**Practical No.6**

**Code:**

/\* -\*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -\*- \*/

#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/internet-apps-module.h"

#include "ns3/csma-module.h"

#include "ns3/ipv4-global-routing-helper.h"

using namespace ns3;

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

int

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

{

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

cmd.Parse (argc, argv);

Time::SetResolution (Time::NS);

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

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

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

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

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

NodeContainer nodes;

NodeContainer router;

nodes.Create(3);

router.Create(2);

NodeContainer net(nodes,router);

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

CsmaHelper csma;

csma.SetChannelAttribute ("DataRate", StringValue ("5Mbps"));

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

csma.SetDeviceAttribute ("Mtu", UintegerValue(1500));

NetDeviceContainer devNet=csma.Install(net);

NodeContainer p2pNodes;

p2pNodes.Add(net.Get(4));

p2pNodes.Create(1);

PointToPointHelper pointToPoint;

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

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

NetDeviceContainer p2pDevices;

p2pDevices=pointToPoint.Install(p2pNodes);

InternetStackHelper tcpip;

tcpip.Install(nodes);

tcpip.Install(router);

tcpip.Install(p2pNodes.Get(1));

Ipv4AddressHelper address;

address.SetBase ("172.30.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;

p2pInterfaces = address.Assign (p2pDevices);

NS\_LOG\_INFO("Setup the Ip address and create DHCP applications.");

DhcpHelper dhcpHelper;

Ipv4InterfaceContainer

fixedNodes=dhcpHelper.InstallFixedAddress(devNet.Get(4),Ipv4Address("172.30.0.17"),Ipv4Mask

("/24"));

fixedNodes.Get(0).first->SetAttribute("IpForward",BooleanValue(true));

Ipv4GlobalRoutingHelper::PopulateRoutingTables();

//DHCP server

ApplicationContainer

dhcpServerApp=dhcpHelper.InstallDhcpServer(devNet.Get(3),Ipv4Address("172.30.0.12"),Ipv4Address("172.30.0.0"),

Ipv4Mask("/24"),Ipv4Address("172.30.0.10"),Ipv4Address("172.30.0.15"),Ipv4Address("172.30.0.17"));

dhcpServerApp.Start(Seconds(0.0));

dhcpServerApp.Stop(Seconds(20.0));

//DHCP clients

NetDeviceContainer dhcpClientNetDevs;

dhcpClientNetDevs.Add(devNet.Get(0));

dhcpClientNetDevs.Add(devNet.Get(1));

dhcpClientNetDevs.Add(devNet.Get(2));

ApplicationContainer dhcpClients=dhcpHelper.InstallDhcpClient(dhcpClientNetDevs);

dhcpClients.Start(Seconds(1.0));

dhcpClients.Stop(Seconds(20.0));

UdpEchoServerHelper echoServer(9);

ApplicationContainer serverApps=echoServer.Install(p2pNodes.Get(1));

serverApps.Start(Seconds(0.0));

serverApps.Stop(Seconds(20.0));

UdpEchoClientHelper echoClient(p2pInterfaces.GetAddress(1),9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (3));

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

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

ApplicationContainer clientApps=echoClient.Install(nodes.Get(1));

clientApps.Start(Seconds(10.0));

clientApps.Stop(Seconds(20.0));

Simulator::Stop(Seconds(20.0)+Seconds(10.0));

csma.EnablePcapAll("dhcp-csma");

pointToPoint.EnablePcapAll("dhcp-p2p");

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

Simulator::Run();

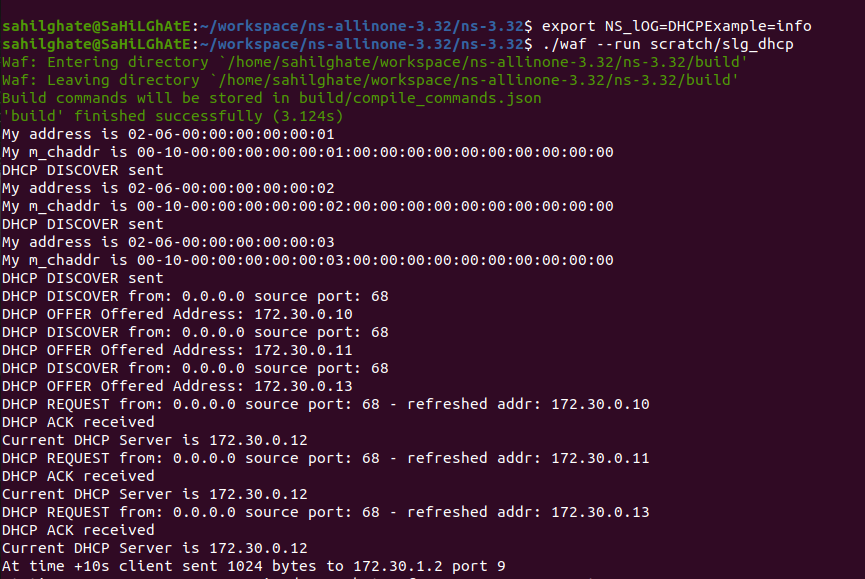
Simulator::Destroy();

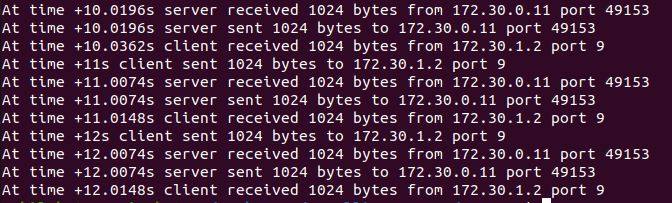
NS\_LOG\_INFO("Done");

return 0;

}

**Output:**

****

****