**Practical No.4**

**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/point-to-point-layout-module.h"

using namespace ns3;

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

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

{

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

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

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.");

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));

OnOffHelper onOffHelper("ns3::TcpSocketFactory",Address());

onOffHelper.SetAttribute("OnTime",StringValue("ns3::ConstantRandomVariable[Constant=1]"));

onOffHelper.SetAttribute("OffTime",StringValue("ns3::ConstantRandomVariable[Constant=0]"));

ApplicationContainer spokesApps;

for(uint32\_t i=0;i<star.SpokeCount();++i)

{

AddressValue remoteAddress(InetSocketAddress(star.GetHubIpv4Address(i),port));

onOffHelper.SetAttribute("Remote",remoteAddress);

spokesApps.Add(onOffHelper.Install(star.GetSpokeNode(i)));

}

spokesApps.Start(Seconds(1.0));

spokesApps.Stop(Seconds(10.0));

NS\_LOG\_INFO("Enable static global routing.");

Ipv4GlobalRoutingHelper::PopulateRoutingTables();

pointTopoint.EnablePcapAll("star");

star.BoundingBox(1,1,100,100);

AnimationInterface anim("mystar.xml");

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

Simulator::Run();

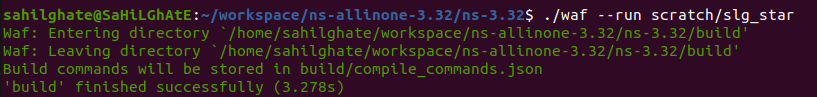
Simulator::Destroy();

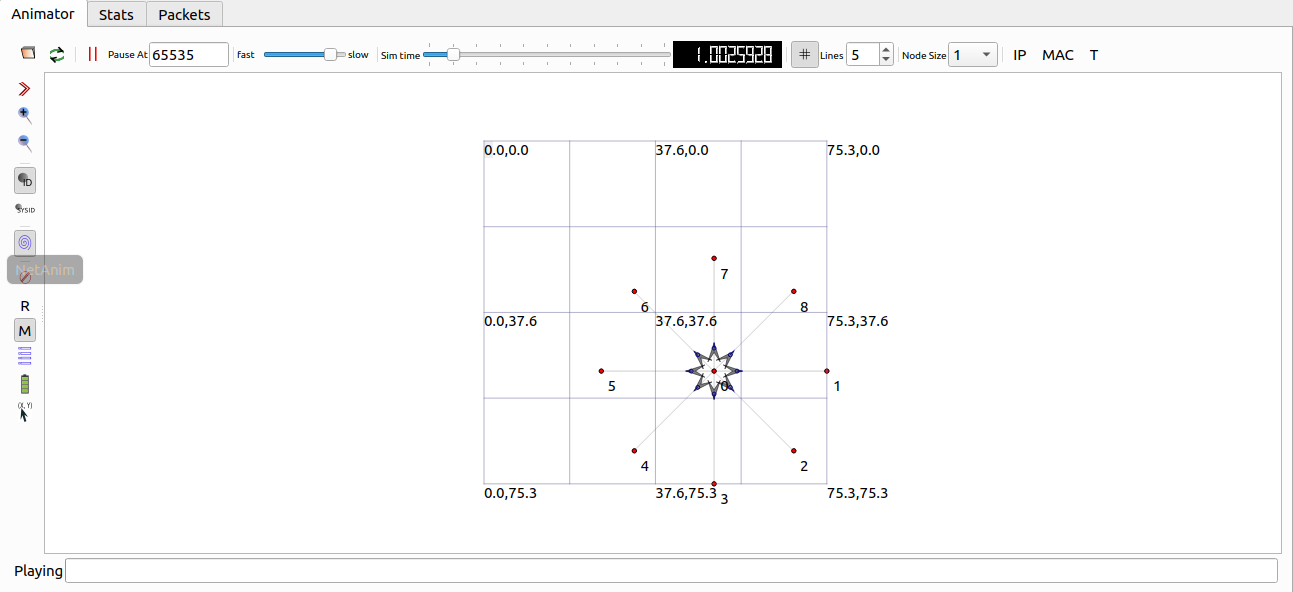
NS\_LOG\_INFO("Done.");

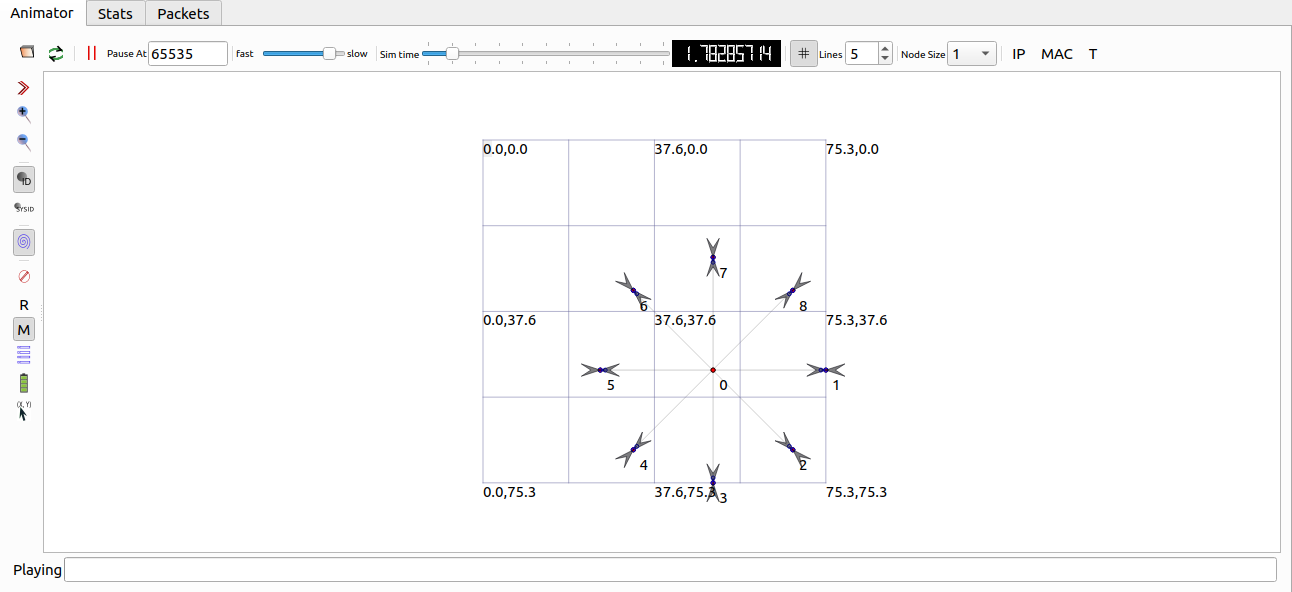
return 0;

}

**Output:**

****

****

****