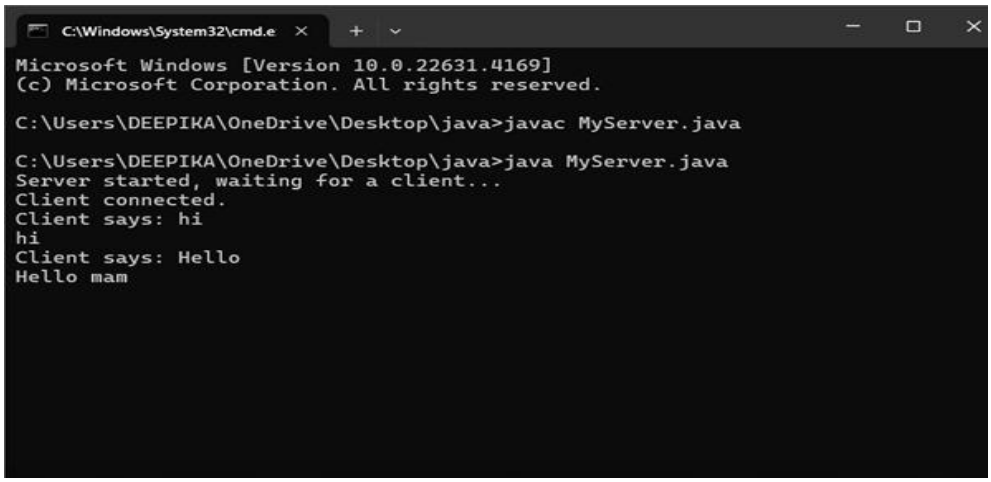
```
{
ServerSocketss=newServerSocket(3333)
; Socket s=ss.accept();
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new
DataOutputStream(s.getOutputStream());
BufferedReaderbr=newBufferedReader(newInputStreamReader(Syste
m.in)); String str="",str2="";
while(!str.equals("stop"))
{
str=din.readUTF();
System.out.println("clientsays:"
+str);

str2=br.readLine(
);
dout.writeUTF(st
r2); dout.flush();
}
din.close();
s.close();
ss.close();
}
```

OUTPUT:

Server:



Client:



PROGRAM:

FileClient.java

```
import java.io.*;
import java.net.*;
public class FileClient
{
public static void main(String args[])throws IOException
{
try
{
Socket soc = new Socket(InetAddress.getLocalHost(),1187);
BufferedReader socIn = new BufferedReader(new
InputStreamReader(soc.getInputStream()));
String input;
while((input = socIn.readLine()) != null)
{
System.out.println(input);
}
System.out.println("The file is received successfully");
}
catch(Exception e)
{
System.out.println("Error: " + e);
}
}
}
```

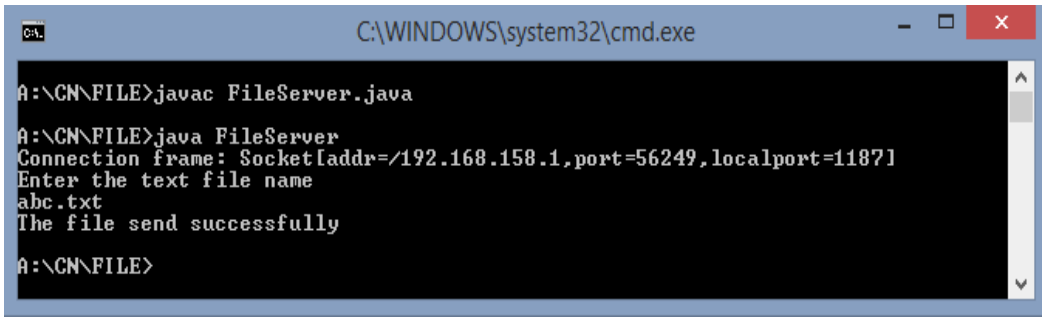
FileServer.java

```
import java.io.*;
import java.net.*;
public class FileServer
{
public static void main(String args[])throws IOException
{
try
{
ServerSocket sersoc = new ServerSocket(1187); Socket soc = sersoc.accept();
System.out.println("Connection frame: " + soc);
PrintStream socOut = new PrintStream(soc.getOutputStream());
BufferedReader keyIn = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the text file name");
String fileName = keyIn.readLine(); File f = new File(fileName);
if(f.exists())
{
BufferedReader fileIn = new BufferedReader(new FileReader(fileName));
String line;
while((line = fileIn.readLine()) != null)
{
```

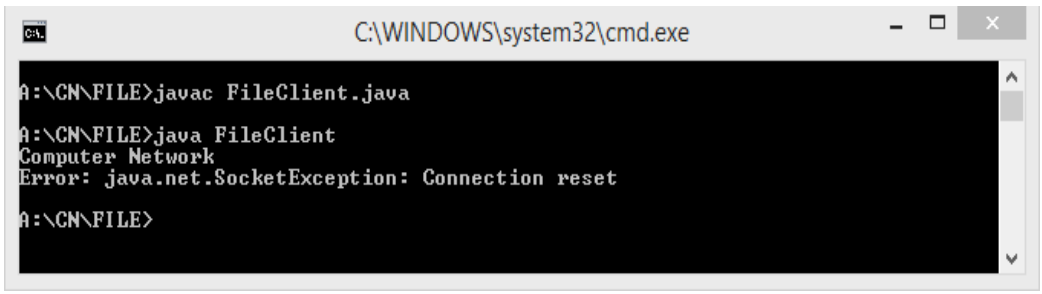
```
socOut.println(line);
}
System.out.println("The file send successfully");
}
else
{
System.out.println("File not exists");
}
}
catch(IOException e)
{
System.out.println("Error: " + e);
}
}
}
```

OUTPUT:

Server:



Client:

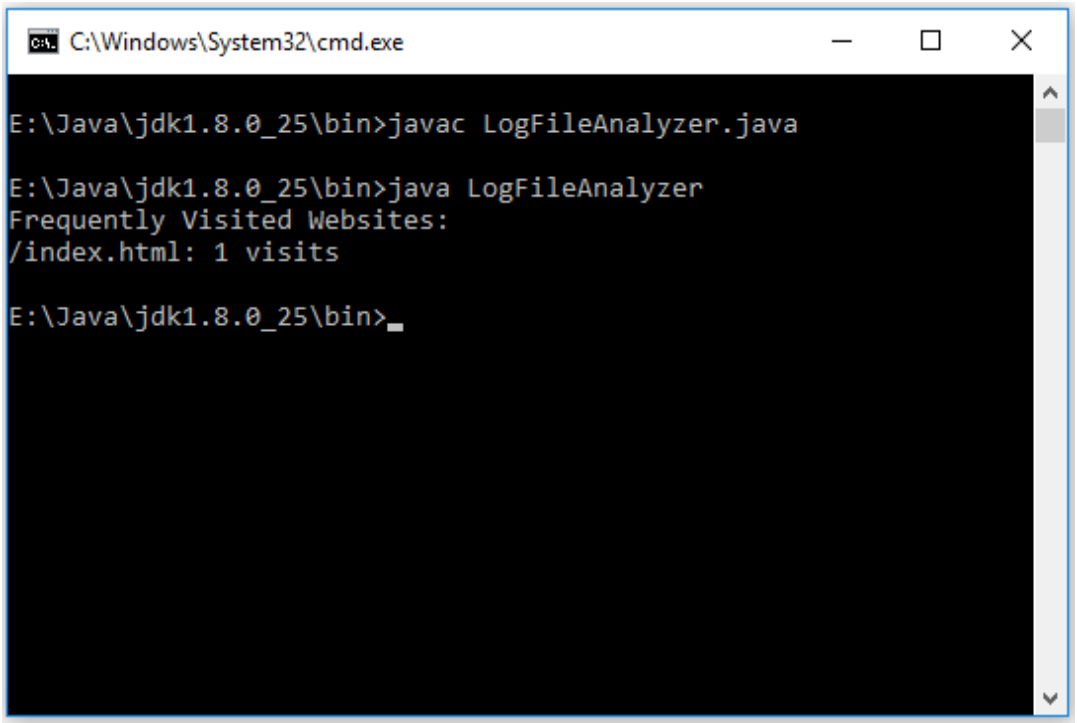


PROGRAM:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;

public class LogFileAnalyzer {

public static void main(String[] args) {
String filePath = "access.log"; // Specify the path to your log file
Map<String, Integer> urlCount = new HashMap<>();

try {
BufferedReader reader = new BufferedReader(new FileReader(filePath));
String line;

while ((line = reader.readLine()) != null) {
String url = extractUrl(line);
if (url != null) {
urlCount.put(url, urlCount.getOrDefault(url, 0) + 1);
}
}
reader.close();
} catch (IOException e) {
System.err.println("Error reading the log file: " + e.getMessage());
}

System.out.println("Frequently Visited Websites:");
urlCount.entrySet().stream()
.sorted((entry1, entry2) -> entry2.getValue().compareTo(entry1.getValue()))
```

```
.forEach(entry -> System.out.println(entry.getKey() + ": " + entry.getValue() + " visits"));
}

private static String extractUrl(String logLine) {
// Regex pattern to match the requested URL from the log line
String regex = "\"GET (.+?) HTTP/1\\.1\"";
java.util.regex.Pattern pattern = java.util.regex.Pattern.compile(regex);
java.util.regex.Matcher matcher = pattern.matcher(logLine);

if (matcher.find()) {
return matcher.group(1);
}
return null;
}
}
```

OUTPUT:

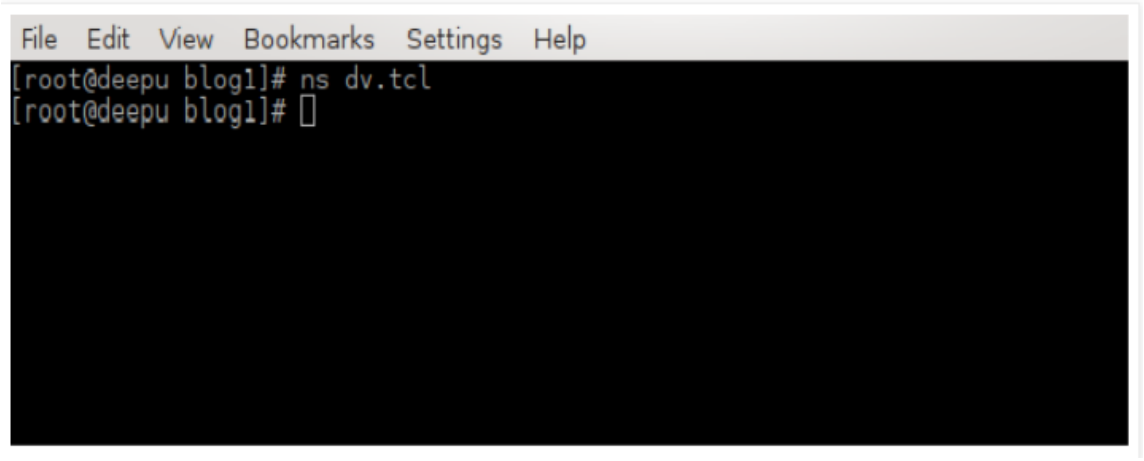


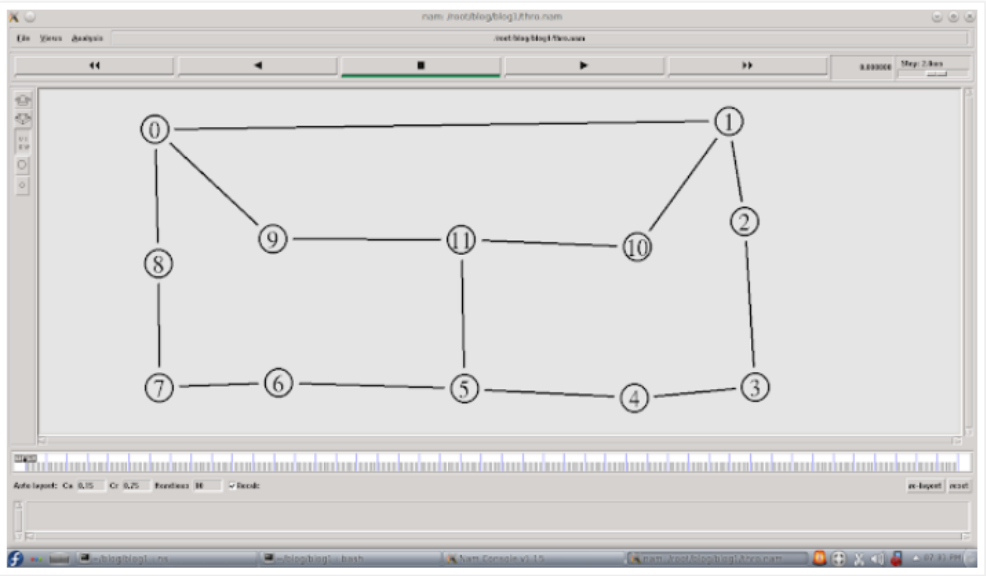
PROGRAM:

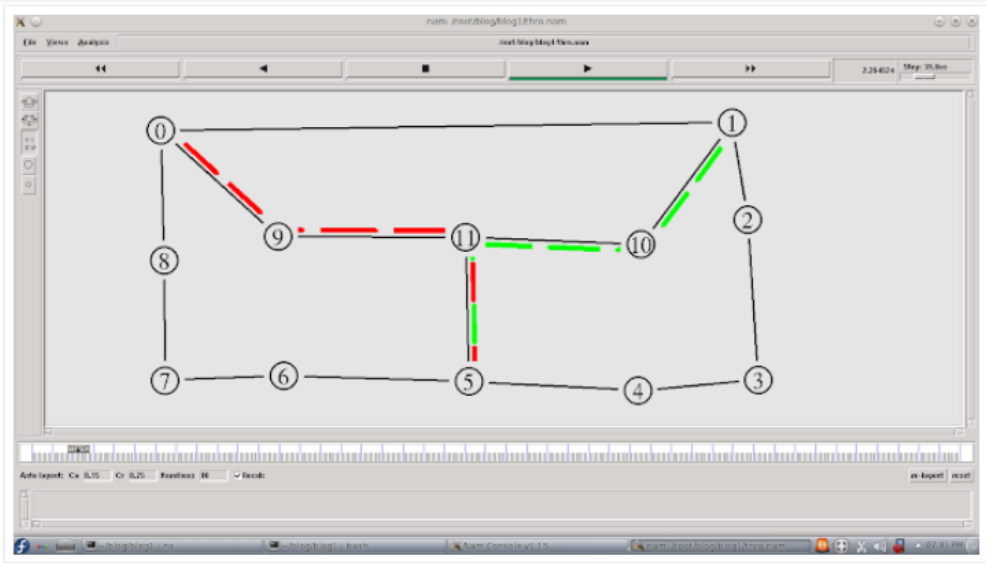
```
set ns [new Simulator]
set nr [open thro.tr w]
$ns trace-all $nr
set nf [open thro.nam w]

$ns namtrace-all $nf
      proc finish { } {
      global ns nr nf
      $ns flush-trace
      close $nf
      close $nr
      exec nam thro.nam &
            exit 0
      }

for { set i 0 } { $i < 12} { incr i 1 } {
set n($i) [$ns node]}

for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set  interval_ 0.005
```

```
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set  interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0
$ns rtproto DV
$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)
$udp0 set fid_ 1
$udp1 set fid_ 2
$ns color 1 Red
$ns color 2 Green
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"
$ns at 45 "finish"
$ns run
```

OUTPUT:


File Edit View Bookmarks Settings Help
[root@deepu blog1]# ns dv.tcl
[root@deepu blog1]#


0
1
2
3
4
5
6
7
8
9
10
11

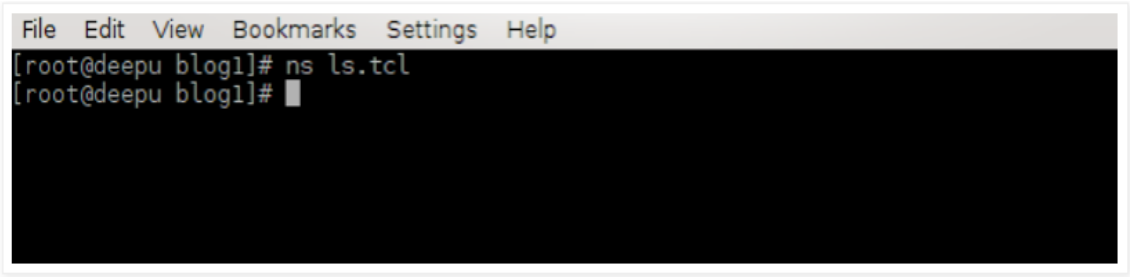

0
1
2
3
4
5
6
7
8
9
10
11

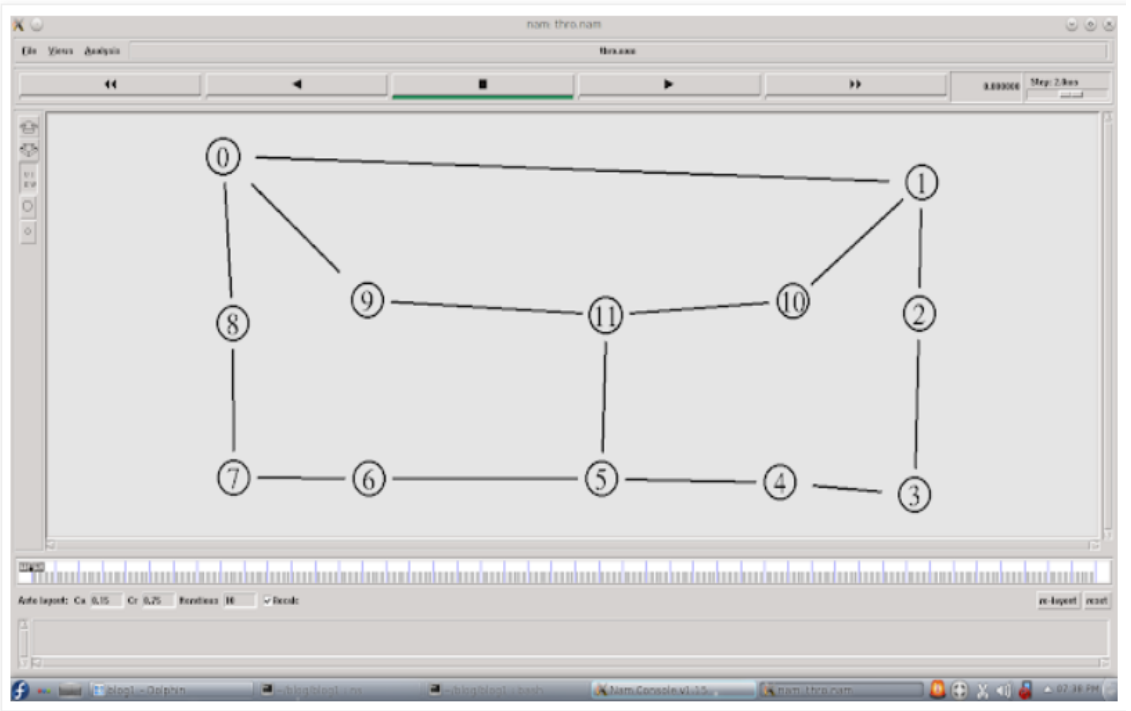
PROGRAM:

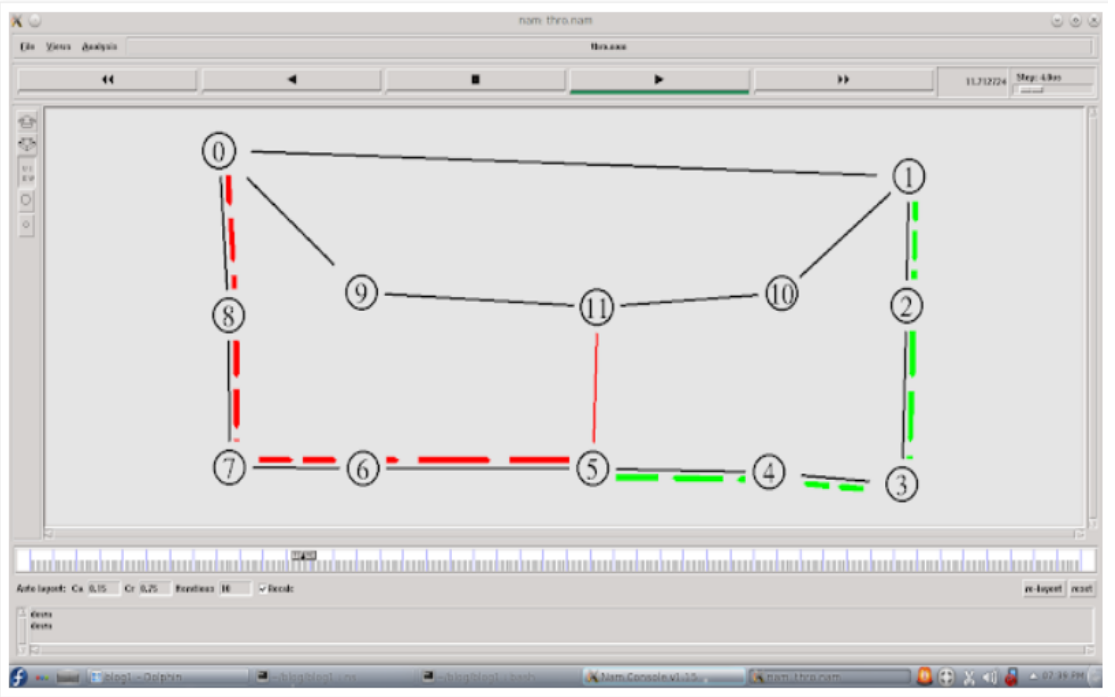
```
set ns [new Simulator]
set nr [open thro.tr w]
$ns trace-all $nr
set nf [open thro.nam w]

$ns namtrace-all $nf
        proc finish { } {
        global ns nr nf
        $ns flush-trace
        close $nf
        close $nr
        exec nam thro.nam &
                exit 0
        }

for { set i 0 } { $i < 12} { incr i 1 } {
set n($i) [$ns node]}

for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail  }

$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 set packetSize_ 500
$cbr0 set  interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0

set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set  interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0

$ns rtproto LS

$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)

$udp0 set fid_ 1
$udp1 set fid_ 2
$ns color 1 Red
$ns color 2 Green
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"
$ns at 45 "finish"
$ns run
```

OUTPUT:


File Edit View Bookmarks Settings Help
[root@deepu blog1]# ns ls.tcl
[root@deepu blog1]#


0
1
2
3
4
5
6
7
8
9
10
11


0
1
2
3
4
5
6
7
8
9
10
11