

## EXPERIMENT 4

**AIM:** Create a topology of P2P (two nodes) and CSMA channel (4 nodes) specifying bandwidth 5 Mbps and 100 Mbps respectively.

### **NS-3 CODE:**

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/ipv4-global-routing-helper.h"
// Default Network Topology
//
//      10.1.1.0
// n0 ----- n1   n2   n3   n4
//     point-to-point |   |   |
//                      =====
//                           LAN 10.1.2.0
using namespace ns3;
NS_LOG_COMPONENT_DEFINE("SecondScriptExample");

int main(int argc, char* argv[])
{
    bool verbose = true;
    uint32_t nCsma = 3;
    CommandLine cmd;
    cmd.AddValue("nCsma", "Number of \\\"extra\\\" CSMA nodes/devices", nCsma);
    cmd.AddValue("verbose", "Tell echo applications to log if true", verbose);
    cmd.Parse(argc, argv);

    if (verbose) {
        LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
        LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);
    }
    nCsma = nCsma == 0 ? 1 : nCsma;

    NodeContainer p2pNodes; // keep track of a set of node pointers.
    p2pNodes.Create(2); // create two nodes, append pointers into NodeContainer

    NodeContainer csmaNodes;
    csmaNodes.Add(p2pNodes.Get(1)); // Append contents of other NodeContainer to this one.
    csmaNodes.Create(nCsma);

    PointToPointHelper pointToPoint;
    pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
    pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));

    NetDeviceContainer p2pDevices;
    p2pDevices = pointToPoint.Install(p2pNodes);
    CsmaHelper csma;
    csma.SetChannelAttribute("DataRate", StringValue("100Mbps"));
    csma.SetChannelAttribute("Delay", TimeValue(NanoSeconds(6560))));
```

```

NetDeviceContainer csmaDevices;
csmaDevices = csma.Install(csmaNodes);

// aggregate IP/TCP/UDP functionality to existing Nodes.
InternetStackHelper stack;
stack.Install(p2pNodes.Get(0));      stack.Install(csmaNodes);

Ipv4AddressHelper address;
address.SetBase("10.1.1.0", "255.255.255.0"); // Set the base network number, network mask
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign(p2pDevices); // Assign IP addresses to the net devices

address.SetBase("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign(csmaDevices);

// Create a server application which waits for input UDP packets and sends them back
UdpEchoServerHelper echoServer(9);

// Create a UdpEchoServerApplication on the specified Node.
ApplicationContainer serverApps = echoServer.Install(csmaNodes.Get(nCsma));
// Arrange for all applications in this container to Start() and Stop() at the Time given
serverApps.Start(Seconds(1.0));      serverApps.Stop(Seconds(10.0));

// Create an application which sends a UDP packet and waits for an echo of this packet.
UdpEchoClientHelper echoClient(csmaInterfaces.GetAddress(nCsma), 9);
echoClient.SetAttribute("MaxPackets", UintegerValue(1));
echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
echoClient.SetAttribute("PacketSize", UintegerValue(1024));

ApplicationContainer clientApps = echoClient.Install(p2pNodes.Get(0));
clientApps.Start(Seconds(2.0));      clientApps.Stop(Seconds(10.0));

// Build a routing database and initialize the routing tables of the nodes
Ipv4GlobalRoutingHelper::PopulateRoutingTables();
// Enable pcap output on each device
pointToPoint.EnablePcapAll("second");
csma.EnablePcap("second", csmaDevices.Get(1), true);

Simulator::Run();      Simulator::Destroy();
return 0;
}

```

## OUTPUT

```

>> ./waf --run "scratch/mysecond --nCsma=4"

Waf: Entering directory `/home/craigdo/repos/ns-3-allinone/ns-3-dev/build'
Waf: Leaving directory `/home/craigdo/repos/ns-3-allinone/ns-3-dev/build'
'build' finished successfully (0.405s)
At time 2s client sent 1024 bytes to 10.1.2.5 port 9
At time 2.0118s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.0118s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.02461s client received 1024 bytes from 10.1.2.5 port 9

```