Computer Networks Lab

Name: Snehil Raj

Roll No: 442/23

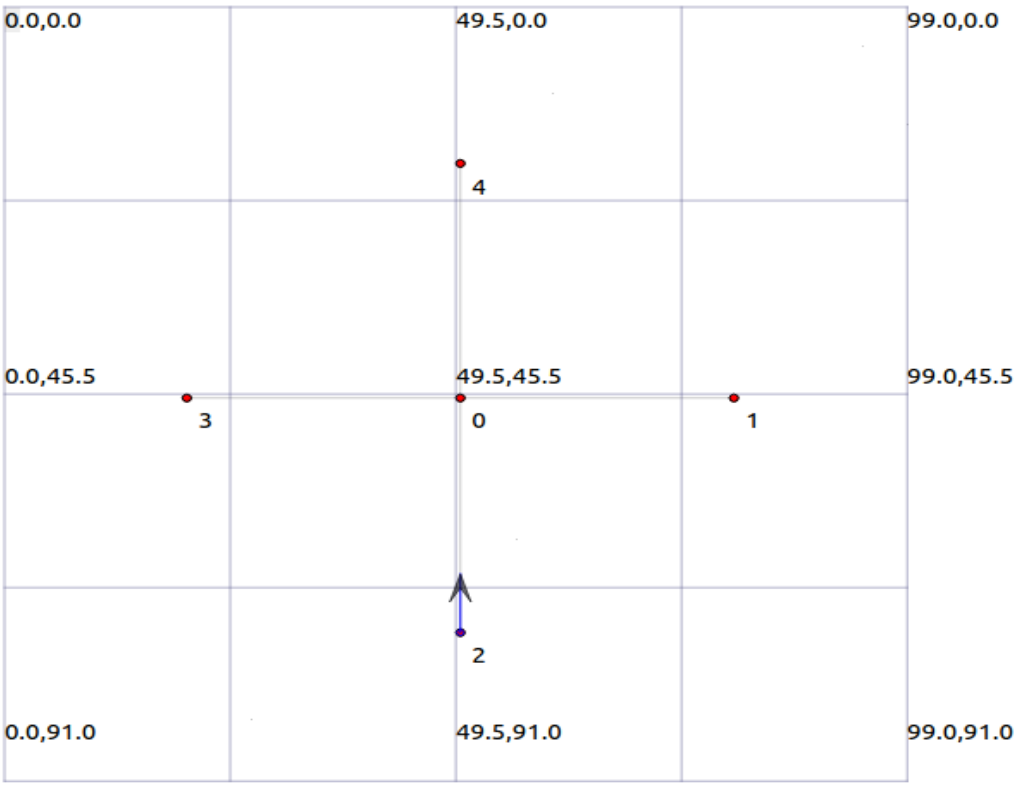
Branch: Computer Engineering

Course: B.Tech

Semester:6th

Year:3rd

Star Topology - Output



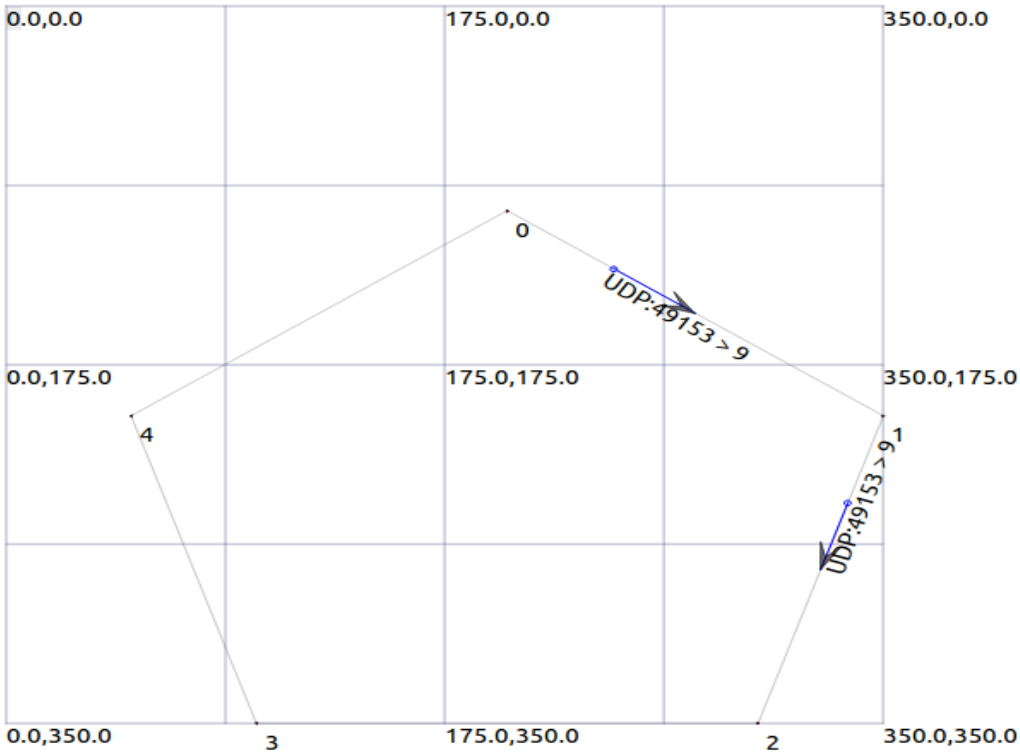

```
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$ ./ns3 run scratch/animationFirst
[0/2] Re-checking globbed directories...
ninja: no work to do.
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
At time +2s client sent 512 bytes to 10.1.1.1 port 9
At time +2.00287s server received 512 bytes from 10.1.1.2 port 49153
At time +2.00287s server sent 512 bytes to 10.1.1.2 port 49153
At time +2.00573s client received 512 bytes from 10.1.1.1 port 9
At time +3s client sent 512 bytes to 10.1.2.1 port 9
At time +3.00287s server received 512 bytes from 10.1.2.2 port 49153
At time +3.00287s server sent 512 bytes to 10.1.2.2 port 49153
At time +3.00573s client received 512 bytes from 10.1.2.1 port 9
At time +4s client sent 512 bytes to 10.1.3.1 port 9
At time +4.00287s server received 512 bytes from 10.1.3.2 port 49153
At time +4.00287s server sent 512 bytes to 10.1.3.2 port 49153
At time +4.00573s client received 512 bytes from 10.1.3.1 port 9
At time +5s client sent 512 bytes to 10.1.4.1 port 9
At time +5.00287s server received 512 bytes from 10.1.4.2 port 49153
At time +5.00287s server sent 512 bytes to 10.1.4.2 port 49153
At time +5.00573s client received 512 bytes from 10.1.4.1 port 9
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$
```

Star Topology - Code

```
#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("StarTopologyWithNetAnim");

int main(int argc, char *argv[])
{
  CommandLine cmd(__FILE__);
  cmd.Parse(argc, argv);
  Time::SetResolution(Time::NS);
  LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
  LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
  // ----------------------------
  // Create Nodes
  // ----------------------------
  NodeContainer nodes;
  nodes.Create(4);
  // Node 0 = Central Hub/Switch
  Ptr<Node> hub = nodes.Get(0);
  // ----------------------------
  // Point-to-Point Links (Star)
  // ----------------------------
  PointToPointHelper p2p;
  p2p.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
  p2p.SetChannelAttribute("Delay", StringValue("2ms"));
  NetDeviceContainer d01 = p2p.Install(hub, nodes.Get(1));
  NetDeviceContainer d02 = p2p.Install(hub, nodes.Get(2));
  NetDeviceContainer d03 = p2p.Install(hub, nodes.Get(3));
  // ----------------------------
  // Internet Stack
  // ----------------------------
  InternetStackHelper stack;
  stack.Install(nodes);
  Ipv4AddressHelper address;
  Ipv4InterfaceContainer i01, i02, i03;
  address.SetBase("10.1.1.0", "255.255.255.0");
  i01 = address.Assign(d01);
  address.SetBase("10.1.2.0", "255.255.255.0");
  i02 = address.Assign(d02);
  address.SetBase("10.1.3.0", "255.255.255.0");
  i03 = address.Assign(d03);
```

```
Ipv4GlobalRoutingHelper::PopulateRoutingTables();
  // ----------------------------
  // UDP Echo Server (Hub Node 0)
  // ----------------------------
  UdpEchoServerHelper echoServer(9);
  ApplicationContainer serverApp = echoServer.Install(hub);
  serverApp.Start(Seconds(1.0));
  serverApp.Stop(Seconds(10.0));
  // UDP Echo Clients (Leaf Nodes)
  // ----------------------------
  UdpEchoClientHelper echoClient(i01.GetAddress(0), 9);
  echoClient.SetAttribute("MaxPackets", UintegerValue(1));
  echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
  echoClient.SetAttribute("PacketSize", UintegerValue(1024));
  ApplicationContainer clientApp1 = echoClient.Install(nodes.Get(1));
  clientApp1.Start(Seconds(2.0));
  clientApp1.Stop(Seconds(10.0));
  ApplicationContainer clientApp2 = echoClient.Install(nodes.Get(2));
  clientApp2.Start(Seconds(3.0));
  clientApp2.Stop(Seconds(10.0));
  ApplicationContainer clientApp3 = echoClient.Install(nodes.Get(3));
  clientApp3.Start(Seconds(4.0));
  clientApp3.Stop(Seconds(10.0));
  // ----------------------------
  // NetAnim Visualization
  // ----------------------------
  AnimationInterface anim("star-topology.xml");
  // Position nodes (Hub center)
  anim.SetConstantPosition(nodes.Get(0), 40, 25);  // Hub
  anim.SetConstantPosition(nodes.Get(1), 10, 40);
  anim.SetConstantPosition(nodes.Get(2), 70, 40);
  anim.SetConstantPosition(nodes.Get(3), 40, 5);
  for (uint32_t i = 0; i < nodes.GetN(); i++)
  {
    std::string desc = (i == 0) ? "Hub (Server)" : "Node " + std::to_string(i);
    anim.UpdateNodeDescription(nodes.Get(i), desc);

    if (i == 0)
      anim.UpdateNodeColor(nodes.Get(i), 255, 0, 0);  // Hub = Red
    else
      anim.UpdateNodeColor(nodes.Get(i), 0, 0, 255);  // Clients = Blue
  }
  Simulator::Run();
  Simulator::Destroy();

  return 0;
}
```

Ring Topology - Output



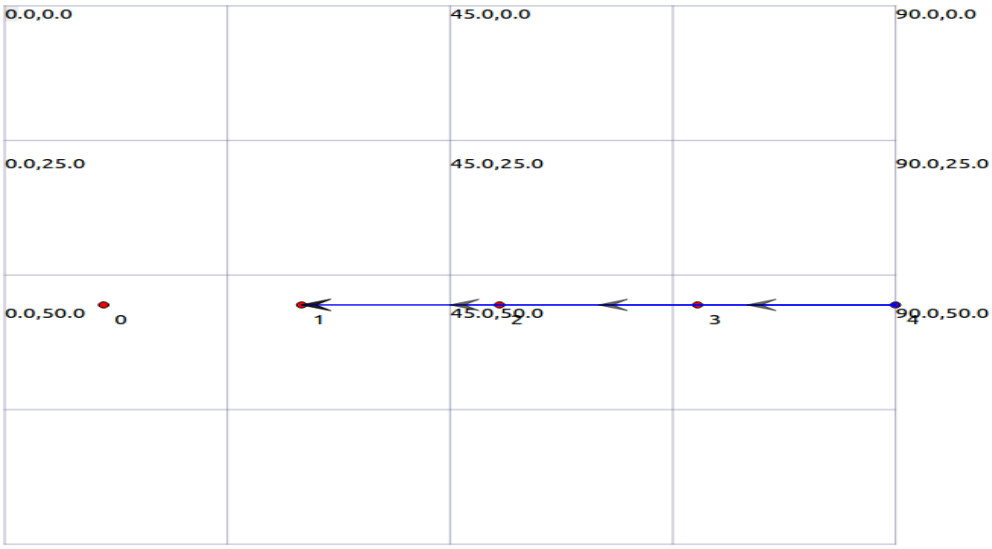

```
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$ ./ns3 run scratch/ring-topology
[0/2] Re-checking globbed directories...
[2/2] Linking CXX executable /home/itnc/ns-allinone-3.36/ns-3.36/build/scratch/ns3.36-ring-topology-default
At time +2s client sent 512 bytes to 10.1.1.2 port 9
At time +2.00243s server received 512 bytes from 10.1.1.1 port 49153
At time +2.00243s server sent 512 bytes to 10.1.1.1 port 49153
At time +2.00487s client received 512 bytes from 10.1.1.2 port 9
At time +3s client sent 512 bytes to 10.1.1.2 port 9
At time +3s client sent 512 bytes to 10.1.2.2 port 9
At time +3.00243s server received 512 bytes from 10.1.1.1 port 49153
At time +3.00243s server sent 512 bytes to 10.1.1.1 port 49153
At time +3.00243s server received 512 bytes from 10.1.2.1 port 49153
At time +3.00243s server sent 512 bytes to 10.1.2.1 port 49153
At time +3.00487s client received 512 bytes from 10.1.1.2 port 9
At time +3.00487s client received 512 bytes from 10.1.2.2 port 9
```

Ring Topology - Code

```
#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("RingTopologyWithNetAnim");
int main(int argc, char *argv[])
{
  CommandLine cmd(__FILE__);
  cmd.Parse(argc, argv);
  Time::SetResolution(Time::NS);
  LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
  LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
  // ----------------------------
  // Create Nodes
  // ----------------------------
  NodeContainer nodes;
  nodes.Create(4);
  // ----------------------------
  // Point-to-Point Links (Ring)
  // ----------------------------
  PointToPointHelper p2p;
  p2p.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
  p2p.SetChannelAttribute("Delay", StringValue("2ms"));
  NetDeviceContainer d01 = p2p.Install(nodes.Get(0), nodes.Get(1));
  NetDeviceContainer d12 = p2p.Install(nodes.Get(1), nodes.Get(2));
  NetDeviceContainer d23 = p2p.Install(nodes.Get(2), nodes.Get(3));
  NetDeviceContainer d30 = p2p.Install(nodes.Get(3), nodes.Get(0));
  // ----------------------------
  // Internet Stack
  // ----------------------------
  InternetStackHelper stack;
  stack.Install(nodes);
  Ipv4AddressHelper address;
  Ipv4InterfaceContainer i01, i12, i23, i30;
  address.SetBase("10.1.1.0", "255.255.255.0");
  i01 = address.Assign(d01);
  address.SetBase("10.1.2.0", "255.255.255.0");
  i12 = address.Assign(d12);
  address.SetBase("10.1.3.0", "255.255.255.0");
  i23 = address.Assign(d23);
  address.SetBase("10.1.4.0", "255.255.255.0");
  i30 = address.Assign(d30);
```

```
 Ipv4GlobalRoutingHelper::PopulateRoutingTables();
   // ----------------------------
   // UDP Echo Server (Node 0)
   // ----------------------------
   UdpEchoServerHelper echoServer(9);
   ApplicationContainer serverApp = echoServer.Install(nodes.Get(0));

   serverApp.Start(Seconds(1.0));
   serverApp.Stop(Seconds(10.0));
   // ----------------------------
   // UDP Echo Client (Node 2 → Node 0)
   // ----------------------------
   UdpEchoClientHelper echoClient(i23.GetAddress(1), 9);

   echoClient.SetAttribute("MaxPackets", UintegerValue(1));
   echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
   echoClient.SetAttribute("PacketSize", UintegerValue(1024));
   ApplicationContainer clientApp = echoClient.Install(nodes.Get(2));

   clientApp.Start(Seconds(2.0));
   clientApp.Stop(Seconds(10.0));

   // ----------------------------
   // NetAnim Visualization
   // ----------------------------
   AnimationInterface anim("ring-topology.xml");
   // Arrange nodes in circular (ring) layout
   anim.SetConstantPosition(nodes.Get(0), 40, 40);
   anim.SetConstantPosition(nodes.Get(1), 70, 25);
   anim.SetConstantPosition(nodes.Get(2), 40, 10);
   anim.SetConstantPosition(nodes.Get(3), 10, 25);

   for (uint32_t i = 0; i < nodes.GetN(); i++)
   {
      std::string desc = "Node " + std::to_string(i);
      anim.UpdateNodeDescription(nodes.Get(i), desc);

      if (i == 0)
         anim.UpdateNodeColor(nodes.Get(i), 255, 0, 0);  // Server = Red
      else
         anim.UpdateNodeColor(nodes.Get(i), 0, 0, 255);  // Others = Blue
   }
   Simulator::Run();
   Simulator::Destroy();

   return 0;
}
```

Bus Topology - Output



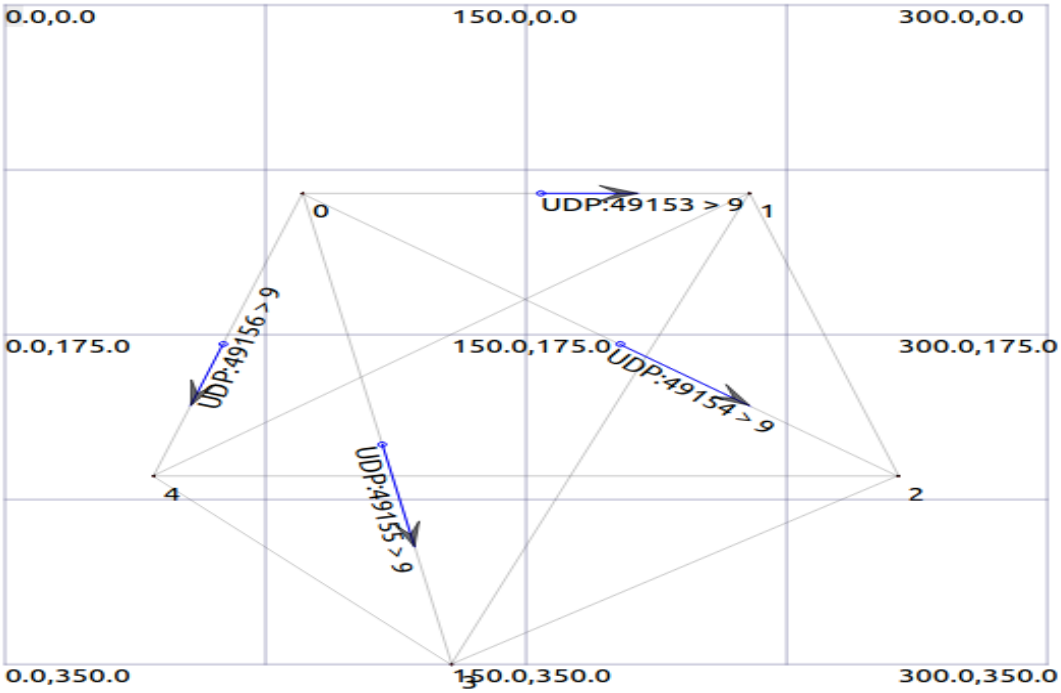

```
Exception: Couldn't find the specified program: scratch/bus-topology
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$ ./ns3 run scratch/bus-topology
[0/2] Re-checking globbed directories...
ninja: no work to do.
At time +2s client sent 1024 bytes to 10.1.1.5 port 9
At time +2.00612s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00612s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.01224s client received 1024 bytes from 10.1.1.5 port 9
At time +3s client sent 1024 bytes to 10.1.1.5 port 9
At time +3.00009s server received 1024 bytes from 10.1.1.1 port 49153
At time +3.00009s server sent 1024 bytes to 10.1.1.1 port 49153
At time +3.00019s client received 1024 bytes from 10.1.1.5 port 9
At time +4s client sent 1024 bytes to 10.1.1.5 port 9
At time +4.00009s server received 1024 bytes from 10.1.1.1 port 49153
At time +4.00009s server sent 1024 bytes to 10.1.1.1 port 49153
At time +4.00019s client received 1024 bytes from 10.1.1.5 port 9
At time +5s client sent 1024 bytes to 10.1.1.5 port 9
At time +5.00009s server received 1024 bytes from 10.1.1.1 port 49153
At time +5.00009s server sent 1024 bytes to 10.1.1.1 port 49153
At time +5.00019s client received 1024 bytes from 10.1.1.5 port 9
At time +6s client sent 1024 bytes to 10.1.1.5 port 9
At time +6.00009s server received 1024 bytes from 10.1.1.1 port 49153
At time +6.00009s server sent 1024 bytes to 10.1.1.1 port 49153
At time +6.00019s client received 1024 bytes from 10.1.1.5 port 9
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$
```

Bus Topology - Code

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE("BusTopologyWithNetAnim");
int main(int argc, char *argv[])
{
  CommandLine cmd(__FILE__);
  cmd.Parse(argc, argv);
  Time::SetResolution(Time::NS);

  LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
  LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
  // ----------------------------
  // Create Nodes (Bus Topology)
  // ----------------------------
  NodeContainer nodes;
  nodes.Create(4);
  // ----------------------------
  // CSMA Channel = Shared Bus
  // ----------------------------
  CsmaHelper csma;
  csma.SetChannelAttribute("DataRate", StringValue("10Mbps"));
  csma.SetChannelAttribute("Delay", TimeValue(NanoSeconds(6560)));

  NetDeviceContainer devices = csma.Install(nodes);
  // ----------------------------
  // Internet Stack
  // ----------------------------
  InternetStackHelper stack;
  stack.Install(nodes);

  Ipv4AddressHelper address;
  address.SetBase("10.1.1.0", "255.255.255.0");
  Ipv4InterfaceContainer interfaces = address.Assign(devices);
  // ----------------------------
  // UDP Echo Server (Node 0)
  // ----------------------------
  UdpEchoServerHelper echoServer(9);
  ApplicationContainer serverApp = echoServer.Install(nodes.Get(0));
```

```
 serverApp.Start(Seconds(1.0));
  serverApp.Stop(Seconds(10.0));
  // ----------------------------
  // UDP Echo Clients (Nodes 1–3)
  // ----------------------------
  for (uint32_t i = 1; i < nodes.GetN(); i++)
  {
    UdpEchoClientHelper echoClient(interfaces.GetAddress(0), 9);
    echoClient.SetAttribute("MaxPackets", UintegerValue(1));
    echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
    echoClient.SetAttribute("PacketSize", UintegerValue(1024));

    ApplicationContainer clientApp = echoClient.Install(nodes.Get(i));

    clientApp.Start(Seconds(2.0 + i));
    clientApp.Stop(Seconds(10.0));
  }
  // ----------------------------
  // NetAnim Visualization
  // ----------------------------
  AnimationInterface anim("bus-topology.xml");
  // Position nodes in straight line (Bus Layout)
  int x_start = 10;
  int y_pos = 30;

  for (uint32_t i = 0; i < nodes.GetN(); i++)
  {
    anim.SetConstantPosition(nodes.Get(i), x_start + i * 20, y_pos);
    std::string desc = "Node " + std::to_string(i);
    anim.UpdateNodeDescription(nodes.Get(i), desc);

    // Server = Red, Clients = Blue
    if (i == 0)
      anim.UpdateNodeColor(nodes.Get(i), 255, 0, 0);
    else
      anim.UpdateNodeColor(nodes.Get(i), 0, 0, 255);
  }
  Simulator::Run();
  Simulator::Destroy();

  return 0;
}
```

Mesh Topology - Output



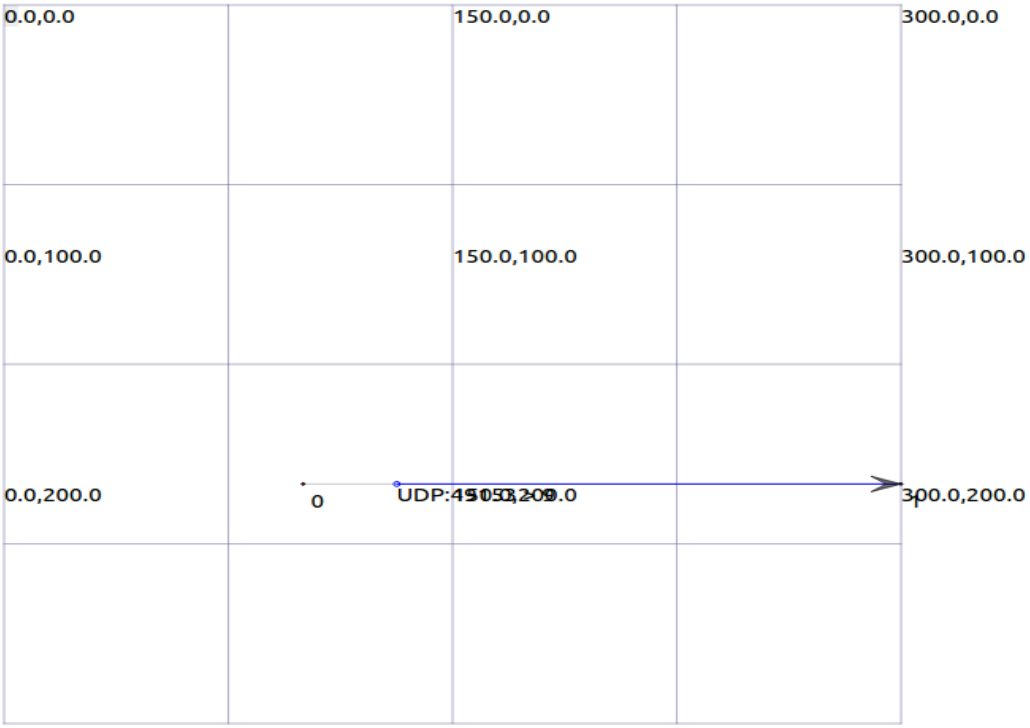

```
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$ ./ns3 run scratch/mesh-topology
[0/2] Re-checking globbed directories...
[2/2] Linking CXX executable /home/itnc/ns-allinone-3.36/ns-3.36/build/scratch/ns3.36-mesh-topology-default
At time +2s client sent 512 bytes to 10.1.1.2 port 9
At time +2s client sent 512 bytes to 10.1.2.2 port 9
At time +2s client sent 512 bytes to 10.1.3.2 port 9
At time +2s client sent 512 bytes to 10.1.4.2 port 9
At time +2.00243s server received 512 bytes from 10.1.1.1 port 49153
At time +2.00243s server sent 512 bytes to 10.1.1.1 port 49153
At time +2.00243s server received 512 bytes from 10.1.2.1 port 49154
At time +2.00243s server sent 512 bytes to 10.1.2.1 port 49154
At time +2.00243s server received 512 bytes from 10.1.3.1 port 49155
At time +2.00243s server sent 512 bytes to 10.1.3.1 port 49155
At time +2.00243s server received 512 bytes from 10.1.4.1 port 49156
At time +2.00243s server sent 512 bytes to 10.1.4.1 port 49156
```

Mesh Topology - Code

```
#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("MeshTopologyWithNetAnim");
int main(int argc, char *argv[])
{
  CommandLine cmd(__FILE__);
  cmd.Parse(argc, argv);
  Time::SetResolution(Time::NS);
  LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
  LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
  // ----------------------------
  // Create Nodes
  // ----------------------------
  NodeContainer nodes;
  nodes.Create(4);
  // ----------------------------
  // Point-to-Point Links (Full Mesh)
  // ----------------------------
  PointToPointHelper p2p;
  p2p.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
  p2p.SetChannelAttribute("Delay", StringValue("2ms"));
  NetDeviceContainer d01 = p2p.Install(nodes.Get(0), nodes.Get(1));
  NetDeviceContainer d02 = p2p.Install(nodes.Get(0), nodes.Get(2));
  NetDeviceContainer d03 = p2p.Install(nodes.Get(0), nodes.Get(3));
  NetDeviceContainer d12 = p2p.Install(nodes.Get(1), nodes.Get(2));
  NetDeviceContainer d13 = p2p.Install(nodes.Get(1), nodes.Get(3));
  NetDeviceContainer d23 = p2p.Install(nodes.Get(2), nodes.Get(3));
  // ----------------------------
  // Internet Stack
  // ----------------------------
  InternetStackHelper stack;
  stack.Install(nodes);
  Ipv4AddressHelper address;
  Ipv4InterfaceContainer i01, i02, i03, i12, i13, i23;
  address.SetBase("10.1.1.0", "255.255.255.0");
  i01 = address.Assign(d01);
  address.SetBase("10.1.2.0", "255.255.255.0");
  i02 = address.Assign(d02);
  address.SetBase("10.1.3.0", "255.255.255.0");
  i03 = address.Assign(d03);
```

```
 address.SetBase("10.1.4.0", "255.255.255.0");
  i12 = address.Assign(d12);
  address.SetBase("10.1.5.0", "255.255.255.0");
  i13 = address.Assign(d13);
  address.SetBase("10.1.6.0", "255.255.255.0");
  i23 = address.Assign(d23);
  Ipv4GlobalRoutingHelper::PopulateRoutingTables();
  // ----------------------------
  // UDP Echo Server (Node 0)
  // ----------------------------
  UdpEchoServerHelper echoServer(9);
  ApplicationContainer serverApp = echoServer.Install(nodes.Get(0));
  serverApp.Start(Seconds(1.0));
  serverApp.Stop(Seconds(10.0));
  // ----------------------------
  // UDP Echo Client (Node 3 → Node 0)
  // ----------------------------
  UdpEchoClientHelper echoClient(i03.GetAddress(0), 9);
  echoClient.SetAttribute("MaxPackets", UintegerValue(1));
  echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
  echoClient.SetAttribute("PacketSize", UintegerValue(1024));
  ApplicationContainer clientApp = echoClient.Install(nodes.Get(3));
  clientApp.Start(Seconds(2.0));
  clientApp.Stop(Seconds(10.0));
  // ----------------------------
  // NetAnim Visualization
  // ----------------------------
  AnimationInterface anim("mesh-topology.xml");
  // Position nodes in mesh layout (square)
  anim.SetConstantPosition(nodes.Get(0), 20, 40);
  anim.SetConstantPosition(nodes.Get(1), 60, 40);
  anim.SetConstantPosition(nodes.Get(2), 20, 10);
  anim.SetConstantPosition(nodes.Get(3), 60, 10);

  for (uint32_t i = 0; i < nodes.GetN(); i++)
  {
     std::string desc = "Node " + std::to_string(i);
     anim.UpdateNodeDescription(nodes.Get(i), desc);
     if (i == 0)
        anim.UpdateNodeColor(nodes.Get(i), 255, 0, 0);  // Server = Red
     else
        anim.UpdateNodeColor(nodes.Get(i), 0, 0, 255);  // Others = Blue
  }
  Simulator::Run();
  Simulator::Destroy();

  return 0;
}
```

Stop-and-Wait Protocol - Output



gg

```
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$ ./ns3 run scratch/stop-wait
[0/2] Re-checking globbed directories...
ninja: no work to do.
At time +2s client sent 1024 bytes to 10.1.1.2 port 9
At time +2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 10.1.1.2 port 9
At time +4s client sent 1024 bytes to 10.1.1.2 port 9
At time +4.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +4.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +4.00737s client received 1024 bytes from 10.1.1.2 port 9
At time +6s client sent 1024 bytes to 10.1.1.2 port 9
At time +6.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +6.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +6.00737s client received 1024 bytes from 10.1.1.2 port 9
At time +8s client sent 1024 bytes to 10.1.1.2 port 9
At time +8.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time +8.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time +8.00737s client received 1024 bytes from 10.1.1.2 port 9
At time +10s client sent 1024 bytes to 10.1.1.2 port 9
At time +10.0037s server received 1024 bytes from 10.1.1.1 port 49153
At time +10.0037s server sent 1024 bytes to 10.1.1.1 port 49153
At time +10.0074s client received 1024 bytes from 10.1.1.2 port 9
itnc@itnc-Veriton-M6650G:~/ns-allinone-3.36/ns-3.36$
```

Stop-and-Wait Protocol - Code

```
#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("StopAndWaitWithNetAnim");
int main(int argc, char *argv[])
{
   CommandLine cmd(__FILE__);
   cmd.Parse(argc, argv);
   Time::SetResolution(Time::NS);
  LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
   LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
   // ----------------------------
   // Create Nodes
   // ----------------------------
   NodeContainer nodes;
   nodes.Create(2);
    // ----------------------------
   // Point-to-Point Channel
   // ----------------------------
   PointToPointHelper p2p;
   p2p.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
   p2p.SetChannelAttribute("Delay", StringValue("2ms"));
   NetDeviceContainer devices = p2p.Install(nodes);
    // ----------------------------
   // Internet Stack
   // ----------------------------
   InternetStackHelper stack;
   stack.Install(nodes);
   Ipv4AddressHelper address;
   address.SetBase("10.1.1.0", "255.255.255.0");
   Ipv4InterfaceContainer interfaces = address.Assign(devices);
   // ----------------------------
   // UDP Echo Server (Receiver)
   // ----------------------------
   UdpEchoServerHelper echoServer(9);
   ApplicationContainer serverApp = echoServer.Install(nodes.Get(1));
   serverApp.Start(Seconds(1.0));
   serverApp.Stop(Seconds(10.0));
   // ----------------------------
   // UDP Echo Client (Sender)
   // Stop-and-Wait behavior:
```

```
    // 1 packet → wait → next packet
    // ----------------------------
    UdpEchoClientHelper echoClient(interfaces.GetAddress(1), 9);

    echoClient.SetAttribute("MaxPackets", UintegerValue(5));
    echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0))); // wait time
    echoClient.SetAttribute("PacketSize", UintegerValue(1024));

    ApplicationContainer clientApp = echoClient.Install(nodes.Get(0));

    clientApp.Start(Seconds(2.0));
    clientApp.Stop(Seconds(10.0));

    // ----------------------------
    // NetAnim Visualization
    // ----------------------------
    AnimationInterface anim("stop-and-wait.xml");

    // Node Positions
    anim.SetConstantPosition(nodes.Get(0), 20, 30); // Sender
    anim.SetConstantPosition(nodes.Get(1), 60, 30); // Receiver

    // Node Descriptions
    anim.UpdateNodeDescription(nodes.Get(0), "Sender");
    anim.UpdateNodeDescription(nodes.Get(1), "Receiver");

    // Node Colors
    anim.UpdateNodeColor(nodes.Get(0), 0, 0, 255);   // Blue = Sender
    anim.UpdateNodeColor(nodes.Get(1), 255, 0, 0);   // Red = Receiver

    Simulator::Run();
    Simulator::Destroy();

    return 0;
}
```