**Aim: Program to simulate star topology**

**Code:** #include "ns3/applications-module.h" #include "ns3/core-module.h" #include "ns3/internet-module.h" #include "ns3/netanim-module.h" #include "ns3/network-module.h"

#include "ns3/point-to-point-layout-module.h" #include "ns3/point-to-point-module.h"

// Network topology (default)

|  |  |  |
| --- | --- | --- |
| // //  // | n2 n3 n4  \ | / | .  . |
| // | \|/ | . |
| // | n1--- n0---n5 | . |
| // | /|\ | . |
| // | / | \ | . |
| // | n8 n7 n6 | . |
| // | | |

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 spokeApps;

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

{

AddressValue remoteAddress(InetSocketAddress(star.GetHubIpv4Address(i), port)); onOffHelper.SetAttribute("Remote", remoteAddress); spokeApps.Add(onOffHelper.Install(star.GetSpokeNode(i)));

}

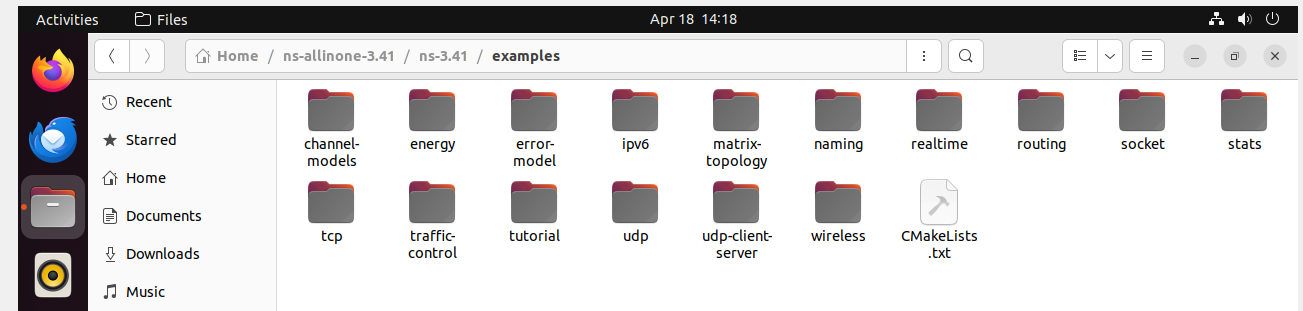
spokeApps.Start(Seconds(1.0)); spokeApps.Stop(Seconds(10.0)); NS\_LOG\_INFO("Enable static global routing."); Ipv4GlobalRoutingHelper::PopulateRoutingTables(); NS\_LOG\_INFO("Enable pcap tracing."); pointToPoint.EnablePcapAll("star"); NS\_LOG\_INFO("Run Simulation."); Simulator::Run();

Simulator::Destroy(); NS\_LOG\_INFO("Done.");

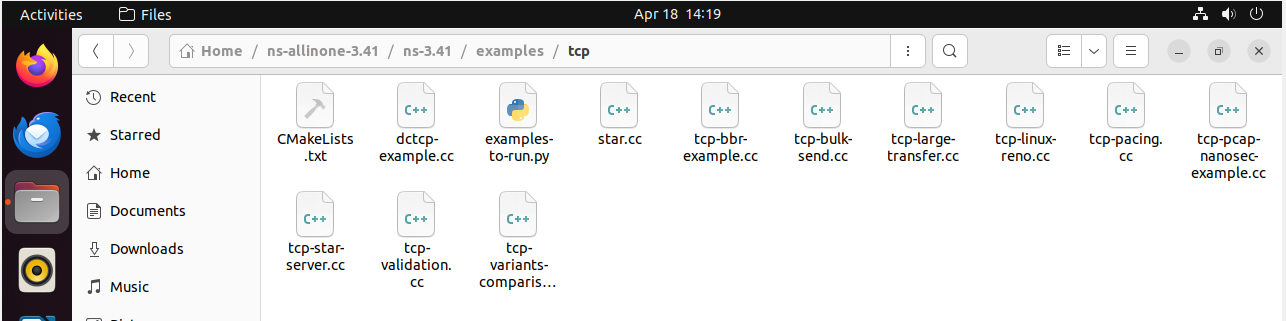
return 0;

}

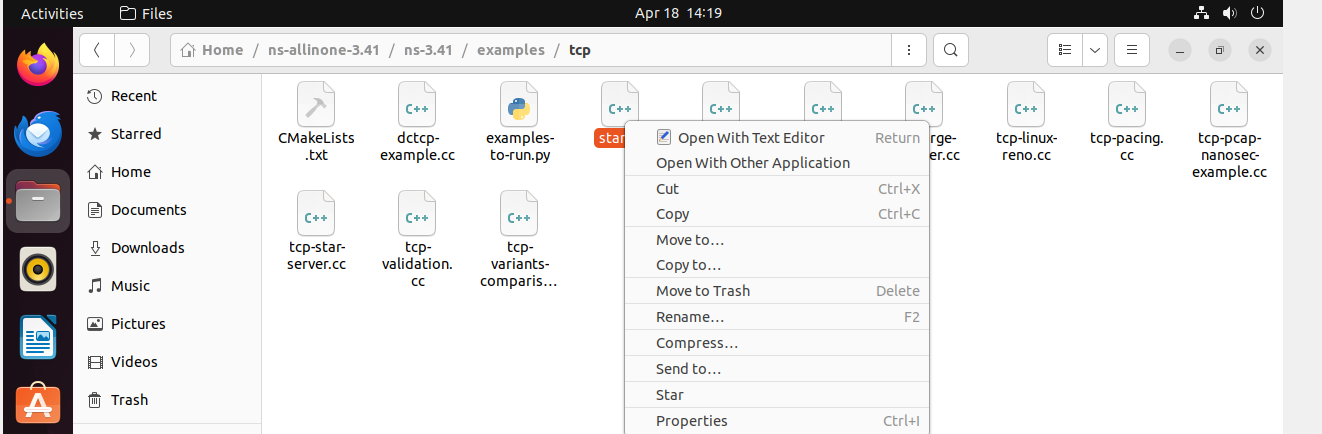
**Output:** Step1: open example folder and open the tcp folder



Step2: copy the  star.cc file

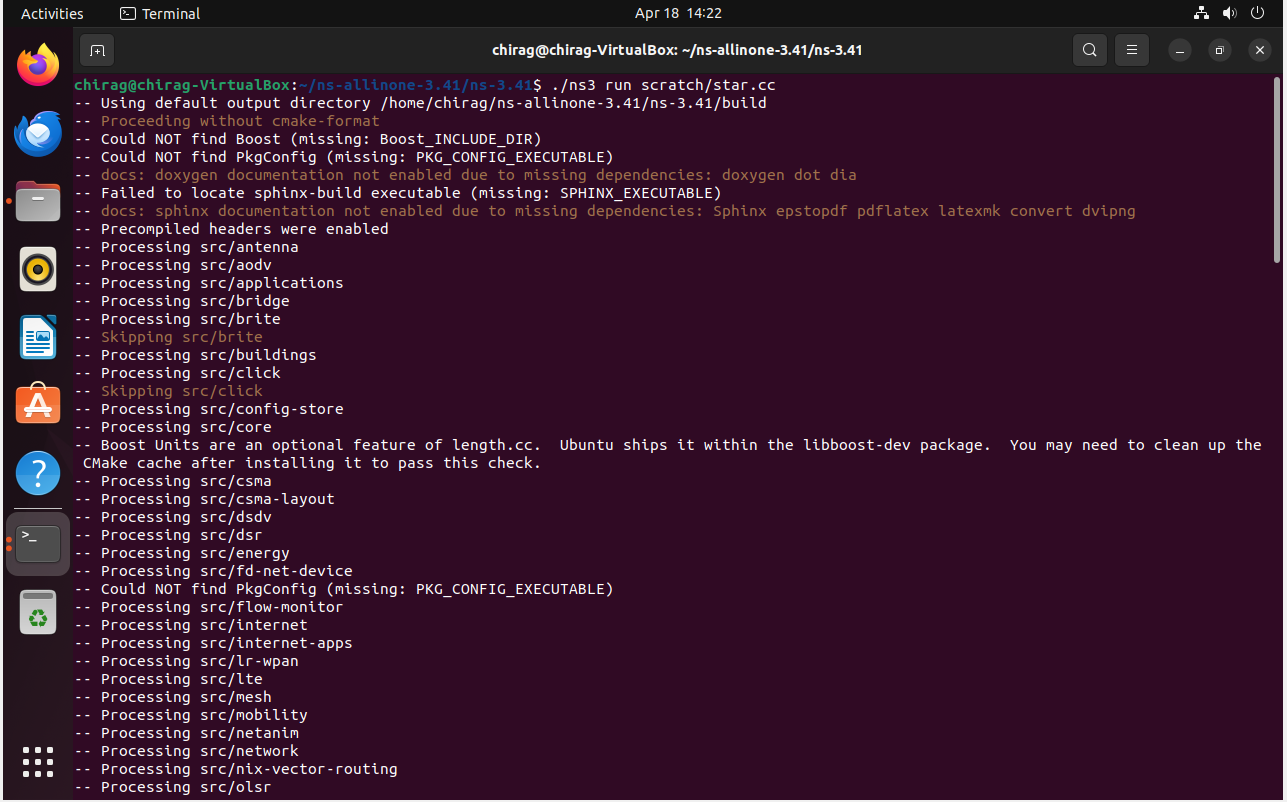


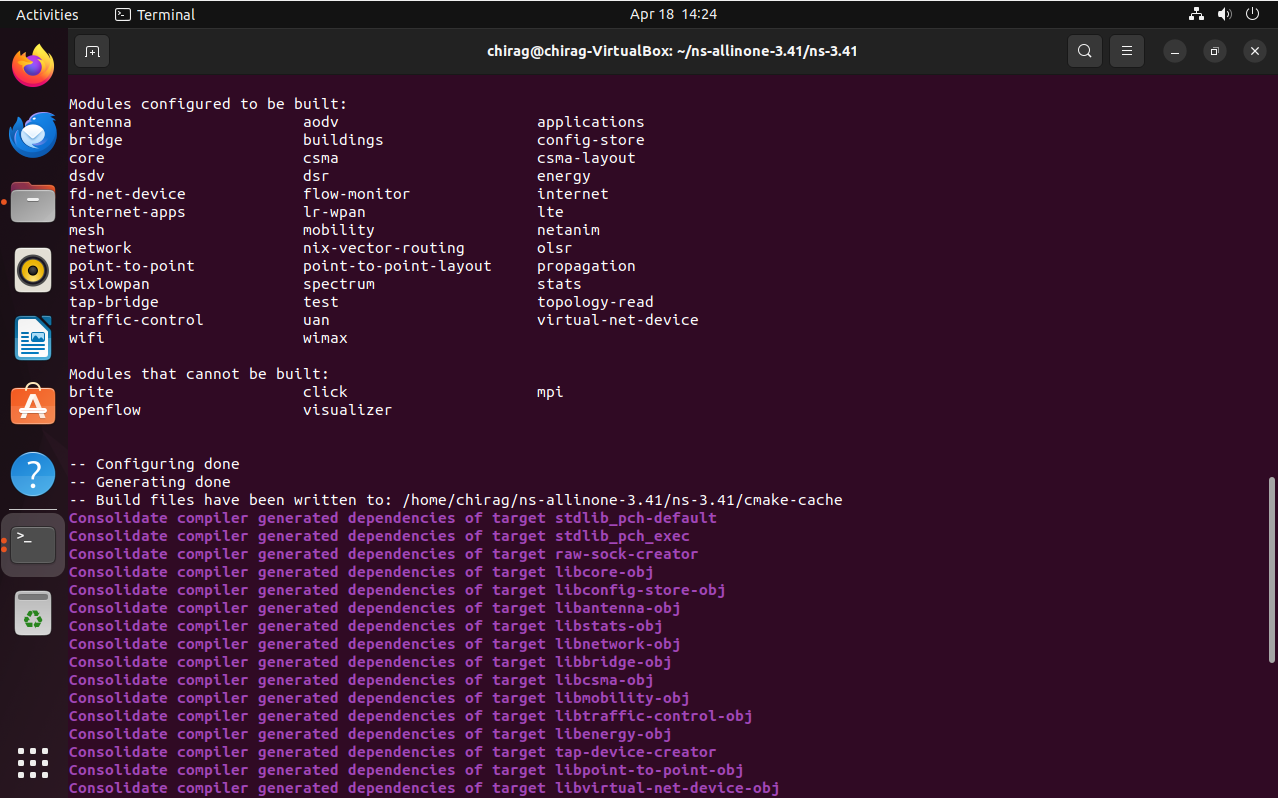
Step4: come back to ns-3.41 and open scratch folder



Step5: Paste the star.cc file here

Step6: Back to ns-3.41 file click on 3 dot it gives the option for open terminal run the command ./ns3 run scratch/star





Step7: ls showing the list directory

