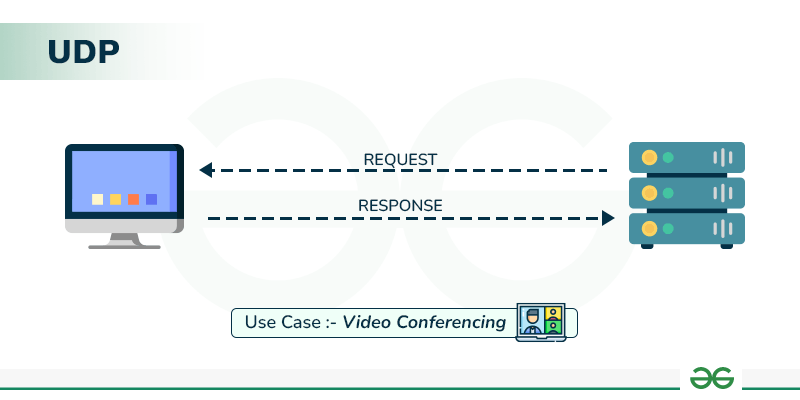
**Practical No 9**

**Aim: -** WAP to simulate UDP with one client.

**Objective: -**

* Understanding of UDP Protocol
* Implementation of UDP server with one client using NS3

**Theory: -**

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection before data transfer. The UDP helps to establish low-latency and loss-tolerating connections over the network. The UDP enables process-to-process communication.

**Advantages of UDP Protocol:**

* Speed: UDP is faster than TCP because it does not have the overhead of establishing a connection and ensuring reliable data delivery.
* Lower latency: Since there is no connection establishment, there is lower latency and faster response time.
* Simplicity: UDP has a simpler protocol design than TCP, making it easier to implement and manage.
* Broadcast support: UDP supports broadcasting to multiple recipients, making it useful for applications such as video streaming and online gaming.

**Disadvantages of UDP Protocol:**

* No reliability: UDP does not guarantee delivery of packets or order of delivery, which can lead to missing or duplicate data.
* No congestion control: UDP does not have congestion control, which means that it can send packets at a rate that can cause network congestion.
* No flow control: UDP does not have flow control, which means that it can overwhelm the receiver with packets that it cannot handle.

**Program: -**

#include <fstream>

#include "ns3/core-module.h"

#include "ns3/csma-module.h"

#include "ns3/applications-module.h"

#include "ns3/internet-module.h"

//netAnimation

#include "ns3/netanim-module.h"

#include "ns3/mobility-module.h"

using namespace ns3;

NS\_LOG\_COMPONENT\_DEFINE ("UdpClientServerExample");

int

main (int argc, char \*argv[]){

//

// Enable logging for UdpClient and

//

LogComponentEnable ("UdpClient", LOG\_LEVEL\_INFO);

LogComponentEnable ("UdpServer", LOG\_LEVEL\_INFO);

bool useV6 = false;

Address serverAddress;

CommandLine cmd (\_\_FILE\_\_);

cmd.AddValue ("useIpv6","Use Ipv6", useV6);

cmd.Parse (argc, argv);

//

// Explicitly create the nodes required by the topology (shown above).

//

NS\_LOG\_INFO ("Create nodes.");

NodeContainer n;

n.Create (2);

InternetStackHelper internet;

internet.Install (n);

NS\_LOG\_INFO ("Create channels.");

//

// Explicitly create the channels required by the topology (shown above).

//

CsmaHelper csma;

csma.SetChannelAttribute ("DataRate", DataRateValue (DataRate (5000000)));

csma.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));

csma.SetDeviceAttribute ("Mtu", UintegerValue (1400));

NetDeviceContainer d = csma.Install (n);

//

// We've got the "hardware" in place. Now we need to add IP addresses.

//

NS\_LOG\_INFO ("Assign IP Addresses.");

if (useV6 == false){

Ipv4AddressHelper ipv4;

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

Ipv4InterfaceContainer i = ipv4.Assign (d);

serverAddress = Address (i.GetAddress (1));

}

else{

Ipv6AddressHelper ipv6;

ipv6.SetBase ("2001:0000:f00d:cafe::", Ipv6Prefix (64));

Ipv6InterfaceContainer i6 = ipv6.Assign (d);

serverAddress = Address(i6.GetAddress (1,1));

}

NS\_LOG\_INFO ("Create Applications.");

//

// Create one udpServer applications on node one.

//

uint16\_t port = 4000;

UdpServerHelper server (port);

ApplicationContainer apps = server.Install (n.Get (1));

apps.Start (Seconds (1.0));

apps.Stop (Seconds (10.0));

//

// Create one UdpClient application to send UDP datagrams from node zero to

// node one.

//

uint32\_t MaxPacketSize = 1024;

Time interPacketInterval = Seconds (0.05);

uint32\_t maxPacketCount = 320;

UdpClientHelper client (serverAddress, port);

client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));

client.SetAttribute ("Interval", TimeValue (interPacketInterval));

client.SetAttribute ("PacketSize", UintegerValue (MaxPacketSize));

apps = client.Install (n.Get (0));

apps.Start (Seconds (2.0));

apps.Stop (Seconds (10.0));

AnimationInterface anim("udp-cs.xml");

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

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

anim.EnablePacketMetadata(true);

csma.EnablePcapAll("udp-cs1.xml");

//

// Now, do the actual simulation.

//

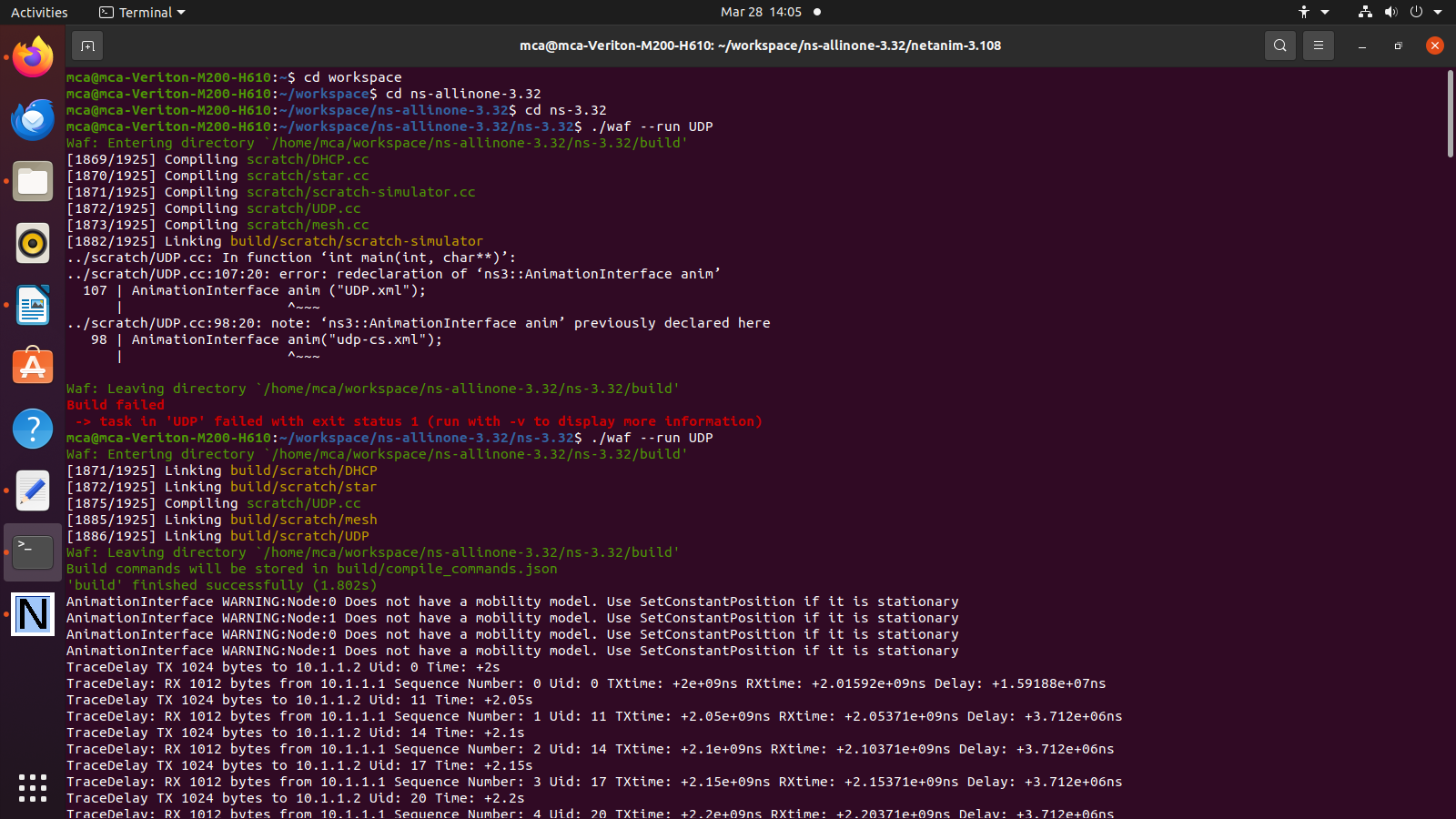
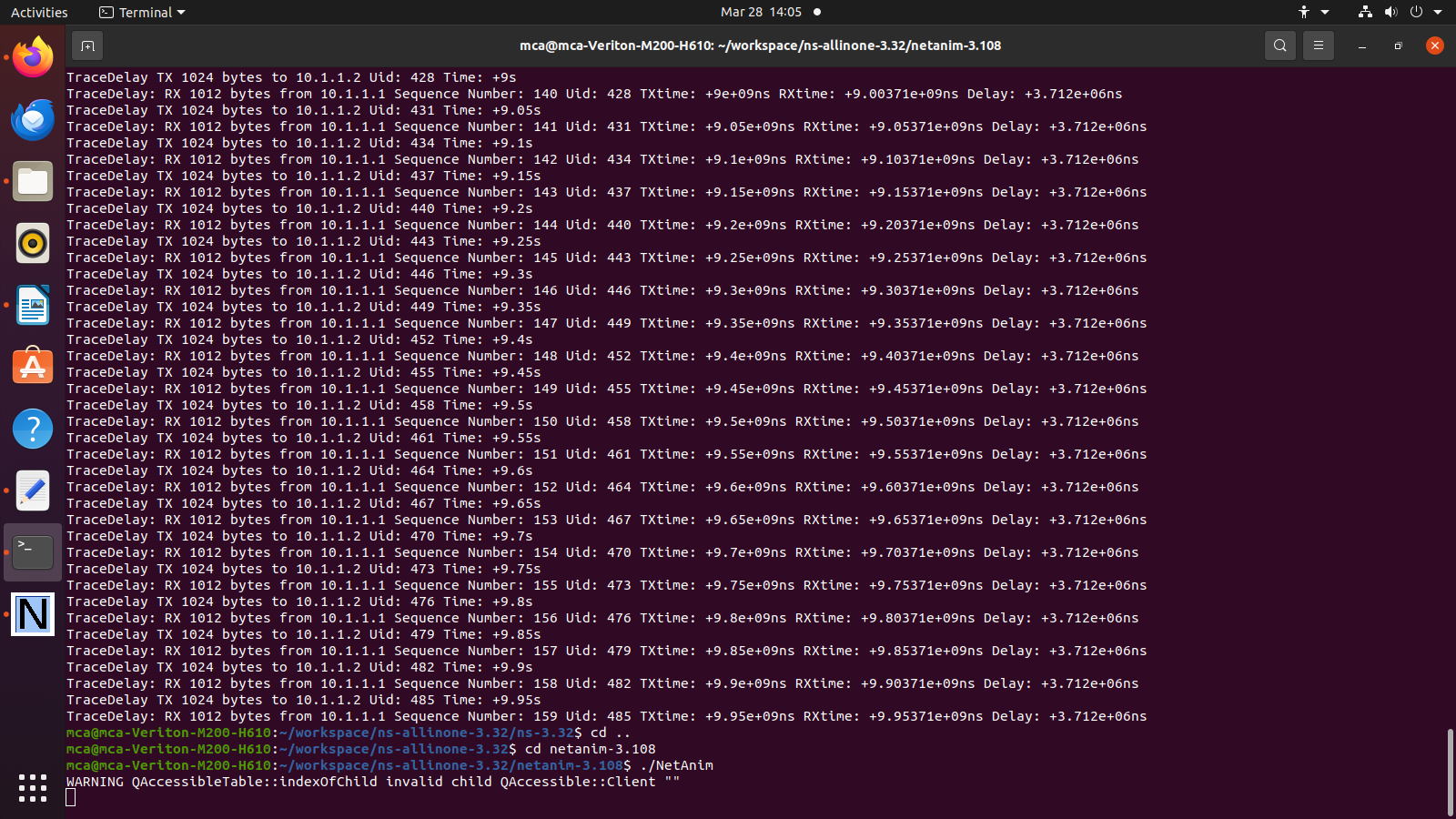
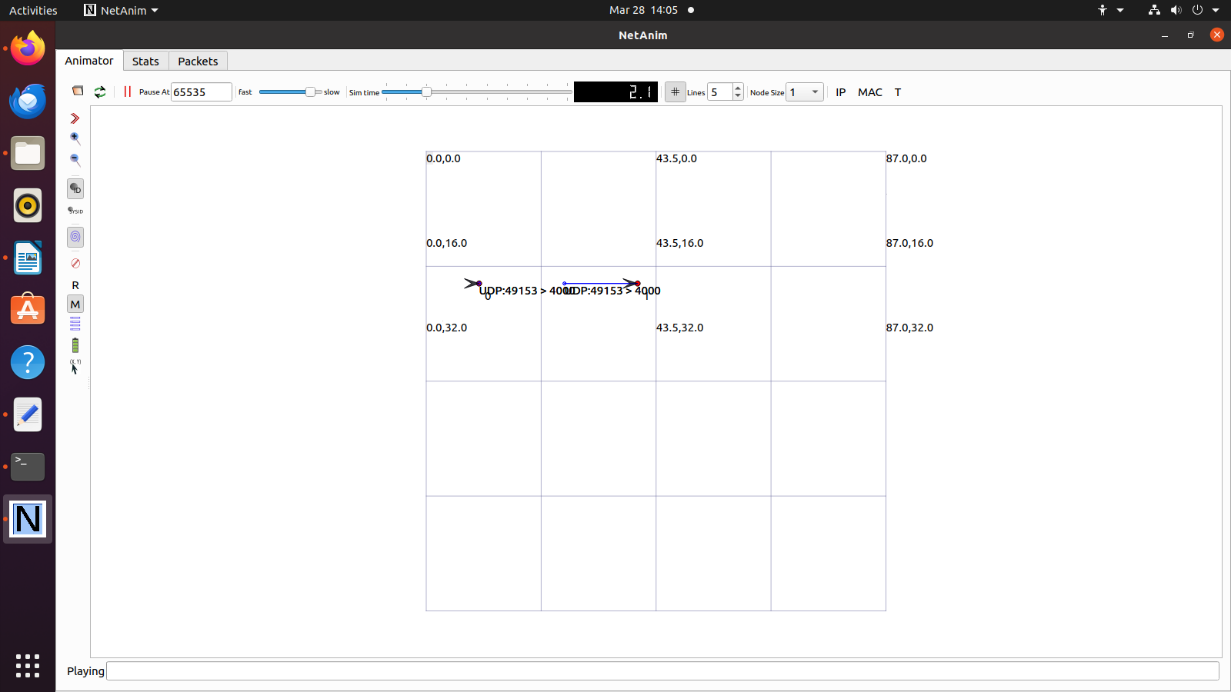
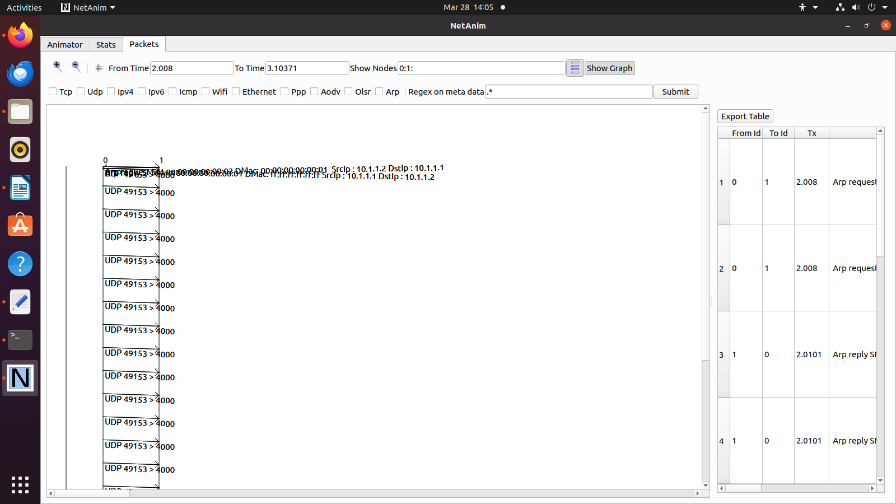
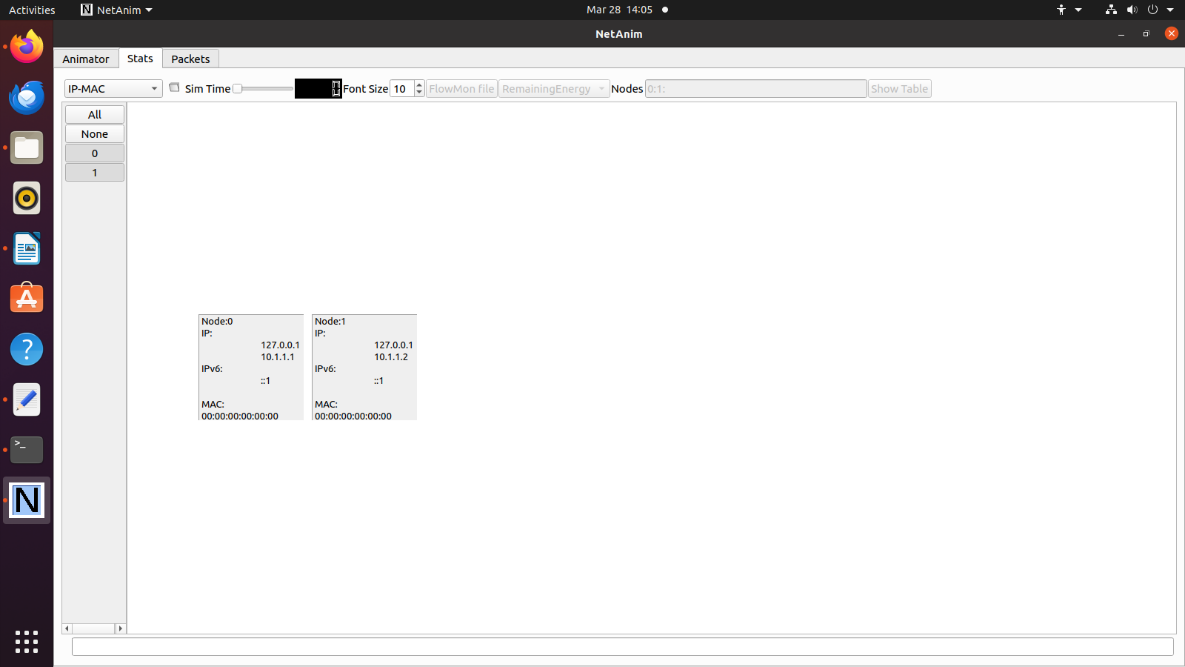
NS\_LOG\_INFO ("Run Simulation.");

Simulator::Run ();

Simulator::Destroy ();

NS\_LOG\_INFO ("Done.");

}

**Output: -**

**Conclusion: -**

Successfully implemented UDP server with one client using NS3.