**Practical No 6**

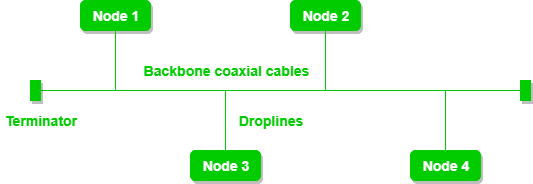
**1.Aim: -** WAP to simulate Bus topology.

**2.Objective: -**

* Understanding of Bus topology
* Implementation of Bus topology using NS3

**3.Theory: -**

Bus topology carries transmitted data through the cable because data reaches each node, the node checks the destination address (MAC/IP address) to determine if it matches their address. If the address does not match with the node, the node does nothing more. But if the addresses of nodes match to addresses contained within the data, then they process knowledge. In the bus, communication between nodes is done through a foremost network cable.



**Advantages of Bus topology:**

* It is the easiest network topology for linearly connecting peripherals or computers.
* It works very efficiently well when there is a small network.
* The length of cable required is less than a star topology.
* It is easy to connect or remove devices in this network without affecting any other device.
* Very cost-effective as compared to another network topology i.e. mesh and star.
* It is easy to understand topology.

**Disadvantages of Bus topology:**

* Bus topology is not good for large networks.
* Identification of problems becomes difficult if the whole network goes down.
* Troubleshooting individual device issues is very hard.
* Need terminators are required at both ends of the main cable.
* Additional devices slow the network down.
* If the main cable is damaged, the whole network fails or splits into two.
* Packet loss is high.
* This network topology is very slow as compared to other topologies.

**4.Program: -**

#include "ns3/core-module.h"

#include "ns3/network-module.h"

#include "ns3/netanim-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"

// Default Network Topology

//

// 10.1.1.0

// n0 -------------- n1 n2 n3 n4

// point-to-point | | | |

// ================

// LAN 10.1.2.0

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 ("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);

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);

// Create the animation object and configure for specified output

AnimationInterface anim ("busT.xml");

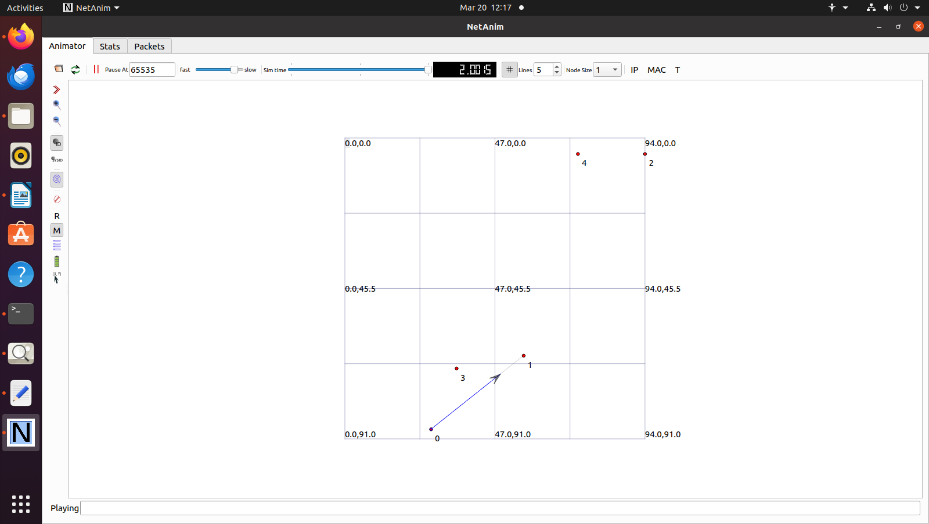
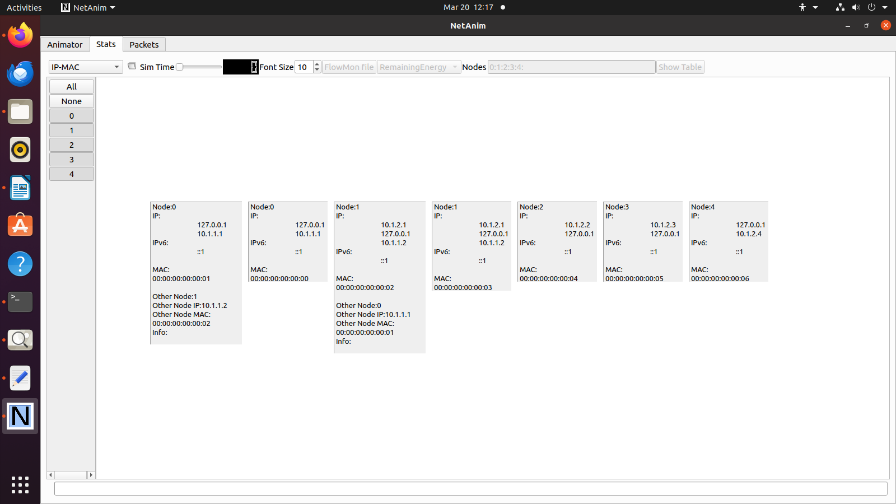
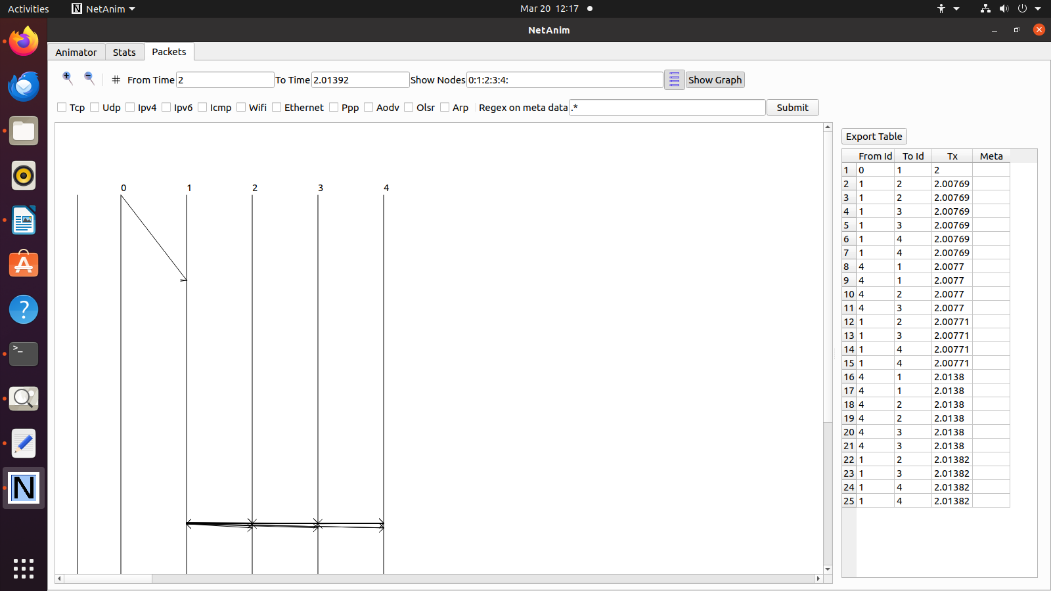
NS\_LOG\_INFO ("Run Simulation.");

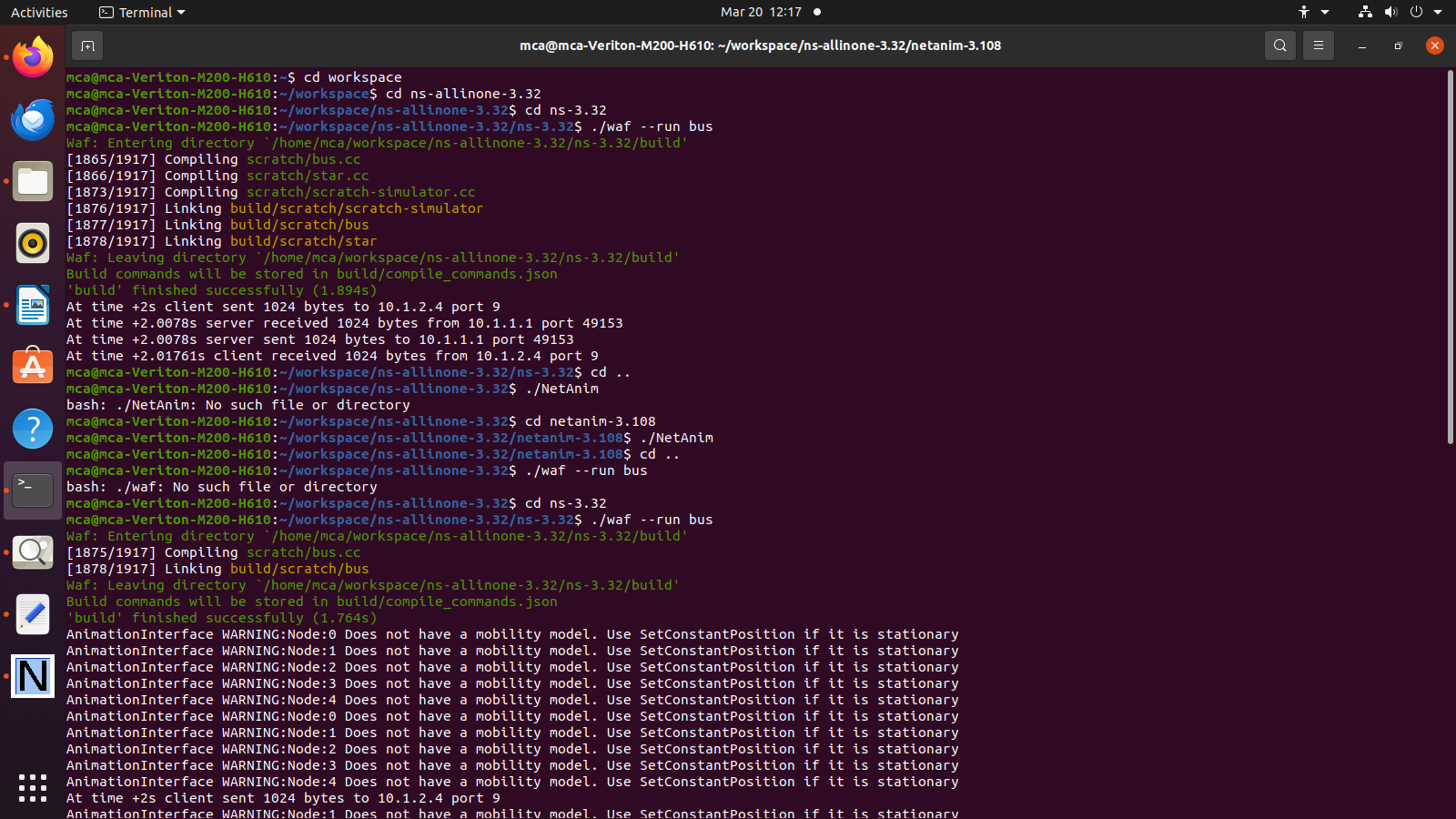
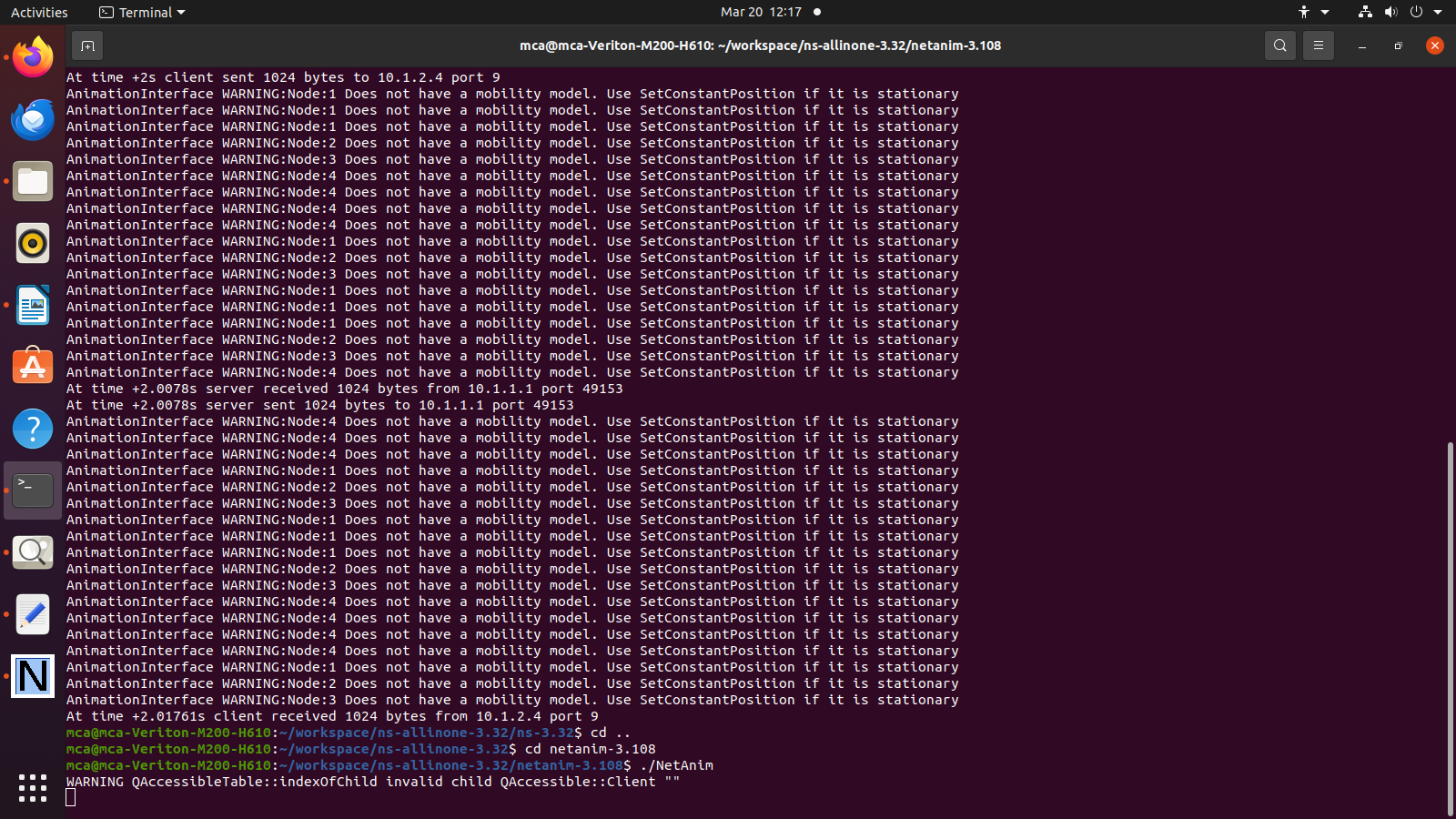
Simulator::Run ();

Simulator::Destroy ();

return 0;

}

**5. Output: -**

**** **6. Conclusion: -**

Successfully implemented Bus topology using NS3.