|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Program 2** | | | | |
| **Problem Statement** | | | |
| Write a NS3 program to connect two nodes with a point to pint link, which have unique interface. Analyze the network performance using UDP client server. | | | | |
| **2** | **Student Learning Outcomes** | | | |
| After successful completion of this lab, the student’s shall be able to   * Understand to establish point to point network. * Create an interface to point to point link * Anylize the Network performance usibg client server | | | | |
| **3** | **Design of the Program** | | | |
| **3.1** | **Theory** | | | |
| **Point to Point Network**  http://www.highteck.net/images/167-Datalink-point-to-point.jpg  Fig:Point to Point Network  This type of network consists of many connections between individual pairs of machines. To go from the source to the destination, a packet of information on this type of network may have to first visit one or more intermediate machines. Often multiple routes, of different length are possible, so routing algorithms play an important role in point-to-point networks.  **Network performance**    Network Performance refers to measures of [service quality](https://en.wikipedia.org/wiki/Service_quality) of a network  **UDP: User Datagram Protocol**  The User Datagram Protocol (UDP) is the simpler of the two standard TCP/IP transport protocols. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer. | | | | |
| **3.2** | | **Coding using C++ Language** | | |
| #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"  using namespace ns3;  NS\_LOG\_COMPONENT\_DEFINE ("FirstScriptExample");  int  main (int argc, char \*argv[])  {  CommandLine cmd;  cmd.Parse (argc, argv);  Time::SetResolution (Time::NS);  LogComponentEnable ("UdpEchoClientApplication", LOG\_LEVEL\_INFO);  LogComponentEnable ("UdpEchoServerApplication", LOG\_LEVEL\_INFO);  NodeContainer nodes;  nodes.Create (2);  PointToPointHelper pointToPoint;  pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));  pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));  NetDeviceContainer devices;  devices = pointToPoint.Install (nodes);  InternetStackHelper stack;  stack.Install (nodes);  Ipv4AddressHelper address;  address.SetBase ("10.1.1.0", "255.255.255.0");  Ipv4InterfaceContainer interfaces = address.Assign (devices);  UdpEchoServerHelper echoServer (9);  ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));  serverApps.Start (Seconds (1.0));  serverApps.Stop (Seconds (10.0));  UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);  echoClient.SetAttribute ("MaxPackets", UintegerValue (1));  echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));  echoClient.SetAttribute ("PacketSize", UintegerValue (1024));  ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));  clientApps.Start (Seconds (2.0));  clientApps.Stop (Seconds (10.0));  AnimationInterface anim ("first.xml");  anim.SetConstantPosition(nodes.Get (0), 10.0, 10.0);  anim.SetConstantPosition(nodes.Get (1), 20.0, 30.0);  Simulator::Run ();  Simulator::Destroy ();  return 0;  } | | | | |
| **3.3** | | | **Expected Results** | |
|  | | | | |