

**Name: Rajvardhan Ganpatrao Patil**  
**Div: A    Roll No: 40**

## **Assignment**

### **1. Point-to-Point Topology**

**Definition:** In point-to-point topology, two devices are directly connected to each other, typically via a dedicated link. This is a simple and direct connection.

#### **Advantages:**

- Simple and reliable.
- Easy to set up and configure.
- Provides high speed and minimal interference.

#### **Disadvantages:**

- Limited to only two devices.
- Not scalable for large networks.

#### **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"

#include "ns3/netanim-module.h"

#include "ns3/mobility-module.h"


//Added for flow monitor

#include "ns3/flow-monitor.h"

#include "ns3/flow-monitor-helper.h"


// Default Network Topology
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

//

// 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);

// create a node

NodeContainer nodes;

nodes.Create(2);

// It instantiates a PointToPointHelper object on the stack.

PointToPointHelper P;

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
P.SetChannelAttribute("Delay",StringValue("2ms"));
```

```
NetDeviceContainer devices;
```

```
devices=P.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)));
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
echoClient.SetAttribute("PacketSize",UIntegerValue(1024));
```

```
ApplicationContainer clientApps = echoClient.Install(nodes.Get(0));
```

```
clientApps.Start(Seconds(2.0));
```

```
clientApps.Stop(Seconds(10.0));
```

```
AnimationInterface anim("point2point.xml");
```

```
AnimationInterface::SetConstantPosition(nodes.Get(0), 10, 25);
```

```
AnimationInterface::SetConstantPosition(nodes.Get(1), 40, 25);
```

```
anim.EnablePacketMetadata(true);
```

```
P.EnablePcapAll("first");
```

```
// Flow monitor
```

```
Ptr<FlowMonitor> flowMonitor;
```

```
FlowMonitorHelper flowHelper;
```

```
flowMonitor = flowHelper.InstallAll();
```

```
Simulator::Stop (Seconds (10.0));
```

```
//Following line is added for flow monitor
```

```
Simulator::Run ();
```

```
flowMonitor->SerializeToXmlFile("first_flow.xml", true, true);
```

Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

Simulator::Destroy();

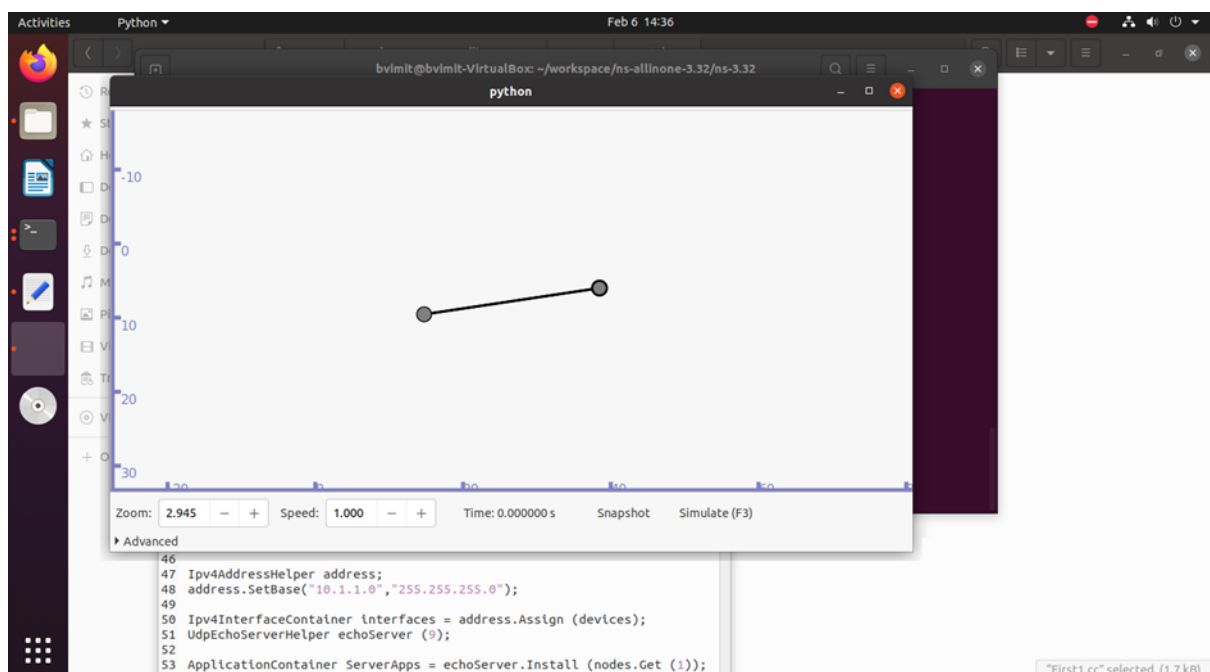
return 0;

}

### ./waf --run scratch/First

```
Waf: Leaving directory `/home/bvmlt/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.235s)
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
Could not load plugin 'show_last_packets.py': No module named 'kiwi'
Could not load icon applets-screenshooter due to missing gnomedesktop Python module
scanning topology: 2 nodes...
scanning topology: calling graphviz layout
scanning topology: all done.
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
At time +2.25932s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.25932s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.51865s client received 1024 bytes from 10.1.1.2 port 9
```

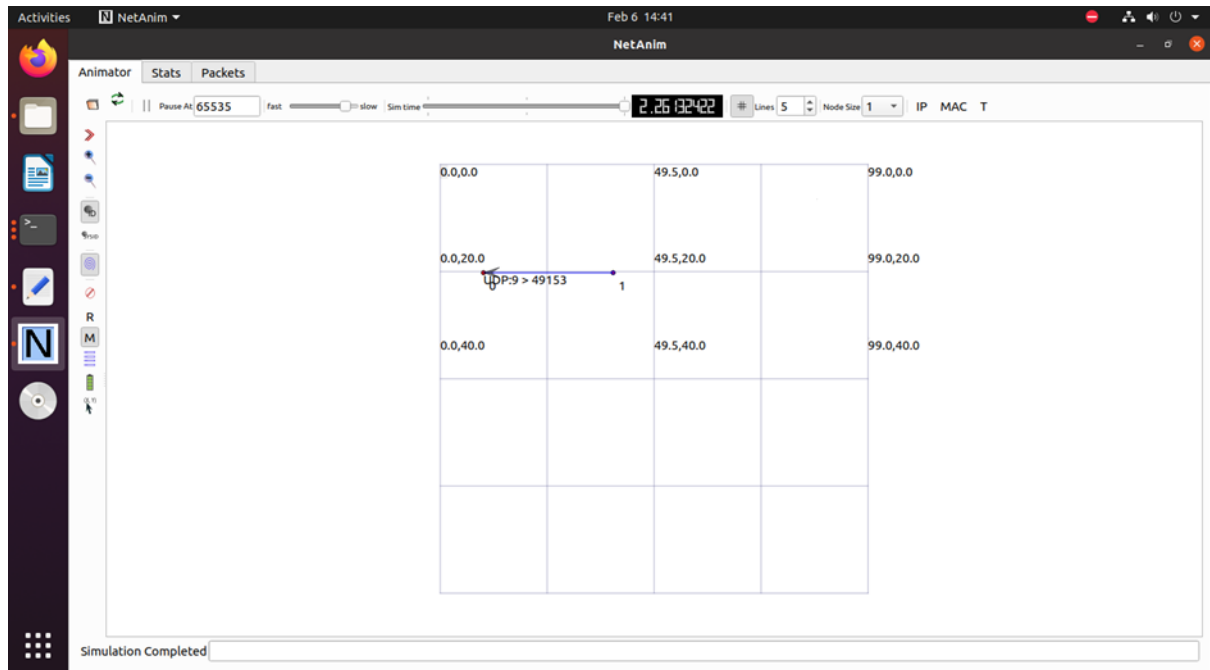
### ./waf --run scratch/First -vis\



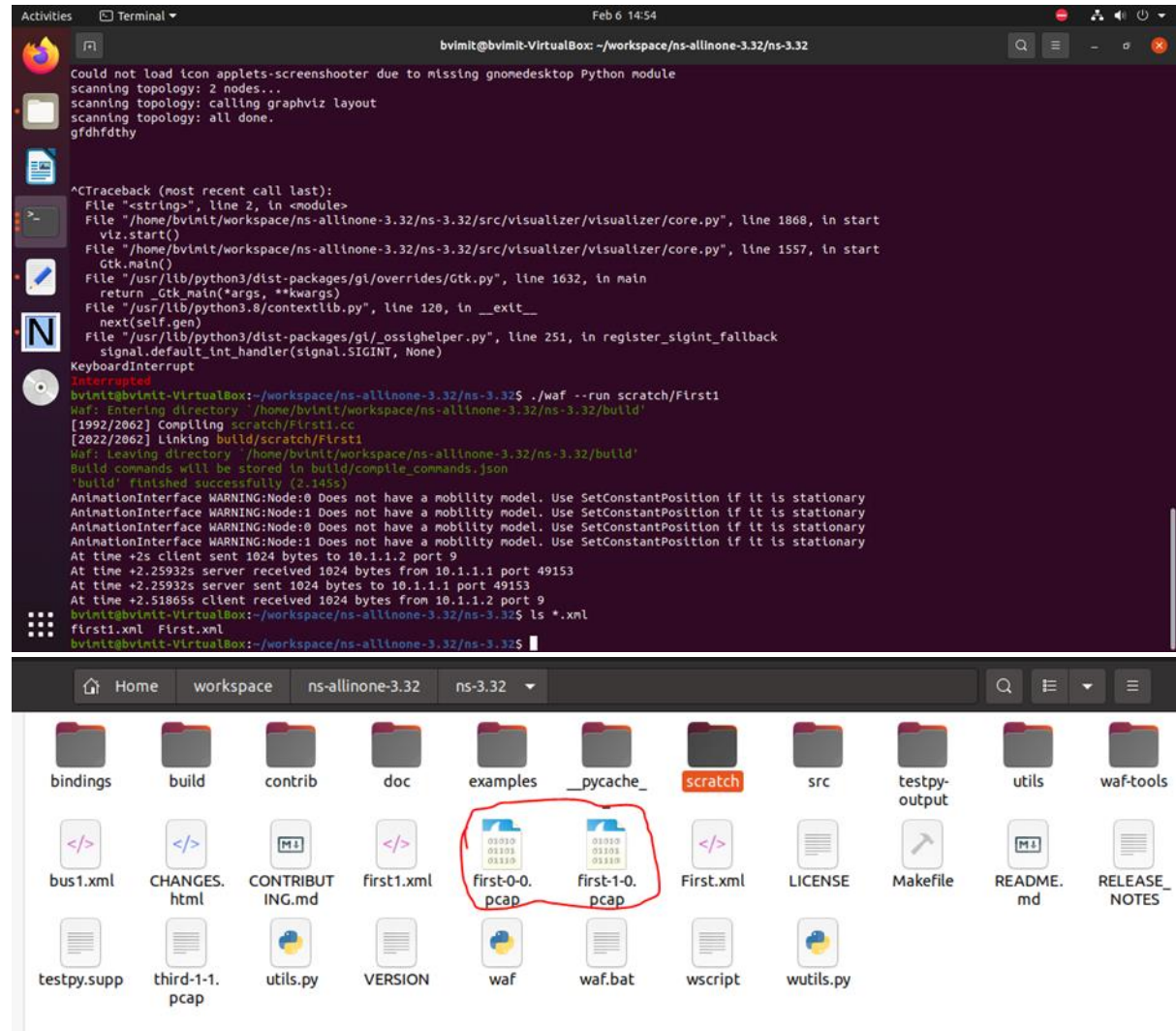
Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

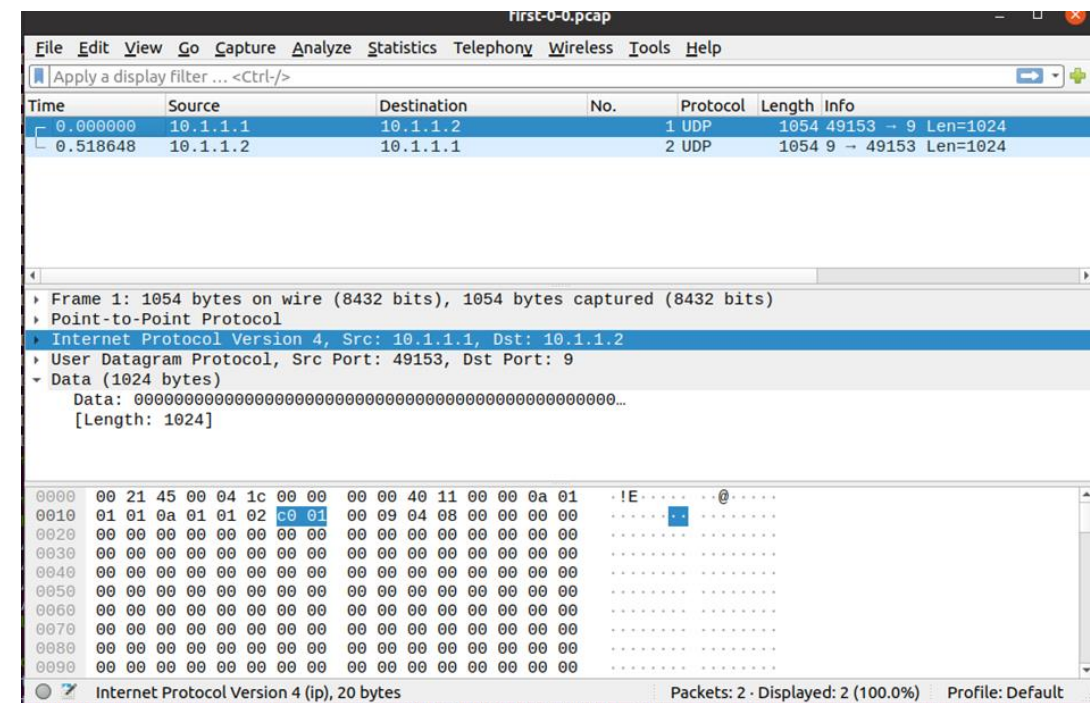
./NetAnim



**Div: A      Roll No: 40**



# Wireshark



Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

## Practical 12 - Flow Monitor

AnimatorStatsPackets

Flow-monitor

Sim TimeFont Size 10FlowMon fileNodesShow Table

AllNone12

Flow Id:1  
=====

UDP 10.1.1.1/49153---->10.1.1.2/9

Tx bitrate:8.416e+09kbps  
Rx bitrate:8.416e+09kbps  
Mean delay:259.324ms  
Packet Loss ratio:0%

timeFirstTxPacket= 2e+09ns  
timeFirstRxPacket= 2.25932e+09ns  
timeLastTxPacket= 2e+09ns  
timeLastRxPacket= 2.25932e+09ns  
delaySum= 2.59324e+08ns  
jitterSum= 0ns  
lastDelay= 2.59324e+08ns  
txBytes= 1052  
rxBytes= 1052  
txPackets= 1  
rxPackets= 1  
lostPackets= 0  
timesForwarded= 0

delayHistogram nBins:260  
Index:259 Start:0.259 Width:0.001 Count:1

jitterHistogram nBins:0

packetSizeHistogram nBins:53  
Index:52 Start:1040 Width:20 Count:1

flowInterruptionsHistogram nBins:0

Flow Id:2  
=====

UDP 10.1.1.2/9---->10.1.1.1/49153

Tx bitrate:8.416e+09kbps  
Rx bitrate:8.416e+09kbps  
Mean delay:259.324ms  
Packet Loss ratio:0%

timeFirstTxPacket= 2.25932e+09ns  
timeFirstRxPacket= 2.51865e+09ns  
timeLastTxPacket= 2.25932e+09ns  
timeLastRxPacket= 2.51865e+09ns  
delaySum= 2.59324e+08ns  
jitterSum= 0ns  
lastDelay= 2.59324e+08ns  
txBytes= 1052  
rxBytes= 1052  
txPackets= 1  
rxPackets= 1  
lostPackets= 0  
timesForwarded= 0

delayHistogram nBins:260  
Index:259 Start:0.259 Width:0.001 Count:1

jitterHistogram nBins:0

packetSizeHistogram nBins:53  
Index:52 Start:1040 Width:20 Count:1

flowInterruptionsHistogram nBins:0

Flow Probes:

Index:0  
FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:0ns  
FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:2.59324e+08ns

Index:1

Index:2  
FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:2.59324e+08ns  
FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:0ns

Index:3





**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

## **2. Bus Topology**

**Definition:** In bus topology, all devices are connected to a single central cable (the bus). Data sent from any device is broadcasted to all devices, but only the intended recipient processes it.

### **Advantages:**

- Simple and cost-effective for small networks.
- Easy to implement and extend.

### **Disadvantages:**

- Performance degrades as more devices are added.
- If the central bus cable fails, the whole network is down.
- Difficult to troubleshoot due to the lack of central control.

### **Code:-**

```
#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"

#include "ns3/netanim-module.h"

#include "ns3/mobility-module.h"

//Added for flow monitor

#include "ns3/flow-monitor.h"

#include "ns3/flow-monitor-helper.h"

using namespace ns3;
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
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);
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
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 ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;

p2pInterfaces = address.Assign (p2pDevices);


address.SetBase ("10.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer csmaInterfaces;

csmaInterfaces = address.Assign (csmaDevices);
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
UdpEchoServerHelper echoServer (9);
```

```
ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));
```

```
serverApps.Start (Seconds (1.0));
```

```
serverApps.Stop (Seconds (10.0));
```

```
UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);
```

```
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);
```

```
// for Net Anim
```

```
MobilityHelper mobility;
```

```
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
```

```
mobility.Install(p2pNodes);
```

```
mobility.Install(csmaNodes);
```

```
AnimationInterface anim("bus1.xml");
```

```
AnimationInterface::SetConstantPosition (p2pNodes.Get(0), 10, 25);
```

```
AnimationInterface::SetConstantPosition (p2pNodes.Get(1), 40, 25);
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
AnimationInterface::SetConstantPosition (csmaNodes.Get(1), 40,25);
```

```
AnimationInterface::SetConstantPosition (csmaNodes.Get(2), 50,25);
```

```
AnimationInterface::SetConstantPosition (csmaNodes.Get(3), 60,25);
```

```
anim.EnablePacketMetadata(true);
```

```
// Flow monitor
```

```
Ptr<FlowMonitor> flowMonitor;
```

```
FlowMonitorHelper flowHelper;
```

```
flowMonitor = flowHelper.InstallAll();
```

```
Simulator::Stop (Seconds (10.0));
```

```
Simulator::Run ();
```

```
//Following line is added for flow monitor
```

```
flowMonitor->SerializeToXmlFile("bus_flow.xml", true, true);
```

```
Simulator::Destroy ();
```

```
return 0;
```

```
}
```

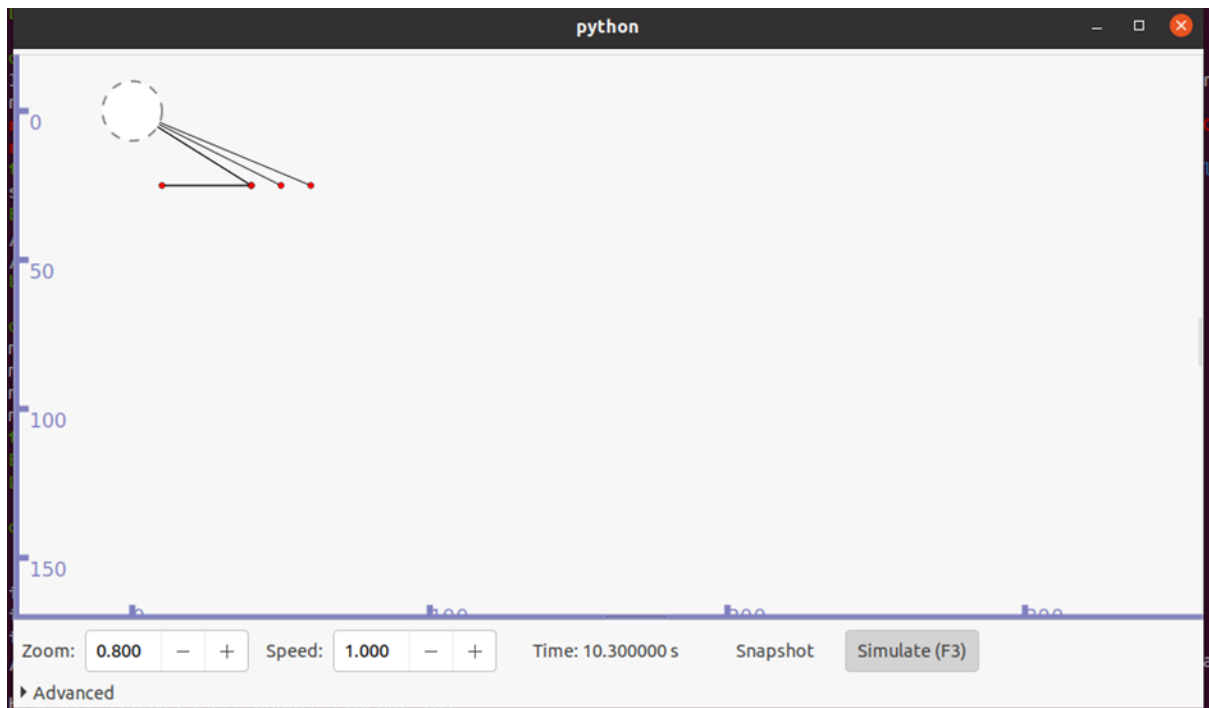
Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

/waf --run scratch/bus

```
bvinit@bvinit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/bus-topology
waf: Entering directory '/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
[2019/2060] Compiling scratch/bus-topology.cc
[2020/2060] Linking build/scratch/bus-topology
waf: Leaving directory '/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.171s)
At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.0078s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.0078s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.0176s client received 1024 bytes from 10.1.2.4 port 9
bvinit@bvinit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$
```

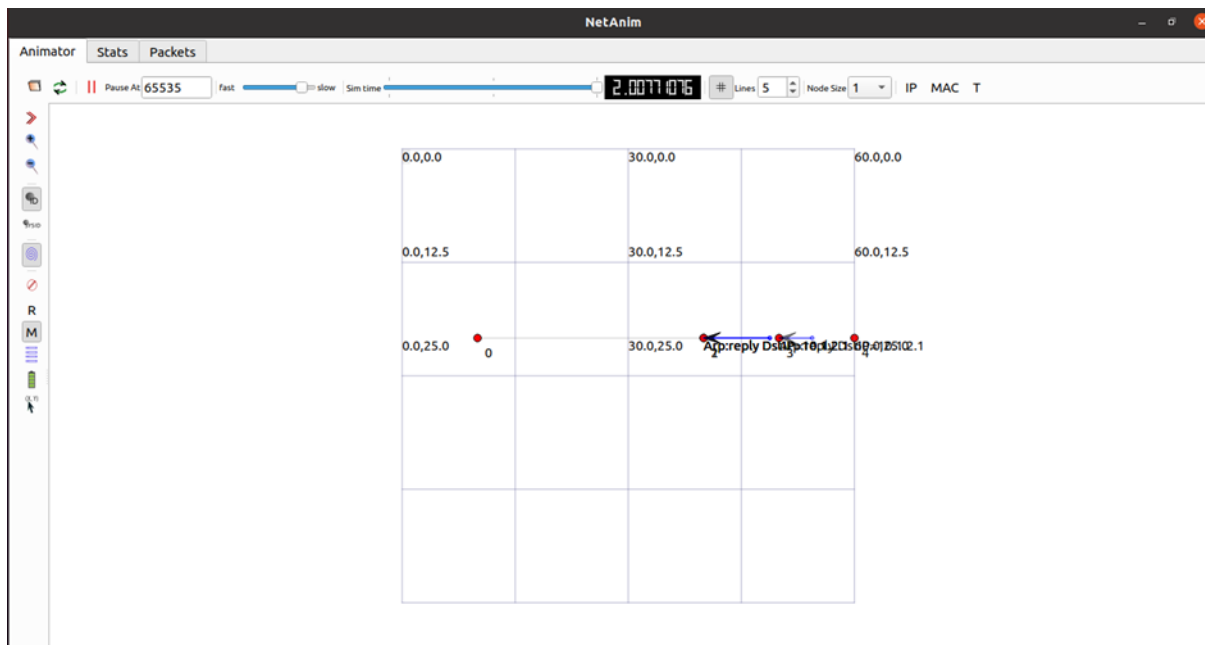
/waf --run scratch/bus --vis



Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

./NetAim



## Wireshark

The Wireshark interface shows a packet capture of 'second-2-0.pcap'. The packet list contains 6 packets:

Time	Source	Destination	No.	Protocol	Length	Info
0.000000	00:00:00_00:00:03	Broadcast	1	ARP	64	who has 10.1.2.4? Tell 10.1
0.000012	00:00:00_00:00:06	00:00:00_00:00:03	2	ARP	64	10.1.2.4 is at 00:00:00:00:
0.000105	10.1.1.1	10.1.2.4	3	UDP	1070	49153 → 9 Len=1024
0.000117	00:00:00_00:00:06	Broadcast	4	ARP	64	Who has 10.1.2.1? Tell 10.1
0.000130	00:00:00_00:00:03	00:00:00_00:00:06	5	ARP	64	10.1.2.1 is at 00:00:00:00:
0.000223	10.1.2.4	10.1.1.1	6	UDP	1070	9 → 49153 Len=1024

The packet details pane shows the first packet (Frame 1) as an Ethernet II, Src: 00:00:00\_00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff). The packet is an Address Resolution Protocol (request).

The packet bytes pane shows the raw data of the first packet:

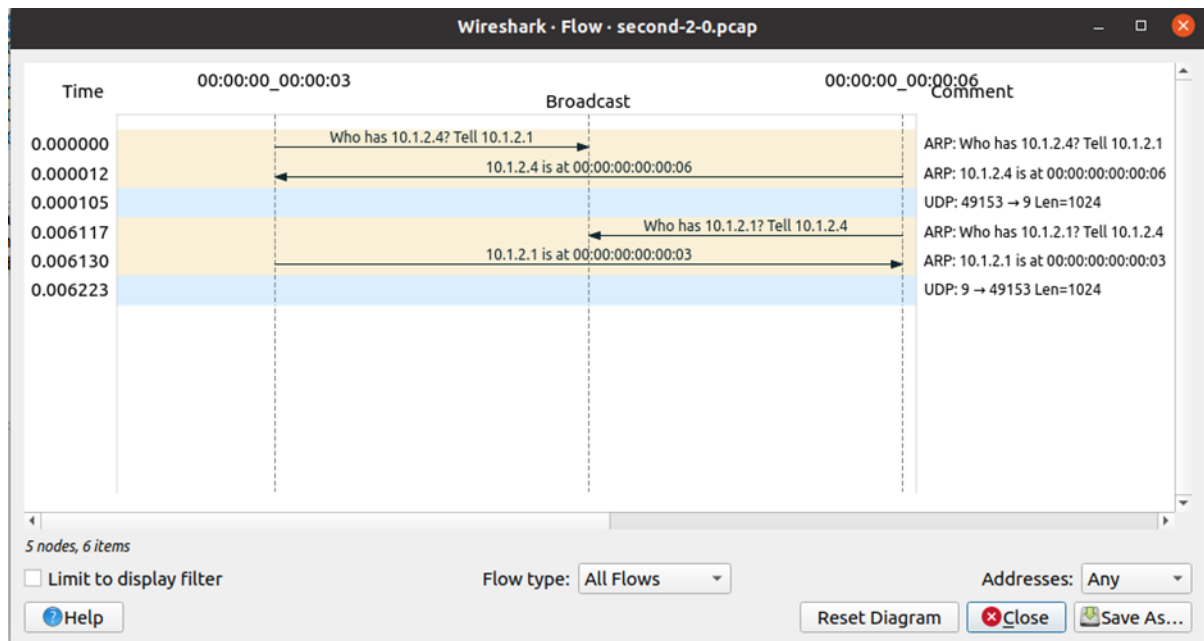
```
0000 ff ff ff ff ff ff 00 00 00 00 03 08 06 00 01 .....
0010 08 00 06 04 00 01 00 00 00 00 03 0a 01 02 01 .....
0020 ff ff ff ff ff ff 0a 01 02 04 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

The file explorer shows a directory with the following files:

Name	Size	Modified	Recency
Makefile	256 bytes	7 Oct 2020	
README.md	4.2 kB	7 Oct 2020	
RELEASE_NOTES	171.8 kB	7 Oct 2020	
second-0-0.pcap	2.2 kB	14:16	
second-1-0.pcap	2.2 kB	14:16	

Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40



## Practical 12 - Flow Monitor

<p>Flow Id:1</p> <p>=====</p> <p>UDP 10.1.1.1/49153----&gt;10.1.2.4/9</p> <p>Tx bitrate:-8.416e+09kbps</p> <p>Rx bitrate:-8.416e+09kbps</p> <p>Mean delay:7.80392ms</p> <p>Packet Loss ratio:0%</p> <p>timeFirstTxPacket= 2e+09ns</p> <p>timeFirstRxPacket= 2.0078e+09ns</p> <p>timeLastTxPacket= 2e+09ns</p> <p>timeLastRxPacket= 2.0078e+09ns</p> <p>delaySum= 7.80392e+06ns</p> <p>jitterSum= 0ns</p> <p>lastDelay= 7.80392e+06ns</p> <p>txBytes= 1052</p> <p>rxBytes= 1052</p> <p>txPackets= 1</p> <p>rxPackets= 1</p> <p>lostPackets= 0</p> <p>timesForwarded= 1</p> <p>delayHistogram nBins:8</p> <p>Index:7 Start:0.007 Width:0.001 Count:1</p> <p>jitterHistogram nBins:0</p> <p>packetSizeHistogram nBins:53</p> <p>Index:52 Start:1040 Width:20 Count:1</p> <p>flowInterruptionsHistogram nBins:0</p>	<p>Flow Id:2</p> <p>=====</p> <p>UDP 10.1.2.4/9----&gt;10.1.1.1/49153</p> <p>Tx bitrate:-8.416e+09kbps</p> <p>Rx bitrate:-8.416e+09kbps</p> <p>Mean delay:9.80392ms</p> <p>Packet Loss ratio:0%</p> <p>timeFirstTxPacket= 2.0078e+09ns</p> <p>timeFirstRxPacket= 2.01761e+09ns</p> <p>timeLastTxPacket= 2.0078e+09ns</p> <p>timeLastRxPacket= 2.01761e+09ns</p> <p>delaySum= 9.80392e+06ns</p> <p>jitterSum= 0ns</p> <p>lastDelay= 9.80392e+06ns</p> <p>txBytes= 1052</p> <p>rxBytes= 1052</p> <p>txPackets= 1</p> <p>rxPackets= 1</p> <p>lostPackets= 0</p> <p>timesForwarded= 1</p> <p>delayHistogram nBins:10</p> <p>Index:9 Start:0.009 Width:0.001 Count:1</p> <p>jitterHistogram nBins:0</p> <p>packetSizeHistogram nBins:53</p> <p>Index:52 Start:1040 Width:20 Count:1</p> <p>flowInterruptionsHistogram nBins:0</p>	<p>Flow Probes:</p> <p>Index:0</p> <p>FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:0ns</p> <p>FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:9.80392e+06ns</p> <p>Index:1</p> <p>Index:2</p> <p>FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:3.6864e+06ns</p> <p>FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:6.11752e+06ns</p> <p>Index:3</p> <p>Index:4</p> <p>Index:5</p> <p>Index:6</p> <p>Index:7</p> <p>Index:8</p> <p>FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:7.80392e+06ns</p> <p>FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:0ns</p> <p>Index:9</p>
---	--	--



**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

### **3. Star Topology**

**Definition:** In star topology, each device is connected to a central device (typically a switch or hub). Data is transmitted from one device to the hub, and then the hub forwards it to the intended device.

**Advantages:**

- Easy to manage and configure.
- If one device fails, the rest of the network remains functional.
- Performance does not degrade as much with the addition of new devices.

**Disadvantages:**

- Relies heavily on the central hub or switch; if the central device fails, the entire network is affected.
- Can become expensive with large numbers of devices because of the need for more cables and central hardware.

**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-point-layout-module.h"


//Added for flow monitor

#include "ns3/flow-monitor.h"

#include "ns3/flow-monitor-helper.h"

// Network topology (default)

//      n2 n3 n4      .
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A Roll No: 40**

```
//      \ | /      .
```

```
//      \|      .
```

```
//      n1--- n0---n5      .
```

```
//      /\      .
```

```
//      / | \      .
```

```
//      n8 n7 n6      .
```

```
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"));
```

```
    uint32_t nSpokes = 8;
```

```
    CommandLine cmd (__FILE__);
```

```
    cmd.AddValue ("nSpoke", "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"));
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

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)

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
{  
  
    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 ("Star1");  
  
  
//Netanim code  
  
std::string animFile = "star1.xml";  
  
  
// Set the bounding box for animation  
  
star.BoundingBox (1, 1, 100, 100);  
  
  
// Create the animation object and configure for specified output  
  
AnimationInterface anim (animFile);
```

**Name: Rajvardhan Ganpatrao Patil**  
**Div: A    Roll No: 40**

```
NS_LOG_INFO ("Run Simulation.");
```

```
// Flow monitor
```

```
Ptr<FlowMonitor> flowMonitor;
```

```
FlowMonitorHelper flowHelper;
```

```
flowMonitor = flowHelper.InstallAll();
```

```
Simulator::Stop (Seconds (10.0));
```

```
//Following line is added for flow monitor
```

```
Simulator::Run ();
```

```
flowMonitor->SerializeToXmlFile("star_flow.xml", true, true);
```

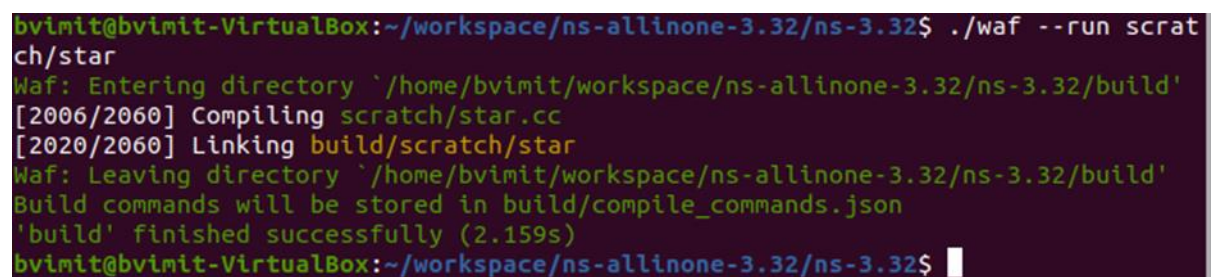
```
Simulator::Destroy ();
```

```
NS_LOG_INFO ("Done.");
```

```
return 0;
```

```
}
```

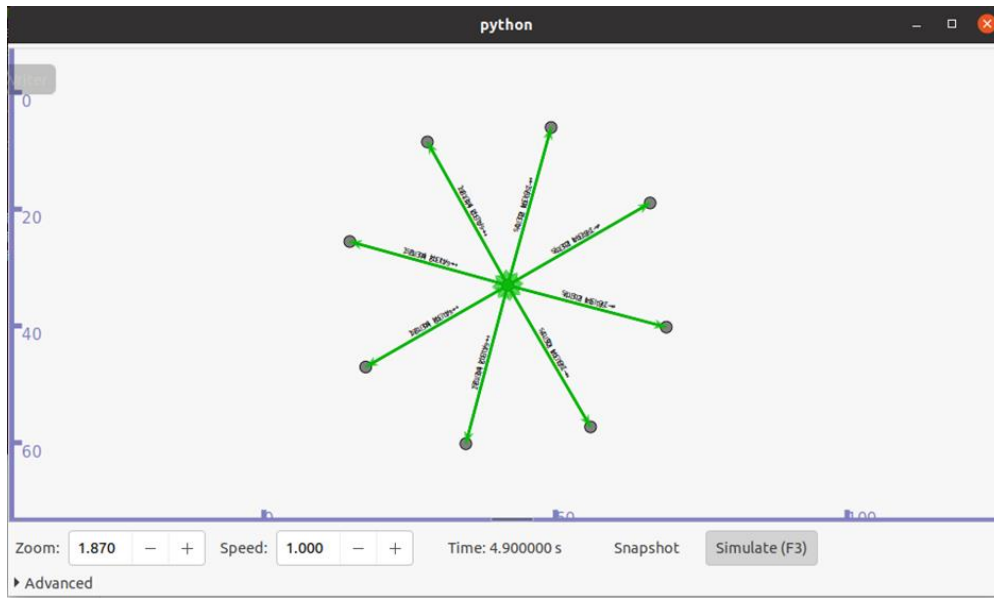
```
./waf --run scratch/star
```



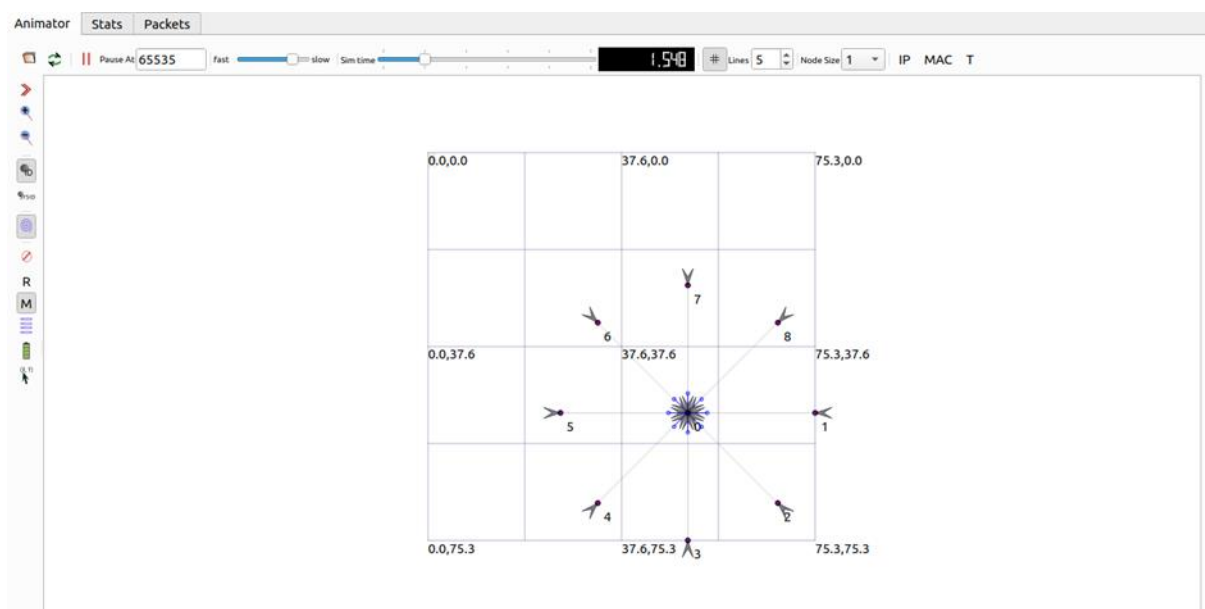
```
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/star
Waf: Entering directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
[2006/2060] Compiling scratch/star.cc
[2020/2060] Linking build/scratch/star
Waf: Leaving directory `/home/bvimit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.159s)
bvimit@bvimit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$
```

Name: Rajvardhan Ganpatrao Patil  
Div: A Roll No: 40

./waf --run scratch/star --vis

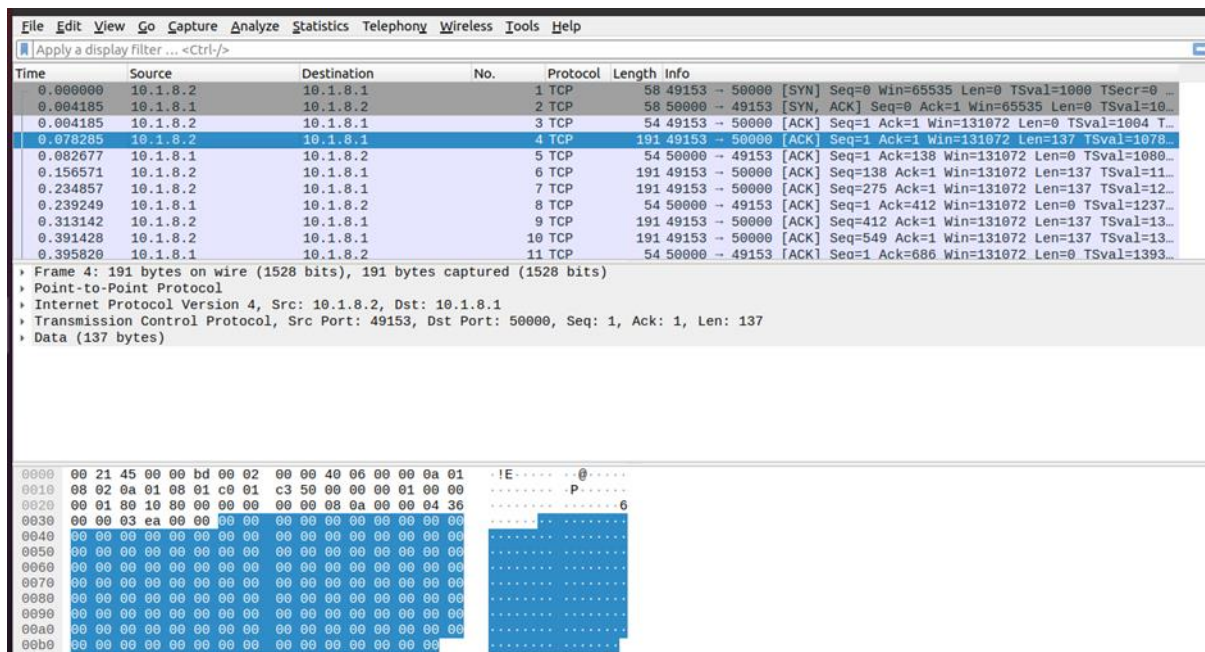
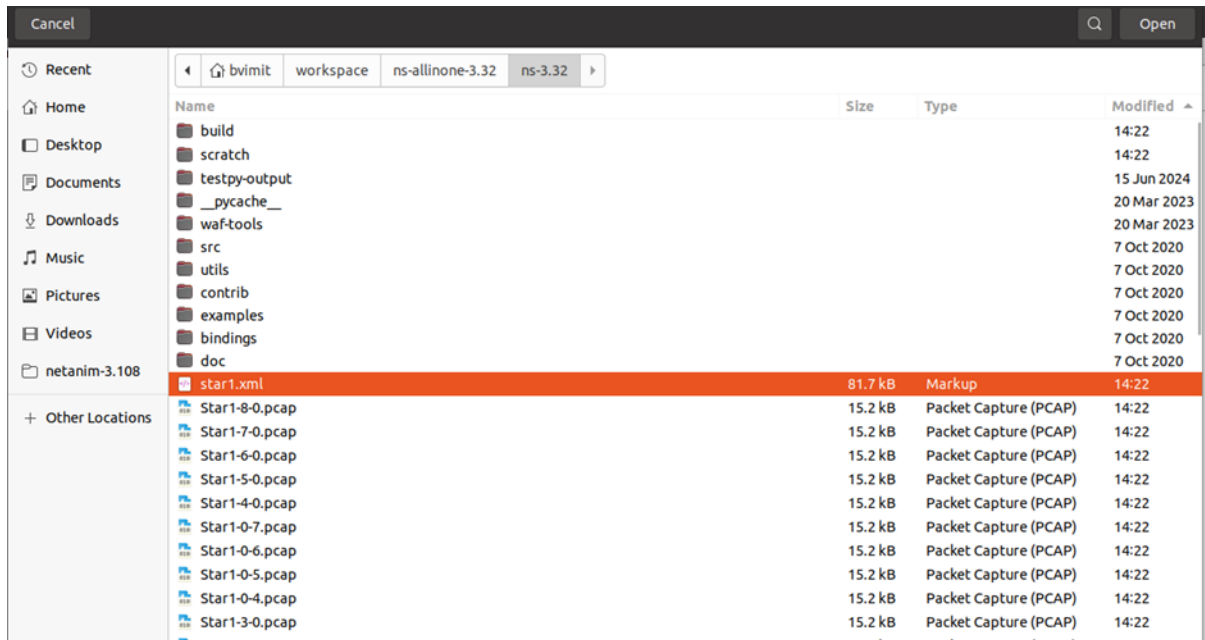


./NetAnim



Name: Rajvardhan Ganpatrao Patil  
Div: A Roll No: 40

## WireShark



Name: Rajvardhan Ganpatrao Patil  
Div: A Roll No: 40

## Practical 12 - Flow Monitor

Animator	Stats	Packets
Flow-monitor	Sim Time	Font Size 10 FlowMon file Nodes Show Table
All	Flow Id:1 =====	Flow Id:2 =====
None	TCP 10.1.1.2/49153---->10.1.1.1/50000	TCP 10.1.2.2/49153---->10.1.2.1/50000
1	Tx bitrate:19.5766kbps	Tx bitrate:19.5766kbps
2	Rx bitrate:19.5753kbps	Rx bitrate:19.5753kbps
3	Mean delay:2.29577ms	Mean delay:2.29577ms
4	Packet Loss ratio:0%	Packet Loss ratio:0%
5	timeFirstTxPacket= 1e+09ns	timeFirstTxPacket= 1e+09ns
6	timeFirstRxPacket= 1.00209e+09ns	timeFirstRxPacket= 1.00209e+09ns
7	timeLastTxPacket= 4.288e+09ns	timeLastTxPacket= 4.288e+09ns
8	timeLastRxPacket= 4.29031e+09ns	timeLastRxPacket= 4.29031e+09ns
9	delaySum= 1.01014e+08ns	delaySum= 1.01014e+08ns
10	jitterSum= 225600ns	jitterSum= 225600ns
11	lastDelay= 1.01014e+08ns	lastDelay= 1.01014e+08ns
12	txBytes= 8046	txBytes= 8046
13	rxBytes= 8046	rxBytes= 8046
14	txPackets= 44	txPackets= 44
15	rxPackets= 44	rxPackets= 44
16	lostPackets= 0	lostPackets= 0
	timesForwarded= 0	timesForwarded= 0
	delayHistogram nBins:3	delayHistogram nBins:3
	Index:2 Start:0.002 Width:0.001 Count:44	Index:2 Start:0.002 Width:0.001 Count:44
	jitterHistogram nBins:1	jitterHistogram nBins:1
	Index:0 Start:0 Width:0.001 Count:43	Index:0 Start:0 Width:0.001 Count:43
	packetSizeHistogram nBins:10	packetSizeHistogram nBins:10
	Index:2 Start:40 Width:20 Count:2	Index:2 Start:40 Width:20 Count:2
	Index:9 Start:180 Width:20 Count:42	Index:9 Start:180 Width:20 Count:42
	flowInterruptionsHistogram nBins:0	flowInterruptionsHistogram nBins:0
	Flow Id:5 =====	Flow Id:6 =====
	TCP 10.1.5.2/49153---->10.1.5.1/50000	TCP 10.1.6.2/49153---->10.1.6.1/50000
	Flow Id:3 =====	Flow Id:7 =====
	TCP 10.1.3.2/49153---->10.1.3.1/50000	TCP 10.1.7.2/49153---->10.1.7.1/50000
	Flow Id:4 =====	Flow Id:8 =====
	TCP 10.1.4.2/49153---->10.1.4.1/50000	TCP 10.1.8.2/49153---->10.1.8.1/50000



**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

## **4. Mesh Topology**

**Definition:** In mesh topology, each device is connected to every other device. This can be full mesh (where every device is connected to all others) or partial mesh (where only some devices are connected to others).

### **Advantages:**

- Provides redundancy; if one connection fails, another route can be used.
- Highly reliable and fault-tolerant.
- Excellent performance in large networks.

### **Disadvantages:**

- Requires a large number of cables and ports, which can make it expensive.
- Complex to set up and maintain.
- Scaling can become difficult due to the large number of connections.

### **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"
//Added for flow monitor
#include "ns3/flow-monitor.h"
#include "ns3/flow-monitor-helper.h"
using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("Mesh");

int main (int argc, char *argv[])
{
    // Set up some default values for the simulation.
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
Config::SetDefault ("ns3::OnOffApplication::PacketSize", UIntegerValue (137));
```

```
Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("14kb/s"));
```

```
uint32_t nNodes = 5; // Number of nodes in the mesh topology
```

```
CommandLine cmd (__FILE__);
```

```
cmd.AddValue ("nNodes", "Number of nodes in the mesh", nNodes);
```

```
cmd.Parse (argc,argv);
```

```
NS_LOG_INFO("Building mesh topology");
```

```
// Create point-to-point links between all pairs of nodes
```

```
PointToPointHelper pointToPoint;
```

```
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
```

```
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
NodeContainer nodes;
```

```
nodes.Create (nNodes); // Create the mesh nodes
```

```
// Create devices and install them on the nodes
```

```
NetDeviceContainer devices;
```

```
for (uint32_t i = 0; i < nNodes; ++i)
```

```
{
```

```
  for (uint32_t j = i + 1; j < nNodes; ++j)
```

```
  {
```

```
    NetDeviceContainer linkDevices = pointToPoint.Install (nodes.Get (i), nodes.Get (j));
```

```
    devices.Add (linkDevices);
```

```
  }
```

```
}
```

```
NS_LOG_INFO ("Install internet stack on all nodes.");
```

```
InternetStackHelper internet;
```

```
internet.Install (nodes);
```

```
NS_LOG_INFO ("Assign IP Addresses.");
```

```
Ipv4AddressHelper ipv4;
```

```
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
```

```
Ipv4InterfaceContainer interfaces = ipv4.Assign (devices);
```

```
NS_LOG_INFO ("Create applications.");
```

```
uint16_t port = 50000;
```

```
ApplicationContainer apps;
```

```
// Create a packet sink on the first node to receive packets.
```

```
Address localAddress (InetSocketAddress (Ipv4Address::GetAny (), port));
```

```
PacketSinkHelper packetSinkHelper ("ns3::TcpSocketFactory", localAddress);
```

```
apps.Add (packetSinkHelper.Install (nodes.Get (0))); // Installing sink on node 0
```

```
apps.Start (Seconds (1.0));
```

```
apps.Stop (Seconds (10.0));
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
// Create OnOff applications to send TCP to the first node from every other node.
OnOffHelper onOffHelper ("ns3::TcpSocketFactory", Address ());
onOffHelper.SetAttribute ("OnTime", StringValue ("ns3::ConstantRandomVariable[Constant=1]"));
onOffHelper.SetAttribute ("OffTime", StringValue ("ns3::ConstantRandomVariable[Constant=0]"));

ApplicationContainer onOffApps;
for (uint32_t i = 1; i < nNodes; ++i)
{
    AddressValue remoteAddress (InetSocketAddress (interfaces.GetAddress (0), port)); // Sending
data to node 0
    onOffHelper.SetAttribute ("Remote", remoteAddress);
    onOffApps.Add (onOffHelper.Install (nodes.Get (i)));
}
onOffApps.Start (Seconds (1.0));
onOffApps.Stop (Seconds (10.0));

NS_LOG_INFO ("Enable static global routing.");
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

NS_LOG_INFO ("Enable pcap tracing.");
pointToPoint.EnablePcapAll ("Mesh");

// Netanim code
std::string animFile = "mesh.xml";

// Set the bounding box for animation
AnimationInterface anim (animFile);
anim.SetMaxPktsPerTraceFile (10000);

NS_LOG_INFO ("Run Simulation.");
// Flow monitor
Ptr<FlowMonitor> flowMonitor;
FlowMonitorHelper flowHelper;
flowMonitor = flowHelper.InstallAll();
Simulator::Stop (Seconds (10.0));

//Following line is added for flow monitor
Simulator::Run ();
flowMonitor->SerializeToXmlFile("mesh_flow.xml", true, true);
Simulator::Destroy ();
NS_LOG_INFO ("Done.");

return 0;
}
```

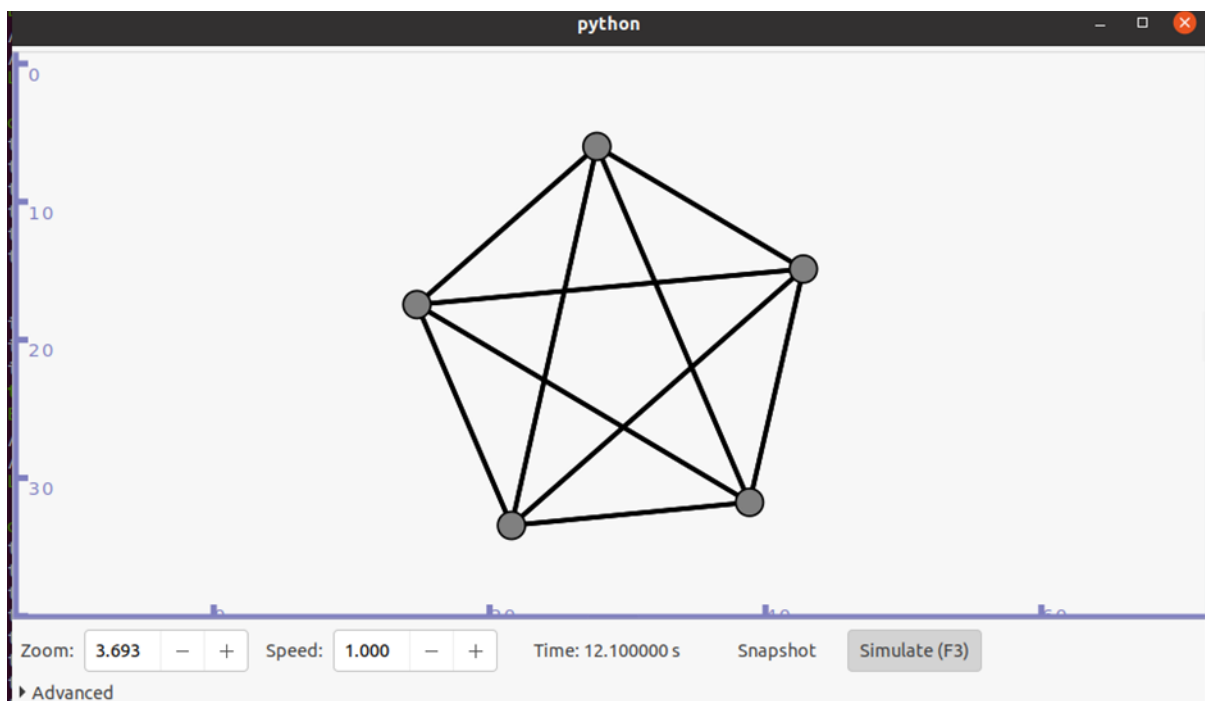
Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

./waf --run scratch/mesh

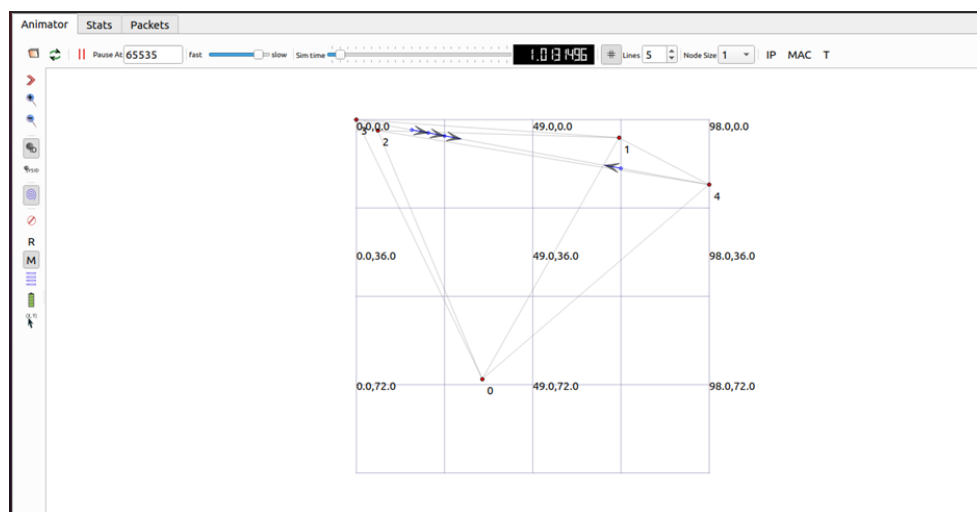
```
bvinit@bvinit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/mesh
Waf: Entering directory `/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
[2014/2062] Compiling scratch/mesh.cc
[2022/2062] Linking build/scratch/mesh
Waf: Leaving directory `/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.129s)
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:6 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:7 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:6 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:7 Does not have a mobility model. Use SetConstantPosition if it is stationary
```

./waf --run scratch/mesh --vis



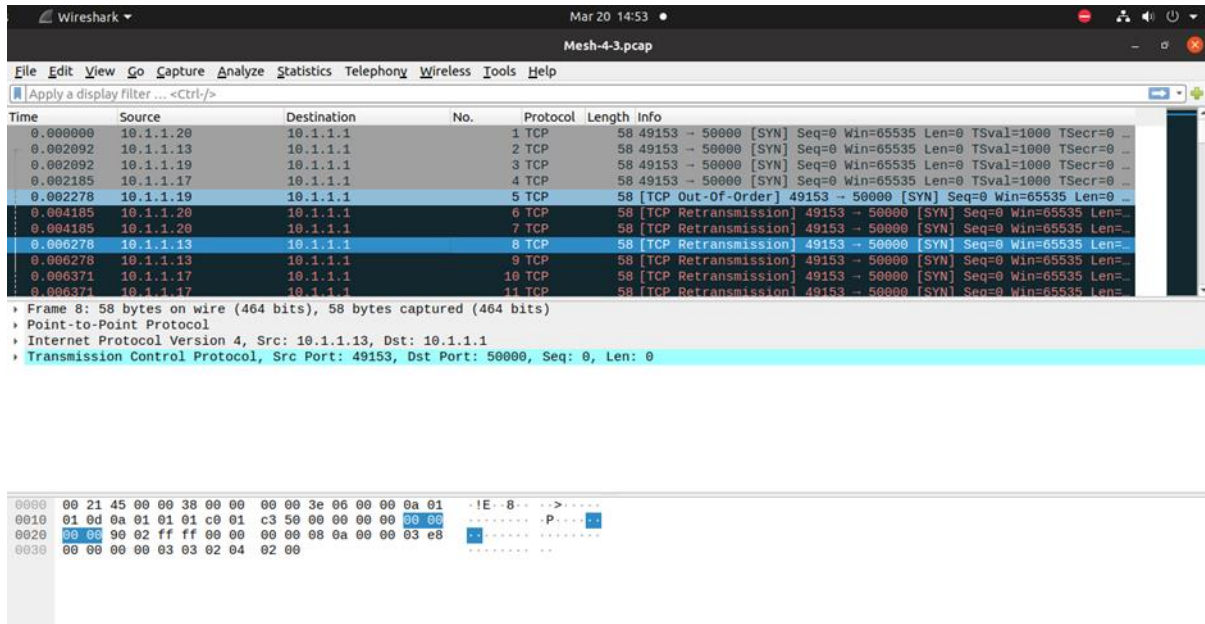
Name: Rajvardhan Ganpatrao Patil  
Div: A Roll No: 40

./NetAnim

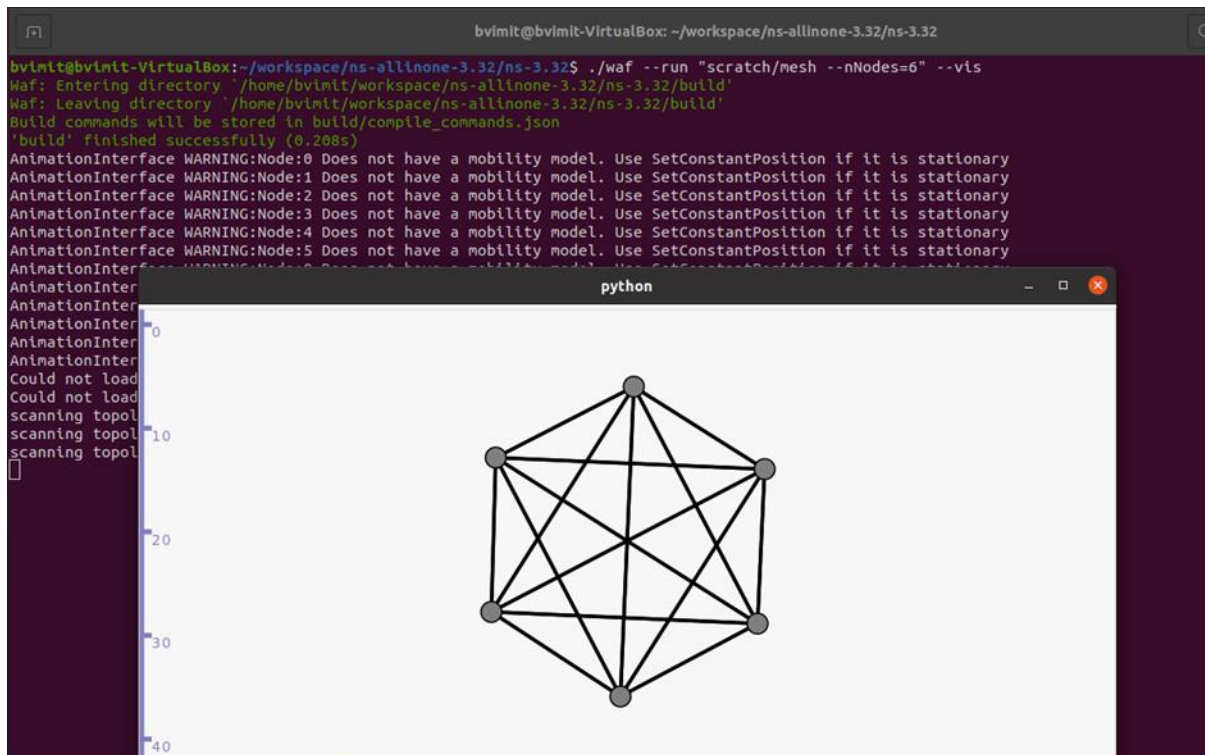


wireshark

Name: Rajvardhan Ganpatrao Patil  
Div: A Roll No: 40



./waf --run "scratch/mesh --nNodes=6" --vis



## Practical 12 - Flow Monitor

Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

Animator	Stats	Packets				
Flow-monitor	Sim Time	Font Size 10	FlowMon file	Nodes	Show Table	
All	Flow Id:1	Flow Id:2	Flow Id:3	Flow Id:4		
None	=====	=====	=====	=====		
1	TCP 10.1.1.13/49153---->10.1.1.1/50000	TCP 10.1.1.17/49153---->10.1.1.1/50000	TCP 10.1.1.19/49153---->10.1.1.1/50000	TCP 10.1.1.20/49153---->10.1.1.1/50000		
2	Tx bitrate:0.298667kbps	Tx bitrate:0.298667kbps	Tx bitrate:0.298667kbps	Tx bitrate:0.298667kbps		
3	Rx bitrate:0kbps	Rx bitrate:0kbps	Rx bitrate:0kbps	Rx bitrate:0kbps		
4	Mean delay:-1ms	Mean delay:-1ms	Mean delay:-1ms	Mean delay:-1ms		
	Packet Loss ratio:100%	Packet Loss ratio:100%	Packet Loss ratio:100%	Packet Loss ratio:100%		
	timeFirstTxPacket= 1e+09ns	timeFirstTxPacket= 1e+09ns	timeFirstTxPacket= 1e+09ns	timeFirstTxPacket= 1e+09ns		
	timeFirstRxPacket= 0ns	timeFirstRxPacket= 0ns	timeFirstRxPacket= 0ns	timeFirstRxPacket= 0ns		
	timeLastTxPacket= 4e+09ns	timeLastTxPacket= 4e+09ns	timeLastTxPacket= 4e+09ns	timeLastTxPacket= 4e+09ns		
	timeLastRxPacket= 0ns	timeLastRxPacket= 0ns	timeLastRxPacket= 0ns	timeLastRxPacket= 0ns		
	delaySum= 0ns	delaySum= 0ns	delaySum= 0ns	delaySum= 0ns		
	jitterSum= 0ns	jitterSum= 0ns	jitterSum= 0ns	jitterSum= 0ns		
	lastDelay= 0ns	lastDelay= 0ns	lastDelay= 0ns	lastDelay= 0ns		
	txBytes= 112	txBytes= 112	txBytes= 112	txBytes= 112		
	rxBytes= 0	rxBytes= 0	rxBytes= 0	rxBytes= 0		
	txPackets= 2	txPackets= 2	txPackets= 2	txPackets= 2		
	rxPackets= 0	rxPackets= 0	rxPackets= 0	rxPackets= 0		
	lostPackets= 2	lostPackets= 2	lostPackets= 2	lostPackets= 2		
	timesForwarded= 0	timesForwarded= 0	timesForwarded= 0	timesForwarded= 0		
	Packets Dropped:	Packets Dropped:	Packets Dropped:	Packets Dropped:		
	No Route:0	No Route:0	No Route:0	No Route:0		
	TTL Expire:2	TTL Expire:2	TTL Expire:2	TTL Expire:2		
	Bytes Dropped:	Bytes Dropped:	Bytes Dropped:	Bytes Dropped:		
	No Route:0	No Route:0	No Route:0	No Route:0		
	TTL Expire:112	TTL Expire:112	TTL Expire:112	TTL Expire:112		
	delayHistogram nBins:0	delayHistogram nBins:0	delayHistogram nBins:0	delayHistogram nBins:0		
	jitterHistogram nBins:0	jitterHistogram nBins:0	jitterHistogram nBins:0	jitterHistogram nBins:0		
	packetSizeHistogram nBins:0	packetSizeHistogram nBins:0	packetSizeHistogram nBins:0	packetSizeHistogram nBins:0		

## 5. Hybrid Topology

**Definition:** A hybrid topology is a combination of two or more different types of topologies (e.g., star and bus, star and ring, etc.). It is often used to meet specific needs and requirements of large and complex networks.

### Advantages:

- Flexible and adaptable to a variety of network requirements.
- Can offer the benefits of multiple topologies.
- Scalable and efficient.

### Disadvantages:

- Expensive to implement and maintain.
- Can be complex to troubleshoot.
- Requires careful planning to ensure compatibility of different topologies.

### Code:-

```
#include "ns3/core-module.h"
```

```
#include "ns3/point-to-point-module.h"
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
#include "ns3/network-module.h"
```

```
#include "ns3/applications-module.h"
```

```
#include "ns3/mobility-module.h"
```

```
#include "ns3/csma-module.h"
```

```
#include "ns3/internet-module.h"
```

```
#include "ns3/yans-wifi-helper.h"
```

```
#include "ns3/ssid.h"
```

```
//Added for flow monitor
```

```
#include "ns3/flow-monitor.h"
```

```
#include "ns3/flow-monitor-helper.h"
```

```
using namespace ns3;
```

```
NS_LOG_COMPONENT_DEFINE ("ThirdScriptExample");
```

```
int
```

```
main (int argc, char *argv[])
```

```
{
```

```
    bool verbose = true;
```

```
    uint32_t nCsma = 3;
```

```
    uint32_t nWifi = 3;
```

```
    bool tracing = false;
```

```
    CommandLine cmd (__FILE__);
```

```
    cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
```

```
    cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
```

```
    cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);
```

```
    cmd.AddValue ("tracing", "Enable pcap tracing", tracing);
```



**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
cmd.Parse (argc,argv);

if (nWifi > 18)

{

    std::cout << "nWifi should be 18 or less; otherwise grid layout exceeds the bounding box"
<< std::endl;

    return 1;

}


if (verbose)

{

    LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);

    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);

}


NodeContainer p2pNodes;

p2pNodes.Create (2);


PointToPointHelper pointToPoint;

pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));


NetDeviceContainer p2pDevices;

p2pDevices = pointToPoint.Install (p2pNodes);


NodeContainer csmaNodes;

csmaNodes.Add (p2pNodes.Get (1));

csmaNodes.Create (nCsmas);
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
CsmaHelper csma;

csma.SetChannelAttribute ("DataRate",StringValue ("100Mbps"));

csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));


NetDeviceContainer csmaDevices;

csmaDevices = csma.Install (csmaNodes);


NodeContainer wifiStaNodes;

wifiStaNodes.Create (nWifi);

NodeContainer wifiApNode = p2pNodes.Get (0);


YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();

YansWifiPhyHelper phy = YansWifiPhyHelper::Default ();

phy.SetChannel (channel.Create ());


WifiHelper wifi;

wifi.SetRemoteStationManager ("ns3::AarfWifiManager");


WifiMacHelper mac;

Ssid ssid = Ssid ("ns-3-ssid");

mac.SetType ("ns3::StaWifiMac",

             "Ssid", SsidValue (ssid),

             "ActiveProbing", BooleanValue (false));


NetDeviceContainer staDevices;

staDevices = wifi.Install (phy, mac, wifiStaNodes);
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
mac.SetType ("ns3::ApWifiMac",
```

```
    "Ssid", SsidValue (ssid));
```

```
NetDeviceContainer apDevices;
```

```
apDevices = wifi.Install (phy, mac, wifiApNode);
```

```
MobilityHelper mobility;
```

```
mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
```

```
    "MinX", DoubleValue (0.0),
```

```
    "MinY", DoubleValue (0.0),
```

```
    "DeltaX", DoubleValue (5.0),
```

```
    "DeltaY", DoubleValue (10.0),
```

```
    "GridWidth", UIntegerValue (3),
```

```
    "LayoutType", StringValue ("RowFirst"));
```

```
mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
```

```
    "Bounds", RectangleValue (Rectangle (-50, 50, -50, 50)));
```

```
mobility.Install (wifiStaNodes);
```

```
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
```

```
mobility.Install (wifiApNode);
```

```
InternetStackHelper stack;
```

```
stack.Install (csmaNodes);
```

```
stack.Install (wifiApNode);
```

```
stack.Install (wifiStaNodes);
```

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

Ipv4AddressHelper address;

address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;

p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("10.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer csmaInterfaces;

csmaInterfaces = address.Assign (csmaDevices);

address.SetBase ("10.1.3.0", "255.255.255.0");

address.Assign (staDevices);

address.Assign (apDevices);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));

serverApps.Start (Seconds (1.0));

serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (nCsma), 9);

echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));

echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));

ApplicationContainer clientApps =

    echoClient.Install (wifiStaNodes.Get (nWifi - 1));

clientApps.Start (Seconds (2.0));

**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

```
clientApps.Stop (Seconds (10.0));
```

```
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

```
// Flow monitor
```

```
Ptr<FlowMonitor> flowMonitor;
```

```
FlowMonitorHelper flowHelper;
```

```
flowMonitor = flowHelper.InstallAll();
```

```
Simulator::Stop (Seconds (10.0));
```

```
if (tracing == true)
```

```
{
```

```
    pointToPoint.EnablePcapAll ("third");
```

```
    phy.EnablePcap ("third", apDevices.Get (0));
```

```
    csma.EnablePcap ("third", csmaDevices.Get (0), true);
```

```
}
```

```
//Following line is added for flow monitor
```

```
Simulator::Run ();
```

```
flowMonitor->SerializeToXmlFile("hybrid_flow.xml", true, true);
```

```
Simulator::Destroy ();
```

```
return 0;
```

```
}
```

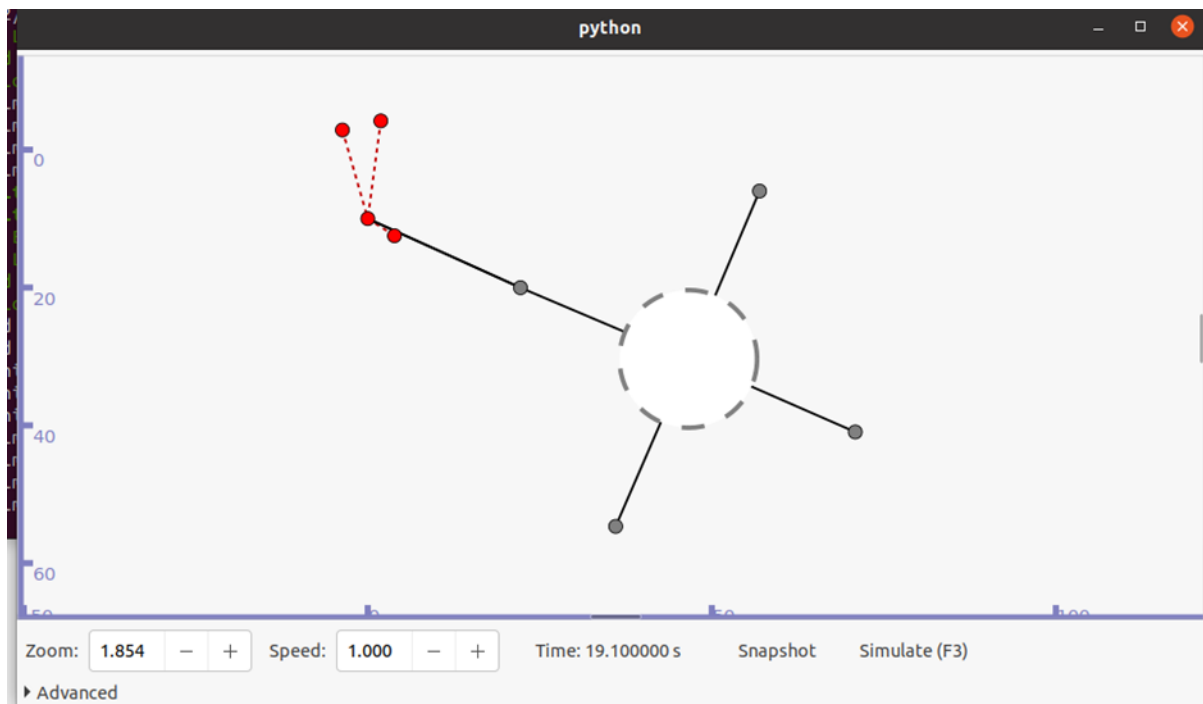
**./waf --run scratch/hybrid-topology**

Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

```
bvinit@bvinit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/hybrid-topology
Waf: Entering directory '/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
[1987/2062] Compiling scratch/hybrid-topology.cc
[1989/2062] Linking build/scratch/subdir/subdir
[1990/2062] Compiling scratch/bus-topology.cc
[2021/2062] Linking build/scratch/hybrid-topology
[2022/2062] Linking build/scratch/bus-topology
Waf: Leaving directory '/home/bvinit/workspace/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.059s)
At time +2s client sent 1024 bytes to 10.1.2.4 port 9
At time +2.01799s server received 1024 bytes from 10.1.3.3 port 49153
At time +2.01799s server sent 1024 bytes to 10.1.3.3 port 49153
At time +2.03371s client received 1024 bytes from 10.1.2.4 port 9
bvinit@bvinit-VirtualBox:~/workspace/ns-allinone-3.32/ns-3.32$
```

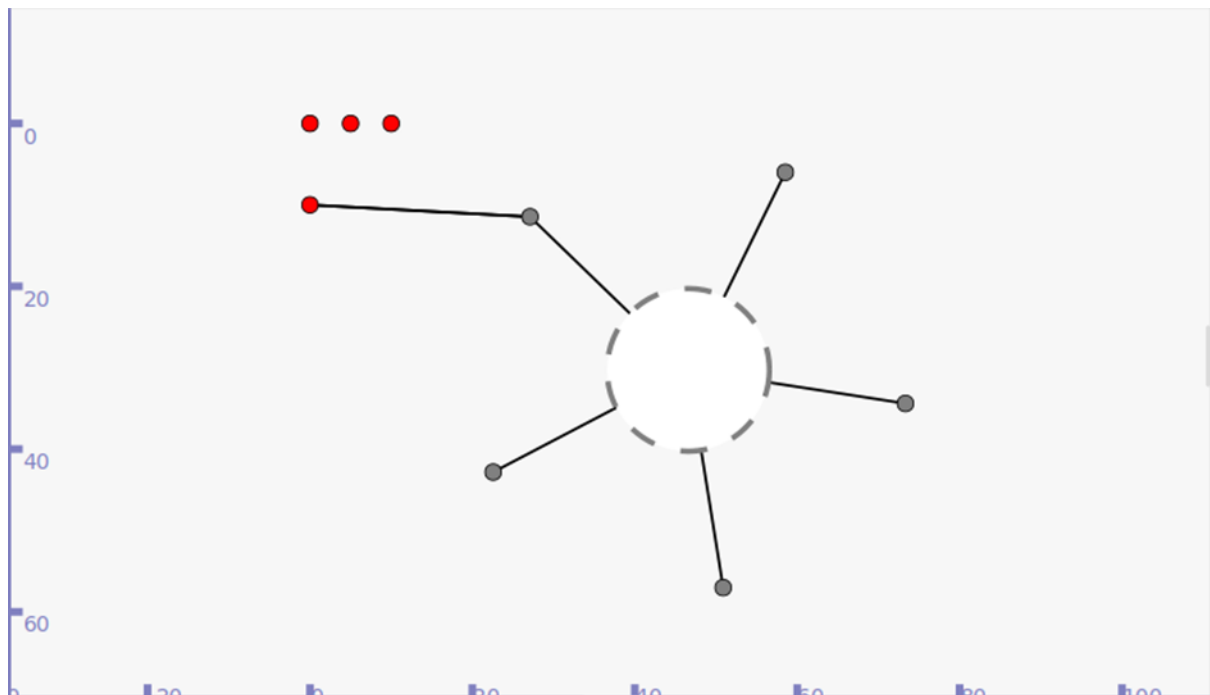
**`./waf --run scratch/hybrid-topology --vis`**



**Name: Rajvardhan Ganpatrao Patil**

**Div: A    Roll No: 40**

**`./waf --run "scratch/hybrid-topology --nCsmma=4 --tracing=true" --vis`**



**wireshark**

Name: Rajvardhan Ganpatrao Patil

Div: A Roll No: 40

second-2-0.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

Time	Source	Destination	No.	Protocol	Length	Info
0.000000	00:00:00_00:00:03	Broadcast	1	ARP	64	Who has 10.1.2.4? Tell 10.1.2.1
0.000012	00:00:00_00:00:06	00:00:00_00:00:03	2	ARP	64	10.1.2.4 is at 00:00:00:00:00:06
0.000105	10.1.1.1	10.1.2.4	3	UDP	1070	49153 → 9 Len=1024
0.006117	00:00:00_00:00:06	Broadcast	4	ARP	64	Who has 10.1.2.1? Tell 10.1.2.4
0.006130	00:00:00_00:00:03	00:00:00_00:00:06	5	ARP	64	10.1.2.1 is at 00:00:00:00:00:03
0.006223	10.1.2.4	10.1.1.1	6	UDP	1070	9 → 49153 Len=1024

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)

Ethernet II, Src: 00:00:00\_00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Destination: Broadcast (ff:ff:ff:ff:ff:ff)

Source: 00:00:00\_00:00:03 (00:00:00:00:00:03)

Type: ARP (0x0806)

Padding: 00000000000000000000000000000000

Frame check sequence: 0x00000000 [unverified]

[FCS Status: Unverified]

Address Resolution Protocol (request)

Hardware type: Ethernet (1)

0000 ff ff ff ff ff ff 00 00 00 00 03 08 06 00 01

0010 08 00 06 04 00 01 00 00 00 00 03 0a 01 02 01

0020 ff ff ff ff ff ff 0a 01 02 04 00 00 00 00 00 00

0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

second-2-0.pcap Packets: 6 · Displayed: 6 (100.0%) Profile: Default

Wireshark · Flow · second-2-0.pcap

Time 00:00:00\_00:00:03 Broadcast 00:00:00\_00:00:06 Comment

0.000000 Who has 10.1.2.4? Tell 10.1.2.1

0.000012 10.1.2.4 is at 00:00:00:00:00:06

0.000105 UDP: 49153 → 9 Len=1024

0.006117 Who has 10.1.2.1? Tell 10.1.2.4

0.006130 10.1.2.1 is at 00:00:00:00:00:03

0.006223 UDP: 9 → 49153 Len=1024

ARP: Who has 10.1.2.4? Tell 10.1.2.1

ARP: 10.1.2.4 is at 00:00:00:00:00:06

ARP: Who has 10.1.2.1? Tell 10.1.2.4

ARP: 10.1.2.1 is at 00:00:00:00:00:03

5 nodes, 6 items

☐ Limit to display filter

Flow type: All Flows

Addresses: Any

Help Reset Diagram Close Save As...



Name: Rajyardhan Ganpatrao Patil

Div: A     Roll No: 40

**Practical 12 - Flow Monitor**

AnimatorStatsPackets

Flow-monitor

Sim Time

Font Size 10

FlowMon file

Nodes

Show Table

AllNone12

Flow Id:1  
=====

UDP 10.1.3.3/49153-->10.1.2.4/9

Tx bitrate:-8.416e+09kbps  
Rx bitrate:-8.416e+09kbps  
Mean delay:17.989ms  
Packet Loss ratio:0%

timeFirstTxPacket= 2e+09ns  
timeFirstRxPacket= 2.01799e+09ns  
timeLastTxPacket= 2e+09ns  
timeLastRxPacket= 2.01799e+09ns  
delaySum= 1.7989e+07ns  
jitterSum= 0ns  
lastDelay= 1.7989e+07ns  
txBytes= 1052  
rxBytes= 1052  
txPackets= 1  
rxPackets= 1  
lostPackets= 0  
timesForwarded= 2

delayHistogram nBins:18  
Index:17 Start:0.017 Width:0.001 Count:1

jitterHistogram nBins:0

packetSizeHistogram nBins:53  
Index:52 Start:1040 Width:20 Count:1

flowInterruptionsHistogram nBins:0

Flow Id:2  
=====

UDP 10.1.2.4/9-->10.1.3.3/49153

Tx bitrate:-8.416e+09kbps  
Rx bitrate:-8.416e+09kbps  
Mean delay:15.72ms  
Packet Loss ratio:0%

timeFirstTxPacket= 2.01799e+09ns  
timeFirstRxPacket= 2.03371e+09ns  
timeLastTxPacket= 2.01799e+09ns  
timeLastRxPacket= 2.03371e+09ns  
delaySum= 1.572e+07ns  
jitterSum= 0ns  
lastDelay= 1.572e+07ns  
txBytes= 1052  
rxBytes= 1052  
txPackets= 1  
rxPackets= 1  
lostPackets= 0  
timesForwarded= 2

delayHistogram nBins:16  
Index:15 Start:0.015 Width:0.001 Count:1

jitterHistogram nBins:0

packetSizeHistogram nBins:53  
Index:52 Start:1040 Width:20 Count:1

flowInterruptionsHistogram nBins:0

Flow Probes:

Index:0  
FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:8.18513e+06ns  
FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:8.80392e+06ns

Index:1

Index:2  
FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:1.18715e+07ns  
FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:5.11752e+06ns

Index:3

Index:4

Index:5

Index:6

Index:7

Index:8  
FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:1.7989e+07ns  
FlowId:2 Packets:1 Bytes:1052 DelayFromFirstProbeSum:0ns

Index:9

Index:10

Index:11

Index:12

Index:13

Index:14  
FlowId:1 Packets:1 Bytes:1052 DelayFromFirstProbeSum:0ns

