Lab-6 Date:02/09/2024

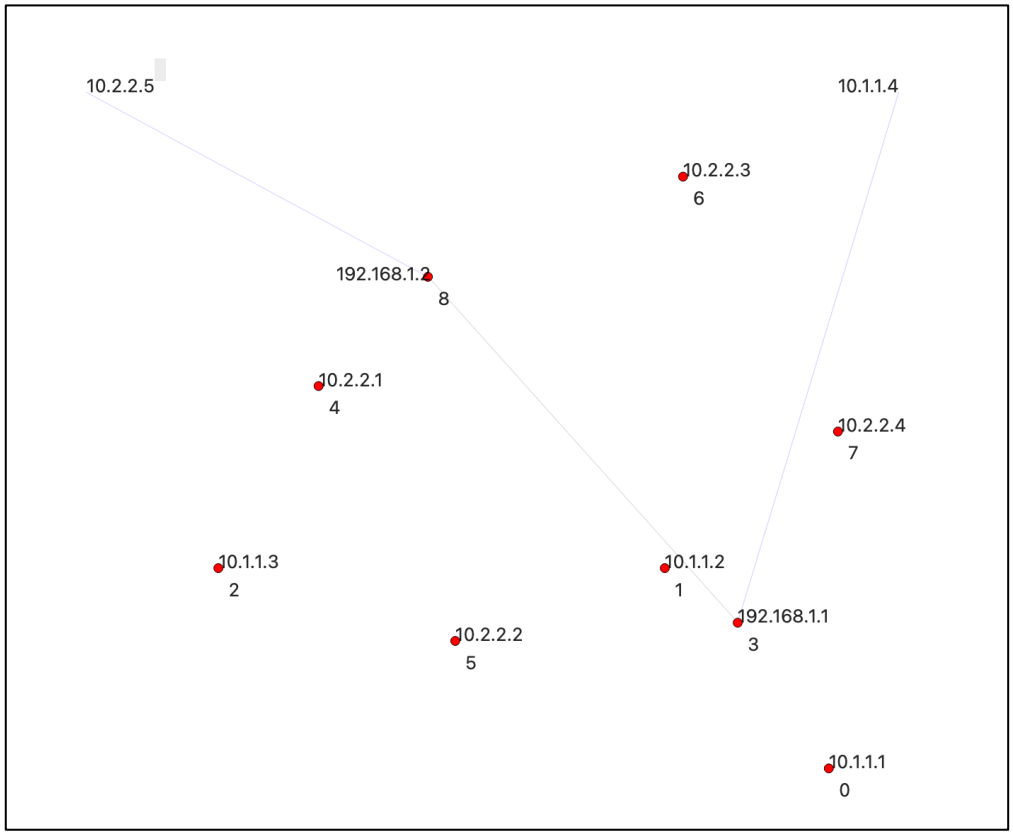
Q1. Create a network with 2 LANs connected and 2 routers (R1, R2), as specified in the below diagram.

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/ipv4-address-helper.h"

using namespace ns3;

int main (int argc, char *argv[])
{
  // Set up LAN 1
  NodeContainer lan1Nodes;
  lan1Nodes.Create (3);

  NodeContainer router1;
  router1.Create (1);

  NodeContainer lan1;
  lan1.Add (lan1Nodes);
  lan1.Add (router1);

  CsmaHelper csma1;
  csma1.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
  csma1.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

  NetDeviceContainer lan1Devices;
  lan1Devices = csma1.Install (lan1);

  // Set up LAN 2
  NodeContainer lan2Nodes;
  lan2Nodes.Create (4);

  NodeContainer router2;
  router2.Create (1);

  NodeContainer lan2;
  lan2.Add (lan2Nodes);
  lan2.Add (router2);

  CsmaHelper csma2;
```

```
csma2.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma2.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

NetDeviceContainer lan2Devices;
lan2Devices = csma2.Install (lan2);

// Set up the point-to-point link between the routers
NodeContainer routers;
routers.Add (router1);
routers.Add (router2);

PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", StringValue ("1Gbps"));
p2p.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));

NetDeviceContainer routerDevices;
routerDevices = p2p.Install (routers);

// Install the Internet stack
InternetStackHelper stack;
stack.Install (lan1);
stack.Install (lan2);

// Assign IP addresses
Ipv4AddressHelper address1;
address1.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer lan1Interfaces;
lan1Interfaces = address1.Assign (lan1Devices);

Ipv4AddressHelper address2;
address2.SetBase ("10.2.2.0", "255.255.255.0");
Ipv4InterfaceContainer lan2Interfaces;
lan2Interfaces = address2.Assign (lan2Devices);

Ipv4AddressHelper addressP2P;
addressP2P.SetBase ("192.168.1.0", "255.255.255.252");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = addressP2P.Assign (routerDevices);

// Enable routing
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// Create some applications (optional)
UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (lan2Nodes.Get (0));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (lan2Interfaces.GetAddress (0), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
```

```
  echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
  echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

  ApplicationContainer clientApps = echoClient.Install (lan1Nodes.Get (0));
  clientApps.Start (Seconds (2.0));
  clientApps.Stop (Seconds (10.0));

  // Enable pcap tracing (optional)
  csma1.EnablePcap ("lan1", lan1Devices.Get (0), true);
  csma2.EnablePcap ("lan2", lan2Devices.Get (0), true);
  p2p.EnablePcapAll ("p2p");

  // Run the simulation
  Simulator::Run ();
  Simulator::Destroy ();

  return 0;
}
```

Simulation:



Q2. Create an ASCII trace file for the above network.

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/csma-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
#include "ns3/ipv4-address-helper.h"

using namespace ns3;

int main (int argc, char *argv[])
{
  // Set up LAN 1
  NodeContainer lan1Nodes;
  lan1Nodes.Create (3);

  NodeContainer router1;
  router1.Create (1);

  NodeContainer lan1;
  lan1.Add (lan1Nodes);
  lan1.Add (router1);

  CsmaHelper csma1;
  csma1.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
  csma1.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

  NetDeviceContainer lan1Devices;
  lan1Devices = csma1.Install (lan1);

  // Set up LAN 2
  NodeContainer lan2Nodes;
  lan2Nodes.Create (4);

  NodeContainer router2;
  router2.Create (1);

  NodeContainer lan2;
  lan2.Add (lan2Nodes);
  lan2.Add (router2);

  CsmaHelper csma2;
  csma2.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
  csma2.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));
```

```
NetDeviceContainer lan2Devices;
lan2Devices = csma2.Install (lan2);

// Set up the point-to-point link between the routers
NodeContainer routers;
routers.Add (router1);
routers.Add (router2);

PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", StringValue ("1Gbps"));
p2p.SetChannelAttribute ("Delay", TimeValue (MilliSeconds (2)));

NetDeviceContainer routerDevices;
routerDevices = p2p.Install (routers);

// Install the Internet stack
InternetStackHelper stack;
stack.Install (lan1);
stack.Install (lan2);

// Assign IP addresses
Ipv4AddressHelper address1;
address1.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer lan1Interfaces;
lan1Interfaces = address1.Assign (lan1Devices);

Ipv4AddressHelper address2;
address2.SetBase ("10.2.2.0", "255.255.255.0");
Ipv4InterfaceContainer lan2Interfaces;
lan2Interfaces = address2.Assign (lan2Devices);

Ipv4AddressHelper addressP2P;
addressP2P.SetBase ("192.168.1.0", "255.255.255.252");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = addressP2P.Assign (routerDevices);

// Enable routing
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// Create some applications (optional)
UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (lan2Nodes.Get (0));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (lan2Interfaces.GetAddress (0), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
```

```
  ApplicationContainer clientApps = echoClient.Install (lan1Nodes.Get (0));
  clientApps.Start (Seconds (2.0));
  clientApps.Stop (Seconds (10.0));

  // Enable ASCII tracing
  AsciiTraceHelper ascii;
  csma1.EnableAsciiAll (ascii.CreateFileStream ("lan1.tr"));
  csma2.EnableAsciiAll (ascii.CreateFileStream ("lan2.tr"));
  p2p.EnableAsciiAll (ascii.CreateFileStream ("p2p.tr"));

  // Enable pcap tracing (optional)
  csma1.EnablePcap ("lan1", lan1Devices.Get (0), true);
  csma2.EnablePcap ("lan2", lan2Devices.Get (0), true);
  p2p.EnablePcapAll ("p2p");

  // Run the simulation
  Simulator::Run ();
  Simulator::Destroy ();

  return 0;
}
```

Output:-

```
+ 2.00512 /NodeList/3/DeviceList/1/$ns3::PointToPointNetDevice/TxQueue/Enqueue
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 63 id 0 protocol 17 offset (bytes) 0 flags [none]
length: 1052 10.1.1.1 > 10.2.2.1) ns3::UdpHeader (length: 1032 49153 > 9)
Payload (size=1024)
- 2.00512 /NodeList/3/DeviceList/1/$ns3::PointToPointNetDevice/TxQueue/Dequeue
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 63 id 0 protocol 17 offset (bytes) 0 flags [none]
length: 1052 10.1.1.1 > 10.2.2.1) ns3::UdpHeader (length: 1032 49153 > 9)
Payload (size=1024)
r 2.00713 /NodeList/8/DeviceList/1/$ns3::PointToPointNetDevice/MacRx
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 63 id 0 protocol 17 offset (bytes) 0 flags [none]
length: 1052 10.1.1.1 > 10.2.2.1) ns3::UdpHeader (length: 1032 49153 > 9)
Payload (size=1024)
+ 2.02336 /NodeList/8/DeviceList/1/$ns3::PointToPointNetDevice/TxQueue/Enqueue
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 63 id 0 protocol 17 offset (bytes) 0 flags [none]
length: 1052 10.2.2.1 > 10.1.1.1) ns3::UdpHeader (length: 1032 9 > 49153)
Payload (size=1024)
- 2.02336 /NodeList/8/DeviceList/1/$ns3::PointToPointNetDevice/TxQueue/Dequeue
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 63 id 0 protocol 17 offset (bytes) 0 flags [none]
length: 1052 10.2.2.1 > 10.1.1.1) ns3::UdpHeader (length: 1032 9 > 49153)
Payload (size=1024)
r 2.02537 /NodeList/3/DeviceList/1/$ns3::PointToPointNetDevice/MacRx
ns3::PppHeader (Point-to-Point Protocol: IP (0x0021)) ns3::Ipv4Header (tos 0x0
DSCP Default ECN Not-ECT ttl 63 id 0 protocol 17 offset (bytes) 0 flags [none]
length: 1052 10.2.2.1 > 10.1.1.1) ns3::UdpHeader (length: 1032 9 > 49153)
Payload (size=1024)
```