**NWL MINI PROJECT**

**GROUP MEMBERS: 18 - GAURI DUDAVADKAR**

**01 - ABHAY ADHIKARI**

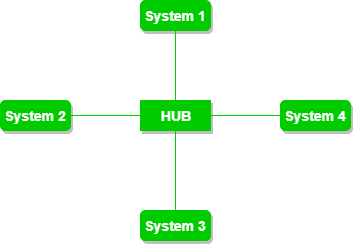
**AIM:** To write a program to simulate star topology (16 NODES) .

**THEORY:**

**Understanding Network Topology:**

* Network topology is the arrangement of the elements (links, nodes, etc.) of a communication network. Network topology can be used to define or describe the arrangement of various types of telecommunication networks, including command and control radio networks, industrial field buses and computer networks.
* Network topology is the topological structure of a network and may be depicted physically or logically. It is an application of graph theory wherein communicating devices are modelled as nodes and the connections between the devices are modelled as links or lines between the nodes. Physical topology is the placement of the various components of a network (e.g., device location and cable installation), while logical topology illustrates how data flows within a network. Distances between nodes, physical interconnections, transmission rates, or signal types may differ between two different networks, yet their logical topologies may be identical. A network’s physical topology is a particular concern of the physical layer of the OSI model.
* Examples of network topologies are found in local area networks (LAN), a common computer network installation. Any given node in the LAN has one or more physical links to other devices in the network; graphically mapping these links results in a geometric shape that can be used to describe the physical topology of the network. A wide variety of physical topologies have been used in LANs, including ring, bus, mesh and star. Conversely, mapping the data flow between the components determines the logical topology of the network. In comparison, Controller Area Networks, common in vehicles, are primarily distributed control system networks of one or more controllers interconnected with sensors and actuators over, invariably, a physical bus topology.
* A network consists of multiple computers connected using some interface. Each has one or more interface devices such as a Network Interface Card (NIC) and a serial device for PPP networking. Each computer is supported by network software that provides the server or client functionality. The hardware used to transmit data across the network is called the media. It may include copper cable, fiber optic, or wireless transmission. The standard cabling used for this document is the 10Base-T category 5 Ethernet cable. It is twisted copper cabling, which appears at the surface to look similar to TV coaxial cable. It is terminated on each end by a connector that looks much like a phone connector. Its maximum segment length is 100 meters.

**Star Topology:**

* Star topology is a network topology where each individual piece of a network is attached to a central node (often called a hub or switch). The attachment of these network pieces to the central component is visually represented in a form similar to a star.
* Star topology is also known as a star network.
* Star topologies are either active or passive networks, depending on the following:
* If the central node performs processes, such as data amplification or regeneration
* If the network actively controls data transit
* If the network requires electrical power sources.
* In star topology, all the devices are connected to a single hub through a cable. This hub is the central node and all other nodes are connected to the central node. The hub can be passive in nature i.e., not an intelligent hub such as broadcasting devices, at the same time the hub can be intelligent known as an active hub. Active hubs have repeaters in them.
* **Advantages of this topology :**
* If N devices are connected to each other in a star topology, then the number of cables required to connect them is N. So, it is easy to set up.
* Each device requires only 1 port i.e. to connect to the hub, therefore the total number of ports required is N.
* **Problems with this topology :**
* If the concentrator (hub) on which the whole topology relies fails, the whole system will crash down.
* The cost of installation is high.
* Performance is based on the single concentrator i.e. hub.

**Code and Output:**

#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"

using namespace ns3;

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

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"));

//

// Default number of nodes in the star. Overridable by command line argument.

//

uint32\_t nSpokes = 16;

std::string animFile = "star-animation.xml";

uint8\_t useIpv6 = 0;

Ipv6Address ipv6AddressBase = Ipv6Address("2001::");

Ipv6Prefix ipv6AddressPrefix = Ipv6Prefix(64);

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

cmd.AddValue ("nSpokes", "Number of spoke nodes to place in the star", nSpokes);

cmd.AddValue ("animFile", "File Name for Animation Output", animFile);

cmd.AddValue ("useIpv6", "use Ipv6", useIpv6);

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.");

if (useIpv6 == 0)

{

star.AssignIpv4Addresses (Ipv4AddressHelper ("10.1.1.0", "255.255.255.0"));

}

else

{

star.AssignIpv6Addresses (ipv6AddressBase, ipv6AddressPrefix);

}

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

//

// Create a packet sink on the star "hub" to receive packets.

//

uint16\_t port = 50000;

Address hubLocalAddress;

if (useIpv6 == 0)

{

hubLocalAddress = InetSocketAddress (Ipv4Address::GetAny (), port);

}

else

{

hubLocalAddress = Inet6SocketAddress (Ipv6Address::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)

{

AddressValue remoteAddress;

if (useIpv6 == 0)

{

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

}

else

{

remoteAddress = AddressValue(Inet6SocketAddress (star.GetHubIpv6Address (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.

//

if (useIpv6 == 0)

{

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

}

// Set the bounding box for animation

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

// Create the animation object and configure for specified output

AnimationInterface anim (animFile);

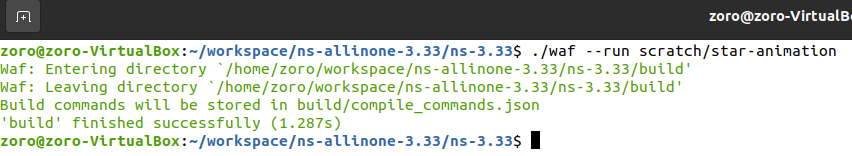
NS\_LOG\_INFO ("Run Simulation.");

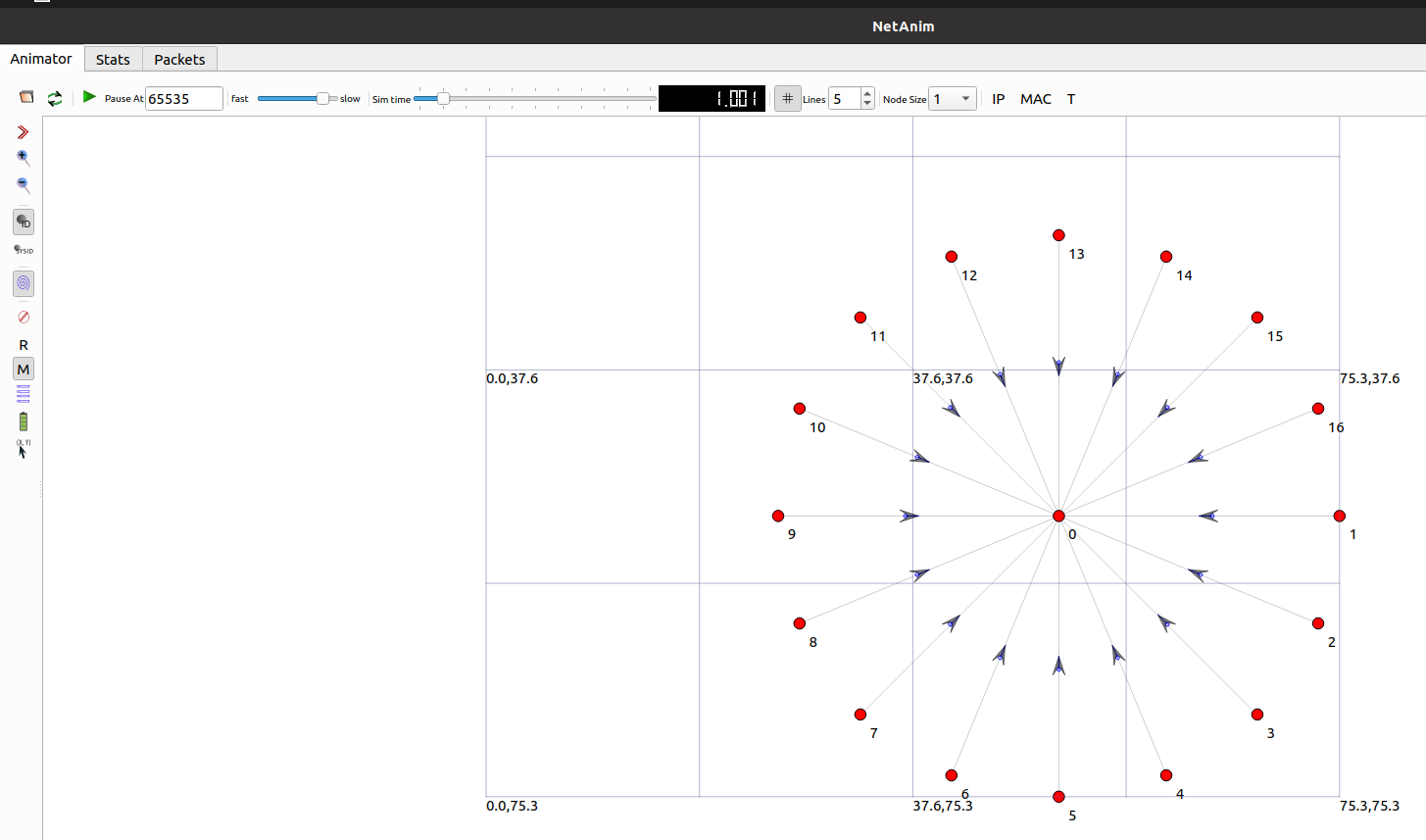
Simulator::Run ();

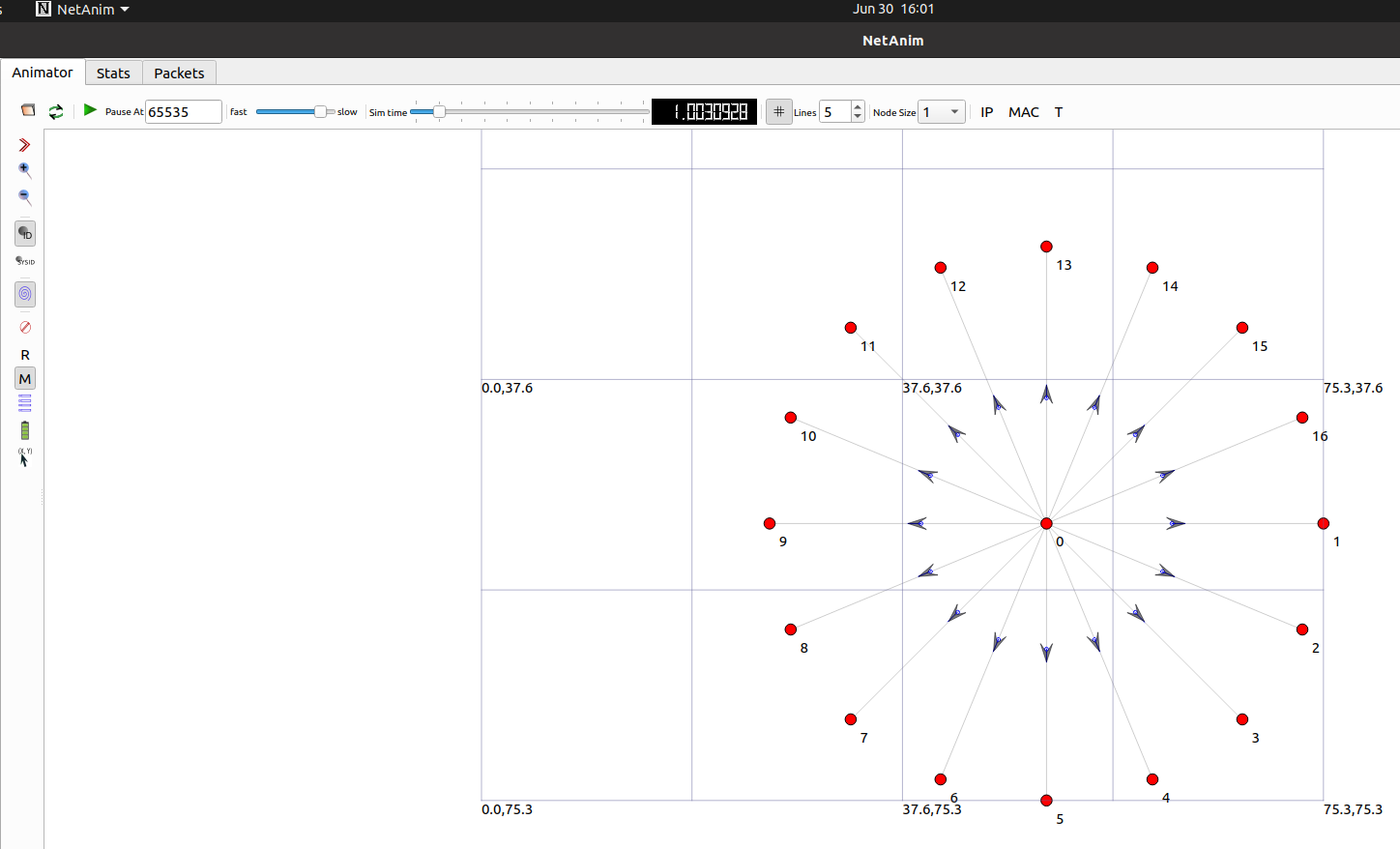
Simulator::Destroy ();

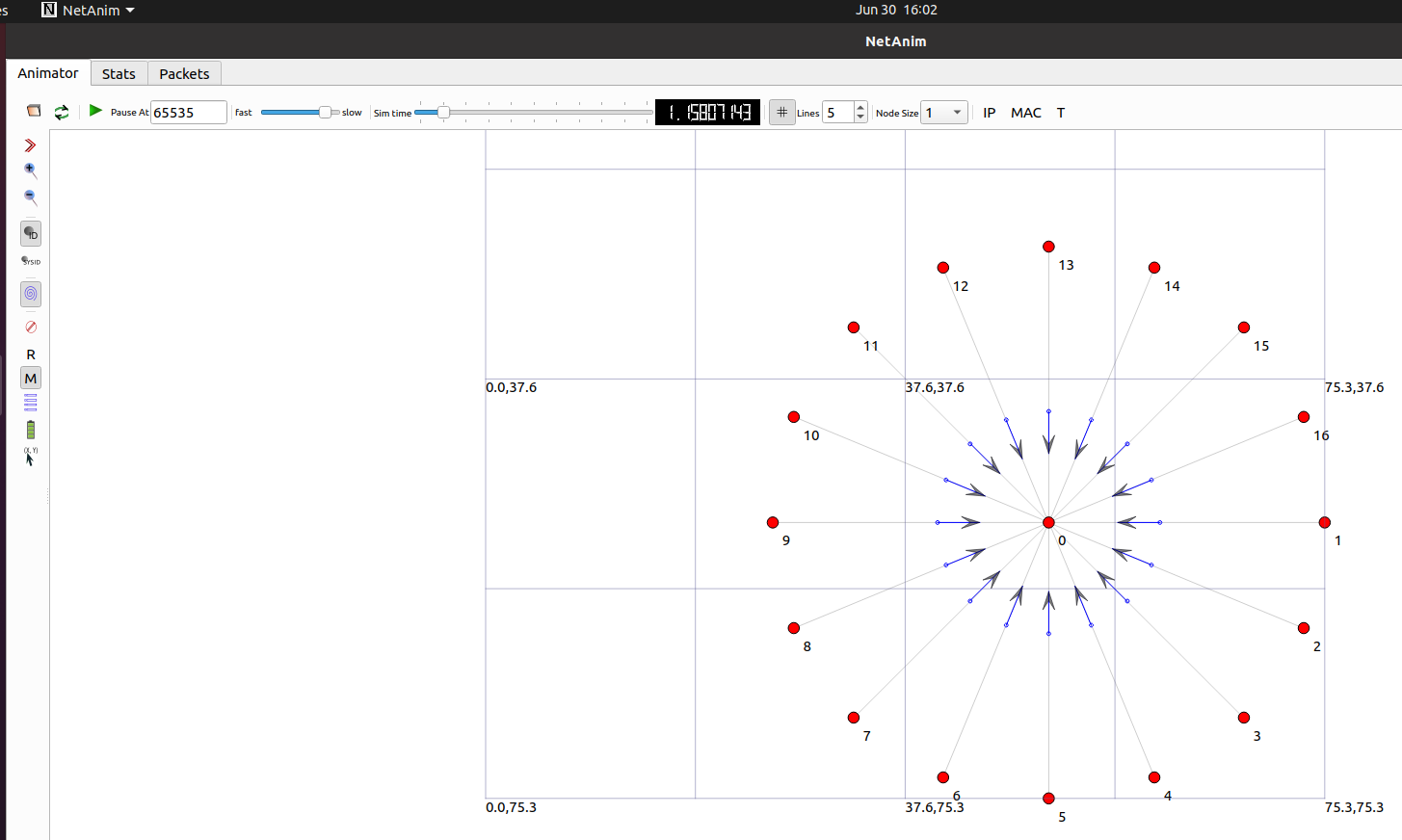
NS\_LOG\_INFO ("Done.");

return 0;

}







**Conclusion:**

Program to simulate star topology (with 16 nodes) executed successfully.