Institute of Distance and Open Learning

Vidya Nagari, Kalina, Santacruz East – 400098.

CERTIFICATE

This is to certify that **Mr. Omkar B Auti** of **Master in Computer Application** (MCA) Semester II has completed the specified term work in the subject of **Networking with Linux** satisfactorily within this institute as laid down by University of Mumbai during the academic year 2023 to 2024.

Subject In-charge	External Examiner	Coordinator – M.C.A

INDEX

PRACTICAL 1: INSTALLING NS-3 IN UBUNTU.

Steps for installing NS-3 in Ubuntu:

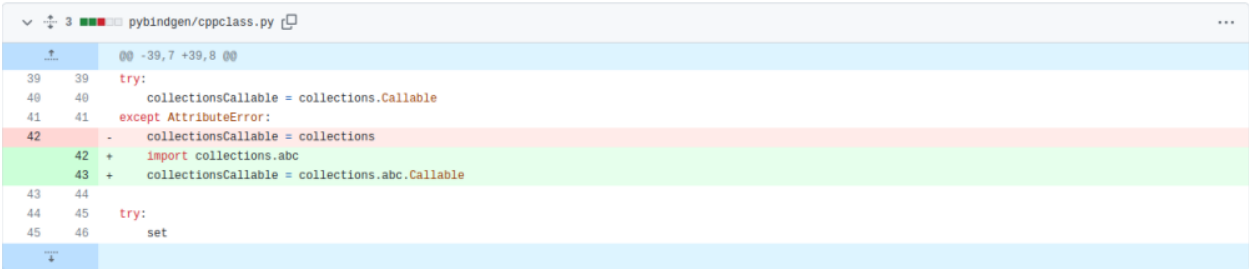
Step 1: Update the system
$ sudo apt update

Step 2: Prerequisites for installing NS-3
$ sudo apt install build-essential autoconf automake libxmu-dev g++ python3 python3-dev pkg-config sqlite3 cmake python3-setuptools git qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools gir1.2-goocanvas-2.0 python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython3 openmpi-binopenmpi-common openmpi-doc libopenmpi-dev autoconf cvs bzr unrar gsl-bin libgsl-dev libgslcblas0wireshark tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev libc6-dev libc6-dev-i386 libclangdev llvm-dev automake python3-pip libxml2 libxml2-dev libboost-all-dev

Now download the ns3 3.35 from https://nsnam.org
Copy the softwares from the Downloads/ folder to the home folder (in my case its /home/ns-3/)
Now extract both the versions using the GUI method.
Just right click and click "Extract Here"
Now we will install ns-3.35

$ cd
$ cd ns-allinone-3.35/
$ ./build.py --enable-examples --enable-tests

In case, if you get the following error pybindgen(ns3 module antenna)
Do this step and repeat the above step



```
3 pybindgen/cppclass.py
@@ -39,7 +39,8 @@
 try:
     collectionsCallable = collections.Callable
except AttributeError:
-    collectionsCallable = collections
+    import collections.abc
+    collectionsCallable = collections.abc.Callable

try:
    set
```

We have installed two version of ns-3.35 successfully in Ubuntu

PRACTICAL 2: INSTALL NETANIM IN UBUNTU.

Steps to Install NetAnim:

You can directly install NetAnim

Otherwise, you have to execute some commands but for this we need NS3installed or compiled.

Step1: sudo apt-get install NetAnim

Step2: NetAnim file.xml

Step3: Select Xml File

Step4: Run the simulation by clicking, NS3 NetAnim successfully.

PRACTICAL 3: INSTALL WIRESHARK IN UBUNTU.

Install Wireshark:

Step 1: Add the stable official PPA. To do this, go to terminal by pressing Ctrl+Alt+T and run:
sudo add-apt-repository ppa:wireshark-dev/stable

Step 2: Update the repository:
sudo apt-get update

Step 3: Install wireshark 2.0:
sudo apt-get install wireshark

Step 4: Run wireshark:
sudo wireshark

If you get an error couldn't run /usr/bin/dumpcap in child process: Permission Denied. go to the terminal again and run:

sudo dpkg-reconfigure wireshark-common

Say YES to the message box. This adds a wireshark group. Then add user to the group by typing

sudo adduser $USER wireshark

PRACTICAL 4: ANALYZE THE NETWORK TRAFFIC USING WIRE SHARK.

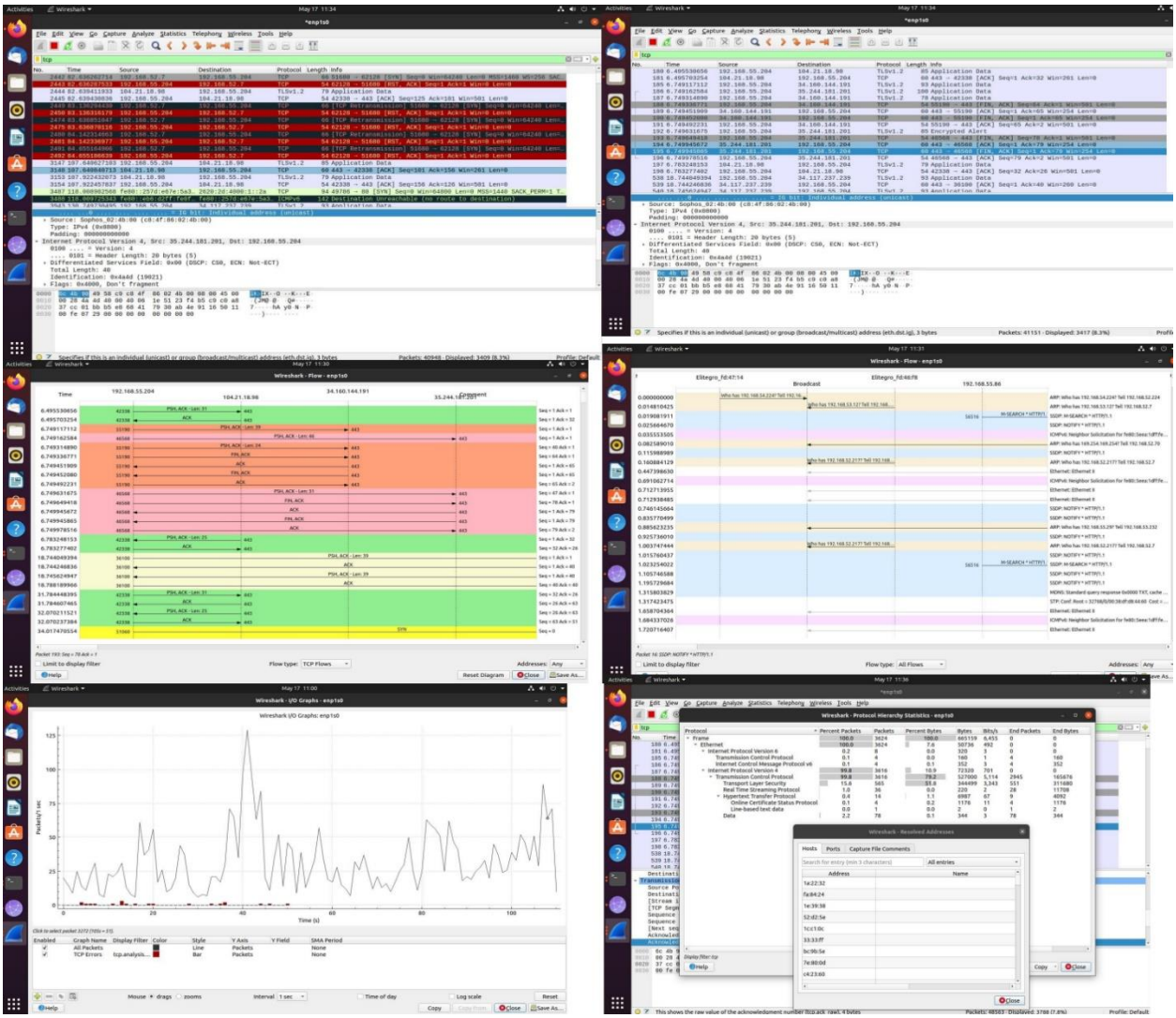
Code:

```
#include "ns3/core-module.h"
#include"ns3/network-module.h"
#include"ns3/netanim-module.h"
#include"ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include"ns3/applications-module.h"
#include "ns3/point-to-pointlayout-module.h"#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"
// Network topology (default)
//
  //      n2 n3 n4            .
  //       \ | /              .
  //        \|/               .
  //    n1--- n0---n5          .
  //        /|\               .
  //       / | \              .
  //      n8 n7 n6            .

//
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("Star");
int main (int argc, char *argv[])
{
        NodeContainer nodes;
        nodes.Create(9);
        //
        // Set up some default values for the simulation.
        //
        Config::SetDefault("ns3::OnOffApplication::PacketSize",UintegerValue(137));
        // ??? try and stick 15kb/s into the data rate
        Config::SetDefault("ns3::OnOffApplication::DataRate", StringValue("14kb/s"));
        //
        // Default number of nodes in the star. Overridable by command lineargument.
```

```
//
uint32_t nSpokes = 8;
CommandLine cmd ( FILE );
cmd.AddValue ("nSpokes", "Number of nodes to place in the star",nSpokes);
cmd.Parse (argc, argv);
NS_LOG_INFO("Build star topology.");
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay",StringValue ("2ms"));
PointToPointStarHelper star (nSpokes, pointToPoint);
NS_LOG_INFO ("Install internet stack on all nodes.");
InternetStackHelper internet;
star.InstallStack (internet);
NS_LOG_INFO ("Assign IP Addresses.");
star.AssignIpv4Addresses (Ipv4AddressHelper ("10.1.1.0", "255.255.255.0"));
NS_LOG_INFO ("Create applications.");
//
// Create a packet sink on the star "hub" to receive packets.
//
uint16_t port = 50000;
Address hubLocalAddress (InetSocketAddress (Ipv4Address::GetAny (),port));
PacketSinkHelperpacketSinkHelper ("ns3::TcpSocketFactory",hubLocalAddress);
ApplicationContainer hubApp = packetSinkHelper.Install(star.GetHub ());
hubApp.Start (Seconds (1.0));
hubApp.Stop (Seconds (10.0));
//
// Create OnOff applications to send TCP to the hub, one on each spokenode.
//
OnOffHelper onOffHelper ("ns3::TcpSocketFactory", Address ());
onOffHelper.SetAttribute("OnTime",
StringValue("ns3::ConstantRandomVariable[Constant=1]"));
onOffHelper.SetAttribute ("OffTime",
StringValue("ns3::ConstantRandomVariable[Constant=0]"));
ApplicationContainer spokeApps;
for (uint32_t i = 0; i < star.SpokeCount (); ++i)
{
        AddressValue remoteAddress (InetSocketAddress (star.GetHubIpv4Address (i),
        port));
        onOffHelper.SetAttribute ("Remote", remoteAddress);
        spokeApps.Add(onOffHelper.Install(star.GetSpokeNode (i)));
```

```
        }
        spokeApps.Start (Seconds (1.0));
        spokeApps.Stop (Seconds (10.0));
        NS_LOG_INFO ("Enable static global routing.");
        //
        // Turn on global static routing so we can actually be routed across the star.
        //
        Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
        NS_LOG_INFO("Enable pcap tracing.");
        //
        // Do pcap tracing on all point-to-point devices on all nodes.
        //
        MobilityHelper mobility;
        mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
        mobility.Install(nodes);
        AnimationInterface anim("star.xml");
        AnimationInterface::SetConstantPosition(nodes.Get(0),10,2);
        AnimationInterface::SetConstantPosition(nodes.Get(1),11,5);
        AnimationInterface::SetConstantPosition(nodes.Get(2),15,2);
        AnimationInterface::SetConstantPosition(nodes.Get(3),19,7);
        anim.EnablePacketMetadata(true);
        pointToPoint.EnablePcapAll ("star");
        NS_LOG_INFO ("Run Simulation.");
        Simulator::Run ();
        Simulator::Destroy ();
        NS_LOG_INFO ("Done.");
        return 0;
}
```

Output:



PRACTICAL 5: PROGRAM TO SIMULATE UDP SERVER CLIENT.

Code:

Program for UDP Server:

```
package UDP;
import java.io.*;
import java.net.*;
public class UDPserver {
	public static void main(String[] args) throws IOException {
		DatagramSocket server = new DatagramSocket(3500);
		byte[] buf = new byte[256];
		DatagramPacket packet = new
		DatagramPacket(buf,buf.length);
		server.receive(packet);
		String response = new String(packet.getData());
		System.out.println("Response Data : "+response);
		server.close();
	}
}
```

Program for UDP Client:

```
package UDP;
import java.io.*;
import java.net.*;
public class UDPclient {
	public static void main(String[] args) throws IOException {
		DatagramSocket client = new DatagramSocket();
		InetAddress add = InetAddress.getByName("localhost");
		String str = "Hello everyone";
		byte[] buf = str.getBytes();
		DatagramPacket p = new
		DatagramPacket(buf,buf.length,add,3500);
		client.send(p);
	}
}
```

Simulated output:

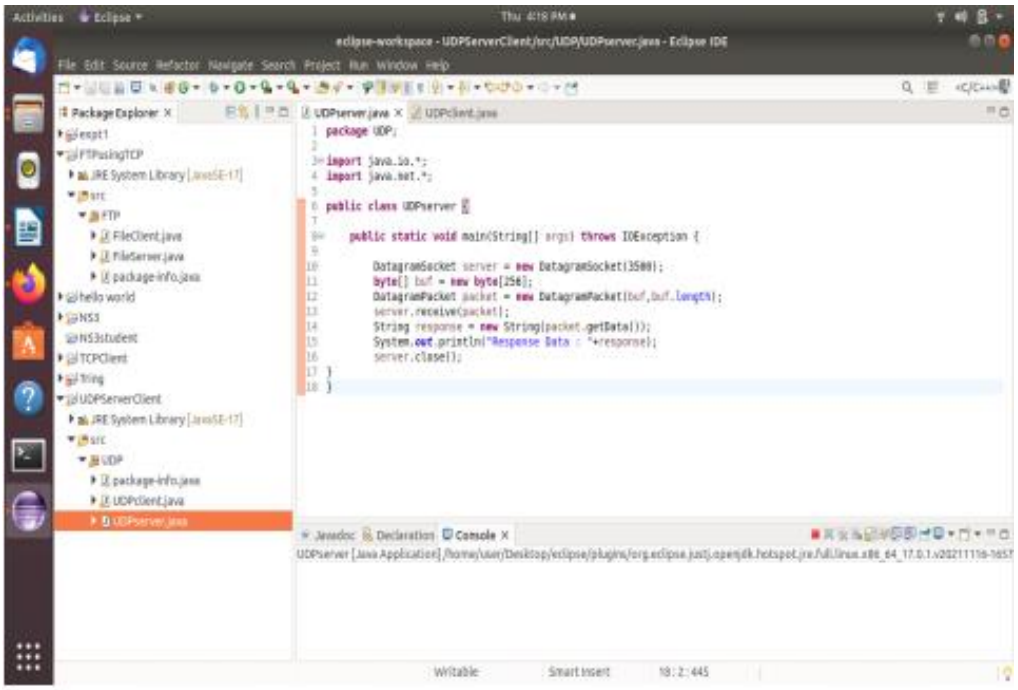


Figure 7 : Run file of UDP server.java

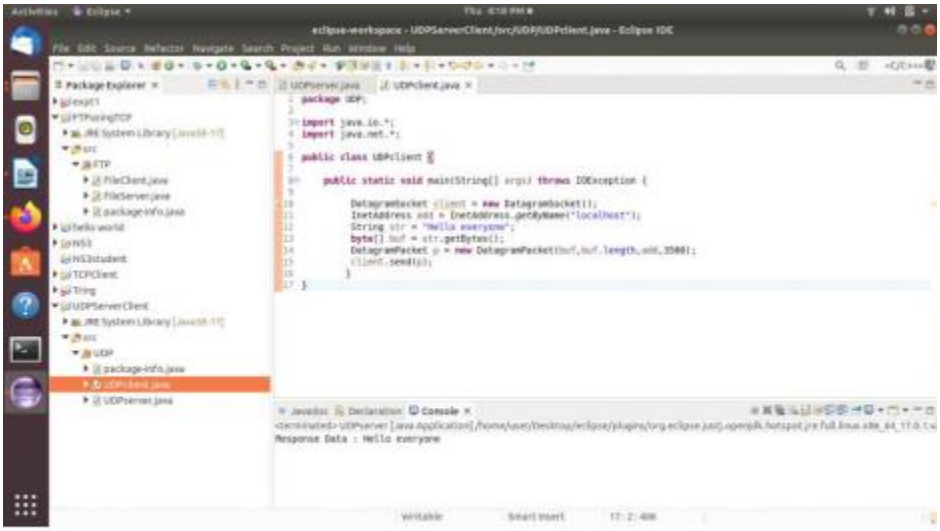


Figure 8: Run file of UDP client.java and Output shown in console window

PRACTICAL 6: PROGRAM TO SIMULATE FTP USING TCP PROTOCOL.

Code:

Program for TCP Server:

```
package FTP;
import java.io.*;
import java.net.*;
import java.util.Arrays;
public class FileServer {
	public static void main(String[] args) throws Exception {
		ServerSocket s = new ServerSocket (4002);
		Socket sr = s.accept(); //for accepting socket
		FileInputStream fr = new FileInputStream("/home/user/Desktop/Test.txt"); //finds
		file location
		byte b[]= new byte[2500]; //shows file size with any random size number
		fr.read(b,0,b.length); //start reading a file from 0th line to length of
		FileOutputStream os = sr.getOutputStream (); //Converting file to stream to send
		to client
		os.write(b, 0, b.length);
	}
}
```

Program for Client:

```
package FTP;
import java.io.*;
import java.net.*;
import java.util.Arrays;
public class FileClient {
	public static void main(String[] args) throws Exception {
		byte []b= new byte[25004];
		Socket sr = new Socket("localhost", 4002);
		InputStream is=sr.getInputStream();
		FileOutputStream fr=new
		fileOutputStream("/media/user/4668C01C49C8F823/cubic.txt");
		is.read(b,0,b.length);
```

```
                fr.write(b,0,b.length);
        }
}
```

Simulated output:

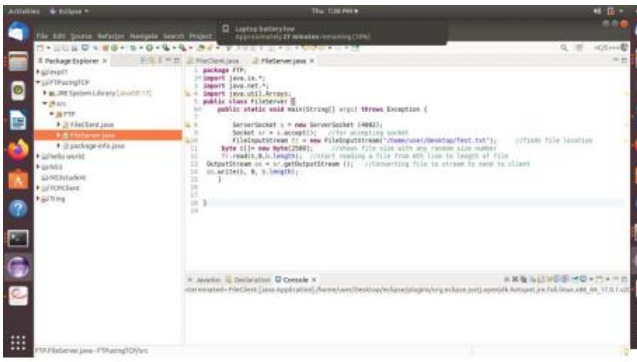


Figure 9 : Run file of TCP server.java

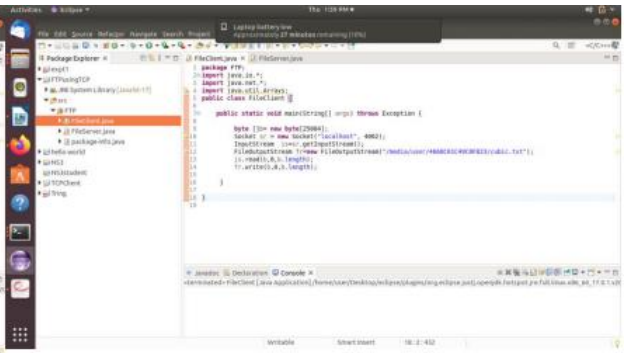


Figure 10 : Run file of TCP client.java

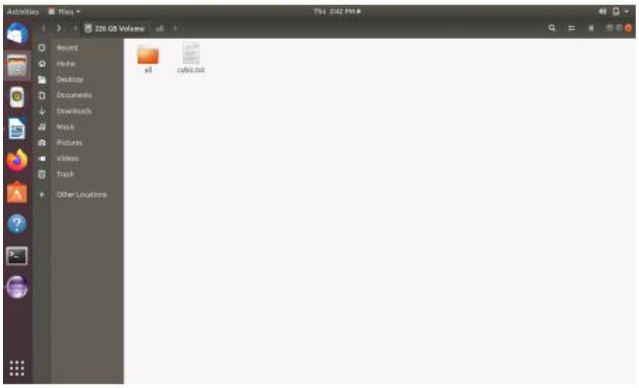


Figure 11 : Transferd file on required location with specified name

Result: The file contents are successfully transferred on new location with another name as shown in Figures 9, 10, 11 by using FTP over TCP connection.

PRACTICAL 7: WRITE A PROGRAM TO SIMULATE STAR TOPOLOGY.

Star Network topology:

```
    n2 n3 n4
       \ | /
        \|/
  n1--- n0---n5
        /|\
       / | \
    n8 n7 n6
```

Program: It is from star.cc[2]

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/netanim-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/point-to-point-layout-module.h"

using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("Star");
int main (int argc, char *argv[])
{
        //Set up some default values for the simulation.
        Config::SetDefault ("ns3::OnOffApplication::PacketSize", UintegerValue(137));
        //try and stick 15kb/s into the data rate
         Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue("14kb/s"));
        //Default number of nodes in the star. Overridable by command line argument.
        uint32_t nSpokes = 8;
        CommandLine cmd (__FILE__);
        cmd.AddValue ("nSpokes", "Number of nodes to place in the star", nSpokes);
        cmd.Parse (argc, argv);
        NS_LOG_INFO ("Build star topology.");
        PointToPointHelper pointToPoint;
        pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
```

```
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
PointToPointStarHelper star (nSpokes, pointToPoint);
NS_LOG_INFO ("Install internet stack on all nodes.");
InternetStackHelper internet;
star.InstallStack (internet);
NS_LOG_INFO ("Assign IP Addresses.");
star.AssignIpv4Addresses (Ipv4AddressHelper ("10.1.1.0", "255.255.255.0"));
NS_LOG_INFO ("Create applications.");
// Create a packet sink on the star "hub" to receive packets.
uint16_t port = 50000;
Address hubLocalAddress (InetSocketAddress (Ipv4Address::GetAny (), port));
PacketSinkHelper packetSinkHelper ("ns3::TcpSocketFactory", hubLocalAddress);
ApplicationContainer hubApp = packetSinkHelper.Install (star.GetHub());
hubApp.Start (Seconds (1.0));
hubApp.Stop (Seconds (10.0));
//Create OnOff applications to send TCP to the hub, one on each spoke node.
OnOffHelper onOffHelper ("ns3::TcpSocketFactory", Address ());
onOffHelper.SetAttribute ("OnTime",
StringValue("ns3::ConstantRandomVariable[Constant=1]"));
onOffHelper.SetAttribute ("OffTime",
StringValue("ns3::ConstantRandomVariable[Constant=0]"));
ApplicationContainer spokeApps;
for (uint32_t  i = 0;  i < star.SpokeCount (); ++i)
{
        AddressValue remoteAddress (InetSocketAddress(star.GetHubIpv4Address (i),
        port));
        onOffHelper.SetAttribute ("Remote", remoteAddress);
        spokeApps.Add (onOffHelper.Install (star.GetSpokeNode (i)));
}
spokeApps.Start (Seconds (1.0));
spokeApps.Stop (Seconds (10.0));
NS_LOG_INFO ("Enable static global routing.");
//Turn on global static routing so we can actually be routed across the star.
Ipv4GlobalRoutingHelper::PopulateRoutingTables();
NS_LOG_INFO ("Enable pcap tracing.");
//Do pcap tracing on all point-to-point devices on all nodes.
pointToPoint.EnablePcapAll("star");
NS_LOG_INFO ("Run Simulation.");
Simulator::Run();
Simulator::Destroy();
```

```
        NS_LOG_INFO("Done.");
        return 0;
}
```

Output by following commands:

1) ~/Desktop/ns-allinone-3.31/ns-3.31$ ./waf --run scratch/star
Build finished successfully

2) ~/Desktop/ns-allinone-3.31/ns-3.31$ ls

3) ~/Desktop/ns-allinone-3.31/ns-3.31$tcpdump -nn -tt -r tcp-starserver-8-1.pcap

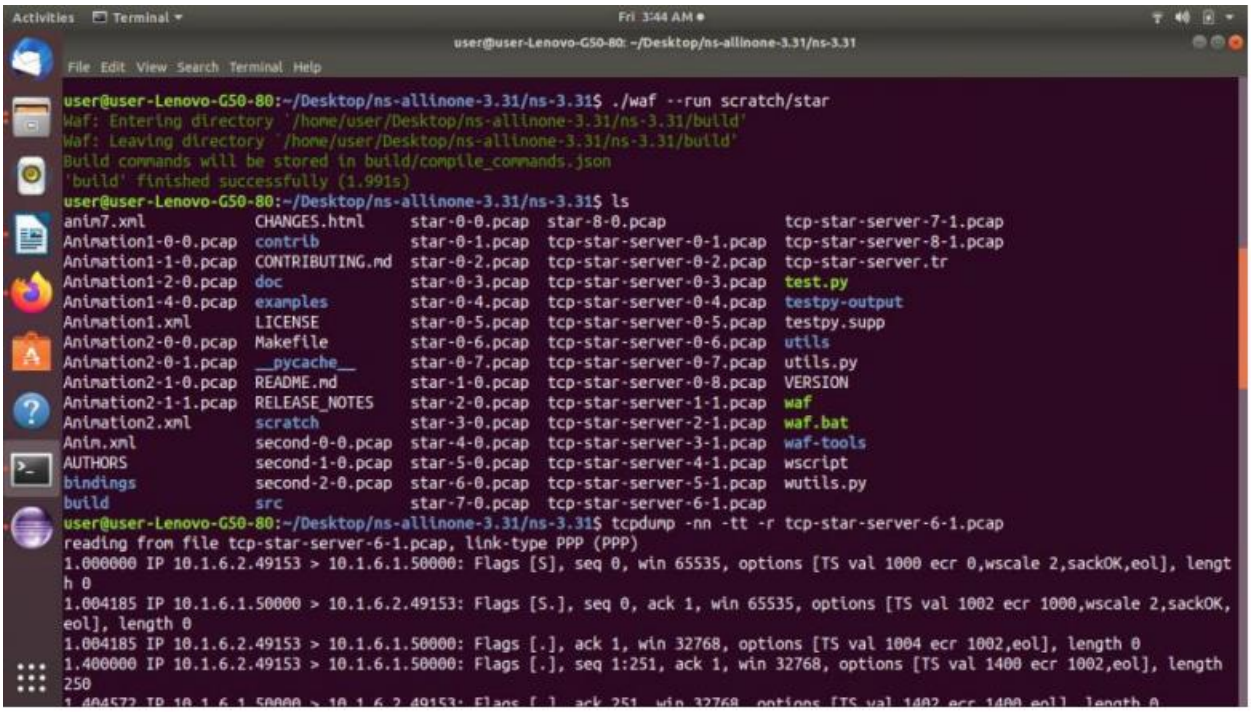


Figure 4 : Output of Star topology as packet sniffer

PRACTICAL 8: WRITE A PROGRAM TO SIMULATE BUS TOPOLOGY.

Bus Network Topology:

194.15.1.0
n0 ---------- n1 n2 n3 n4
point-to-point | | | |
==========
LAN(BUS) 194.15.2.0

Program: Made changes in second.cc

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"

using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("SecondScriptExample");
int main (int argc, char *argv[])
{
	bool verbose = true;
	uint32_t nCsma = 3;
	CommandLine cmd (__FILE__);
	cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
	cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);
	cmd.Parse(argc,argv);
	if (verbose)
	{
		LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
		LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
	}
	nCsma = nCsma == 0 ? 1 : nCsma;
	NodeContainer p2pNodes;
	p2pNodes.Create (2);
	NodeContainer csmaNodes;
	csmaNodes.Add (p2pNodes.Get (1));
```

```
    csmaNodes.Create (nCsma);
    PointToPointHelper pointToPoint;
    pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
    pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
    NetDeviceContainer p2pDevices;
    p2pDevices = pointToPoint.Install (p2pNodes);
    CsmaHelper csma;
    csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
    csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));
    NetDeviceContainer csmaDevices;
    csmaDevices = csma.Install (csmaNodes);
    InternetStackHelper stack;
    stack.Install (p2pNodes.Get (0));
    stack.Install (csmaNodes);
    Ipv4AddressHelper address;
    address.SetBase ("194.15.1.0", "255.255.255.0");
    Ipv4InterfaceContainer p2pInterfaces;
    p2pInterfaces = address.Assign (p2pDevices);
    address.SetBase ("194.15.2.0", "255.255.255.0");
    Ipv4InterfaceContainer csmaInterfaces;
    csmaInterfaces = address.Assign (csmaDevices);
    UdpEchoServerHelper echoServer (15);
    ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));
    serverApps.Start (Seconds (1.0));
    serverApps.Stop (Seconds (10.0));
    UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 15);
    echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
    echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
    echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
    ApplicationContainer clientApps = echoClient.Install (p2pNodes.Get(0));
    clientApps.Start (Seconds (2.0));
    clientApps.Stop (Seconds (10.0));
    Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
    pointToPoint.EnablePcapAll("second");
    csma.EnablePcap ("second", csmaDevices.Get (1), true);
    Simulator::Run();
    Simulator::Destroy();
    return 0;
}
```

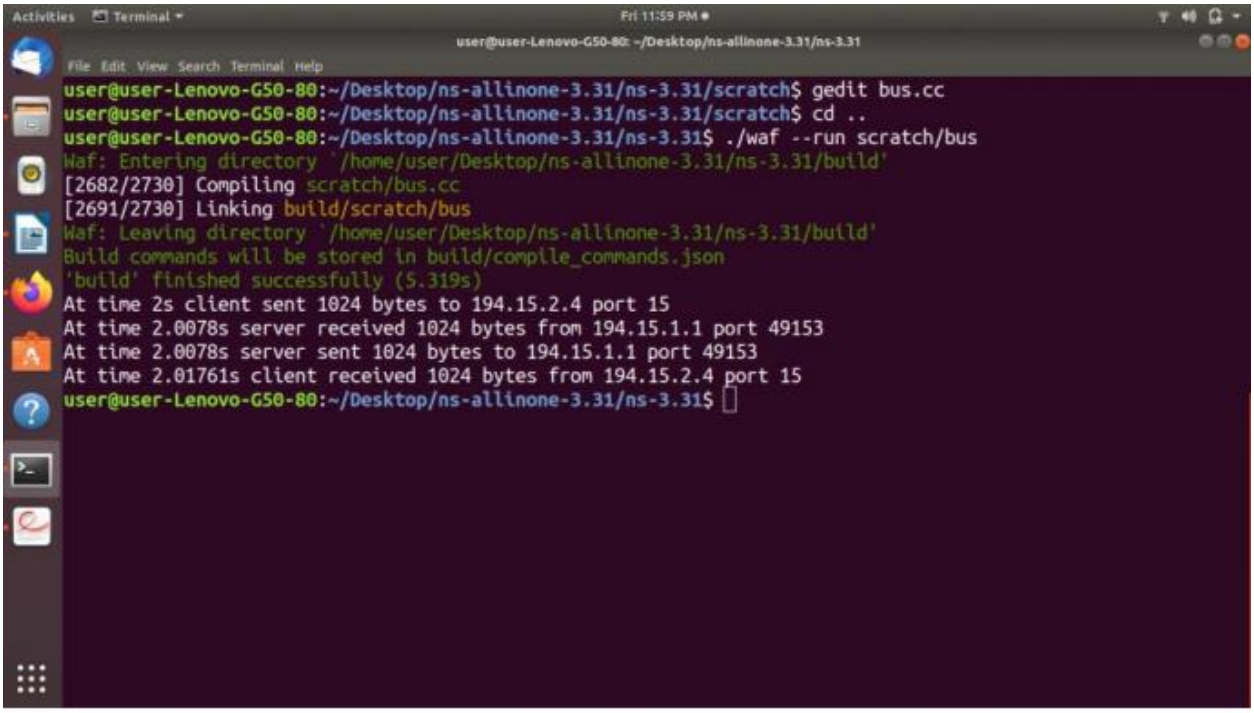


Figure 5 : Output of Bus topology

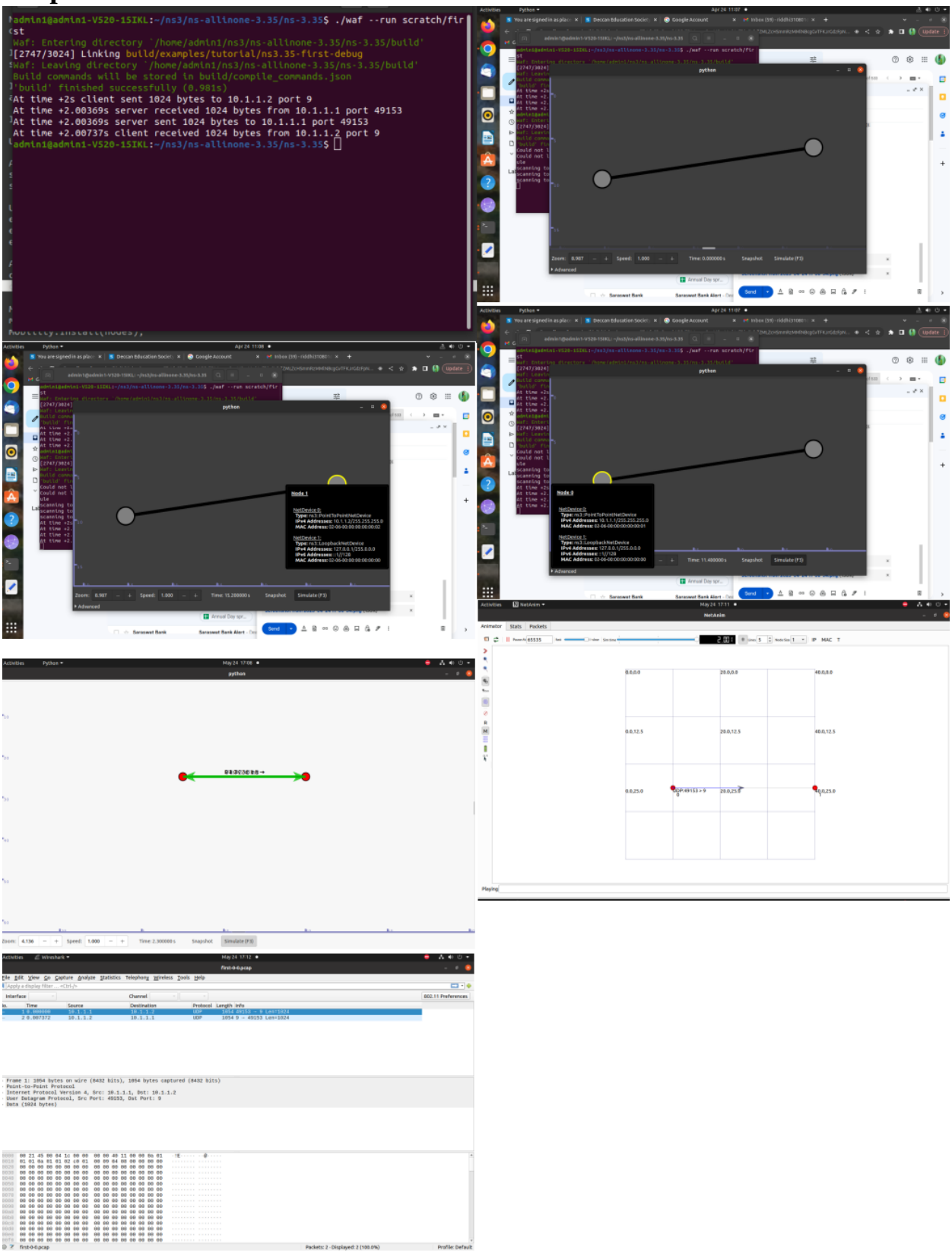
PRACTICAL 9: PROGRAM TO SIMULATE TRAFFIC BETWEEN TWO NODES.

Code:

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
//netanimator
#include "ns3/netanim-module.h"
#include "ns3/mobility-module.h"

// Default Network Topology
//
// 10.1.1.0
// n0 -------------- n1
// point-to-point
//

using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
int main (int argc, char *argv[])
{
	CommandLine cmd (__FILE__);
	cmd.Parse (argc, argv);

	Time::SetResolution (Time::NS);
	LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
	LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
	NodeContainer nodes;
	nodes.Create (2);
	PointToPointHelper pointToPoint;
	pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
	pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
	NetDeviceContainer devices;
	devices = pointToPoint.Install (nodes);
	InternetStackHelper stack;
	stack.Install (nodes);
```

```
    Ipv4AddressHelper address;
    address.SetBase ("10.1.1.0", "255.255.255.0");
    Ipv4InterfaceContainer interfaces = address.Assign (devices);
    UdpEchoServerHelper echoServer (9);
    ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
    serverApps.Start (Seconds (1.0));
    serverApps.Stop (Seconds (10.0));
    UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
    echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
    echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
    echoClient.SetAttribute ("PacketSize", UintegerValue (1060));
    ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
    clientApps.Start (Seconds (2.0));
    clientApps.Stop (Seconds (10.0));
    MobilityHelper mobility;
    mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
    mobility.Install(nodes);
    AnimationInterface anim("first.xml");
    AnimationInterface::SetConstantPosition(nodes.Get(0),10,25);
    AnimationInterface::SetConstantPosition(nodes.Get(1),40,25);

    anim.EnablePacketMetadata(true);
    Simulator::Run ();
    Simulator::Destroy ();
    return 0;
}
```

Output:


admin1@admin1-V520-15IKL:~/ns3/ns-allinone-3.35/ns-3.35$ ./waf --run scratch/first
Waf: Entering directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
[2747/3024] Linking build/examples/tutorial/ns3.35-first-debug
Waf: Leaving directory `/home/admin1/ns3/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.981s)
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
admin1@admin1-V520-15IKL:~/ns3/ns-allinone-3.35/ns-3.35$