Lab Assignment 1

Germain Mucyo & Brian Kubinec

mucyo.g@northeastern.edu

kubinec.b@northeastern.edu

NU ID (Germain Mucyo): 002301781

NU ID (Brian Kubinec): 001005207

Instructor: Professor Bonati

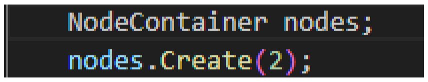
Submit Date: 1/27/2025

Due Date: 1/28/2025

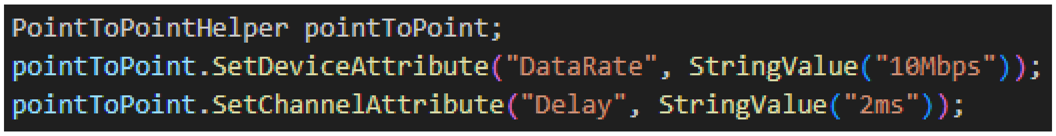
Task 1

Experimental Setup

The objective of this first assignment is to become familiar with the ns-3 working environment and the simulation workflow. We first created a working environment with ns-3 and Wireshark. The first task was to simulate a network of 2 nodes with 1 interface at each node. There is a point-to-point link of 10 Mbps data rate with a delay of 2 ms. IP address assignment 192.168.2.0/24. The application is a UDP Echo Server on port 63 with a packet size of 256 bytes. The Network Simulator, ns-3.42 was used. The first.cc program was to be changed for this task. The specific lines were as follows:

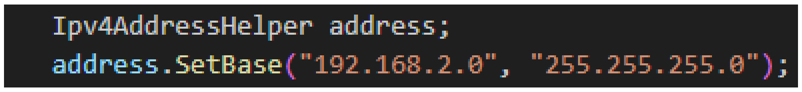


```
    NodeContainer nodes;
    nodes.Create(2);
```



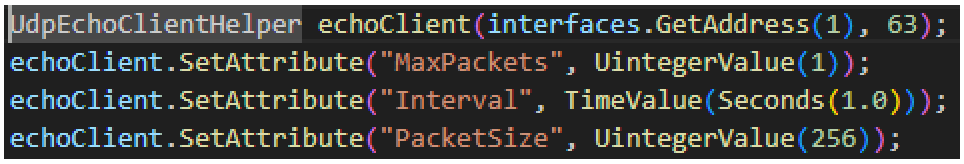
```
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute("DataRate", StringValue("10Mbps"));
pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
```

The nodes are two nodes. StringValue were changed for both SetDeviceAttribute and SetChannelAttribute corresponding for the Data Rate and Delay from 5 Mbps to 10 Mbps and the delay was to 2 ms.



```
   Ipv4AddressHelper address;
   address.SetBase("192.168.2.0", "255.255.255.0");
```

The address.SetBase was changed from 10.1.1.0 top 192.168.2.0

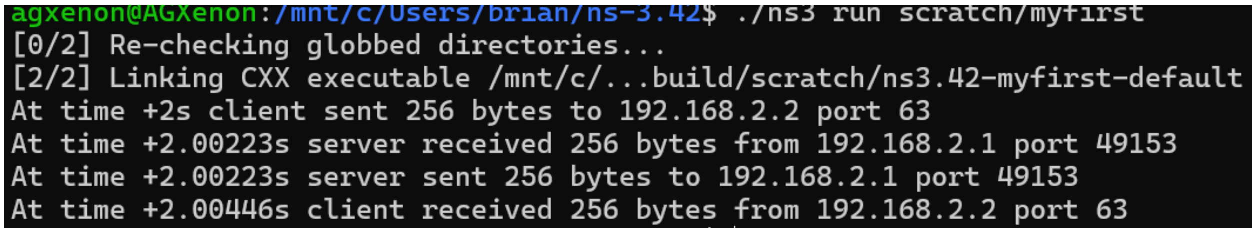


```
UdpEchoClientHelper echoClient(interfaces.GetAddress(1), 63);
echoClient.SetAttribute("MaxPackets", UintegerValue(1));
echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
echoClient.SetAttribute("PacketSize", UintegerValue(256));
```

echoClient was changed from port 9 to port 63 along with the packet size in echClient.SetAttribute(“PacketSize”, UintegerValue(1024 -> 256)).

Results

The program was then run with the output as follows:

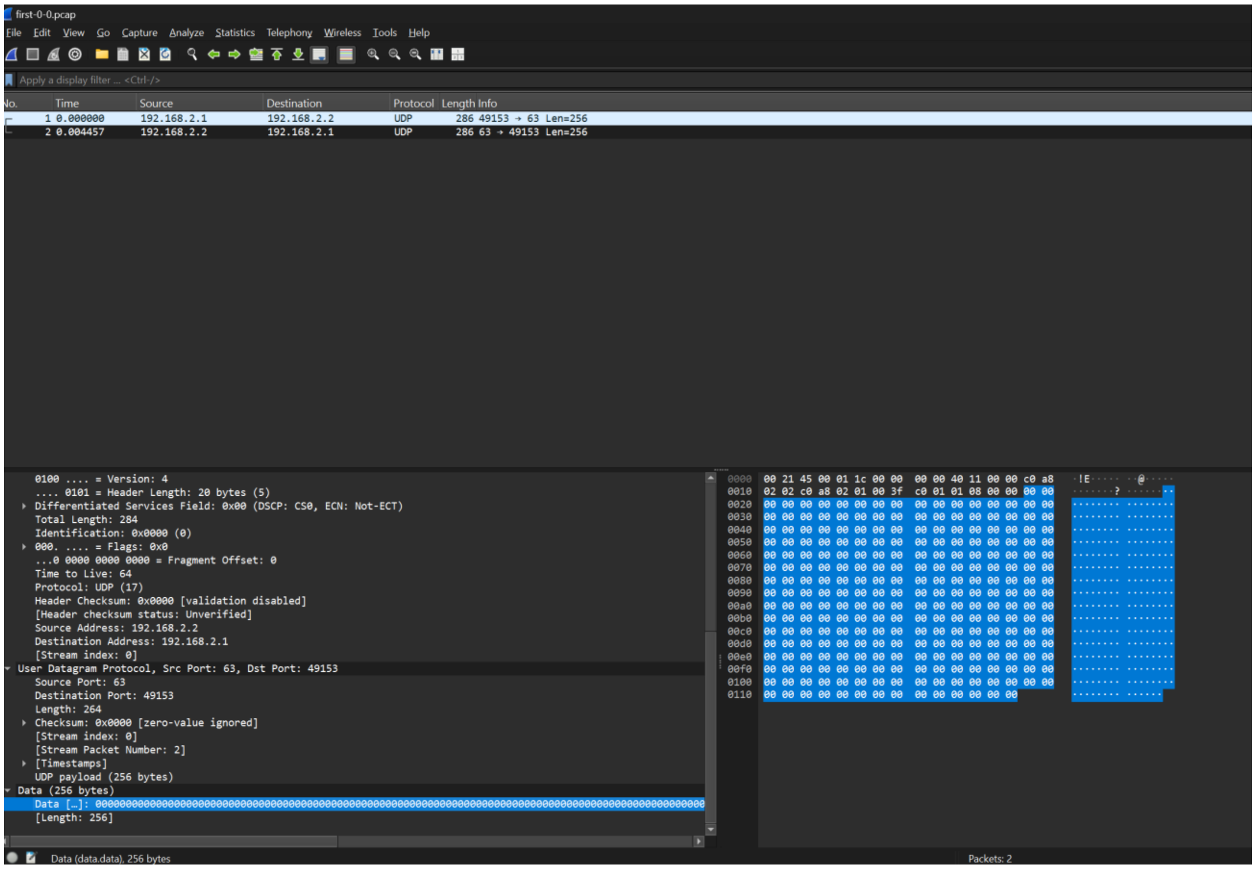


```
agxenon@AGXenon:/mnt/c/Users/brian/ns-3.42$ ./ns3 run scratch/myfirst
[0/2] Re-checking globbed directories...
[2/2] Linking CXX executable /mnt/c/...build/scratch/ns3.42-myfirst-default
At time +2s client sent 256 bytes to 192.168.2.2 port 63
At time +2.00223s server received 256 bytes from 192.168.2.1 port 49153
At time +2.00223s server sent 256 bytes to 192.168.2.1 port 49153
At time +2.00446s client received 256 bytes from 192.168.2.2 port 63
```

Figure 2. Output of UDP Echo Server-Client interaction of 2 nodes.

Explanation: The output demonstrates a simple point-to-point network in ns-3 with two nodes connected at 10 Mbps and 2 ms delay in the 192.168.2.0/24 subnet. A UDP Echo Server on Node 1 (port 63) and a

UDP Echo Client on Node 0 (Port 49153 randomly generated by our local system) successfully exchanged 256-byte packets, achieving a round-trip delay of approximately 4.46 ms. The results confirm the accurate configuration and functionality of the network and the UDP Echo application. Thereafter, we added pointToPoint.EnablePcapAll () to enable .pcpa tracing files (first 0-0 and first 1-0), which will be used to visualize our results in detail.


first-0-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

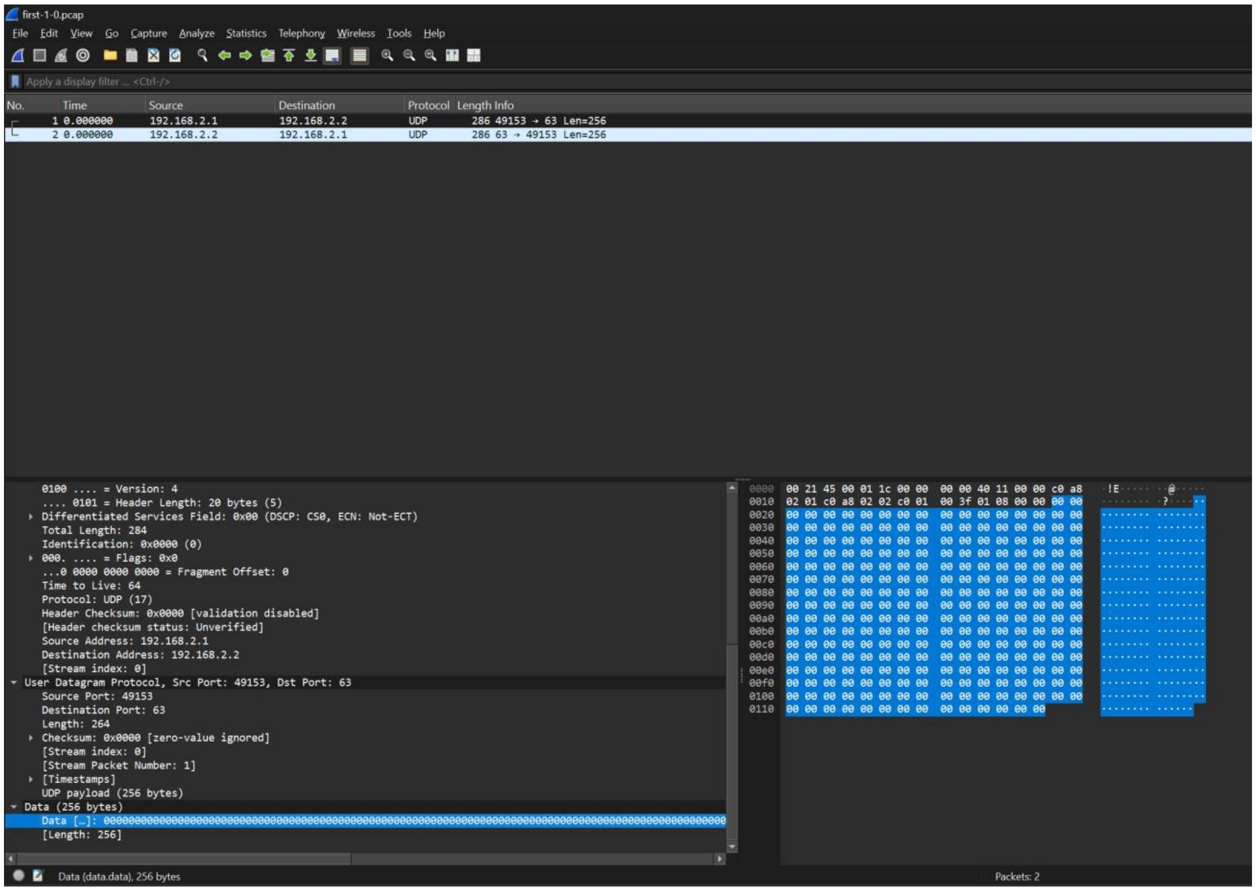
```
No.  Time      Source        Destination   Protocol Length Info
   1 0.000000  192.168.2.1   192.168.2.2   UDP      286 49153 → 63 Len=256
   2 0.004457  192.168.2.2   192.168.2.1   UDP      286 63 → 49153 Len=256
```

```
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 284
    Identification: 0x0000 (0)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.2.2
    Destination Address: 192.168.2.1
    [Stream index: 0]
▾ User Datagram Protocol, Src Port: 63, Dst Port: 49153
    Source Port: 63
    Destination Port: 49153
    Length: 264
  ▸ Checksum: 0x0000 [zero-value ignored]
    [Stream index: 0]
    [Stream Packet Number: 2]
  ▸ [Timestamps]
    UDP payload (256 bytes)
▾ Data (256 bytes)
    Data […]: 00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
    [Length: 256]
```

```
0000  00 21 45 00 01 1c 00 00  00 00 40 11 00 00 c0 a8   ·!E·····  ··@·····
0010  02 02 c0 a8 02 01 00 3f  c0 01 01 08 00 00 00 00   ·······?  ········
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00         ········  ······
```

```
Data (data.data), 256 bytes                                   Packets: 2
```

Figure 3. Wireshark simulation of client-server interaction of node 0.



```
first-1-0.pcap
File  Edit  View  Go  Capture  Analyze  Statistics  Telephony  Wireless  Tools  Help
Apply a display filter ... <Ctrl-/>

No.     Time        Source          Destination     Protocol  Length Info
      1 0.000000    192.168.2.1     192.168.2.2     UDP         286 49153 → 63 Len=256
      2 0.000000    192.168.2.2     192.168.2.1     UDP         286 63 → 49153 Len=256
```

```
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 284
    Identification: 0x0000 (0)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.2.1
    Destination Address: 192.168.2.2
    [Stream index: 0]
▾ User Datagram Protocol, Src Port: 49153, Dst Port: 63
    Source Port: 49153
    Destination Port: 63
    Length: 264
  ▸ Checksum: 0x0000 [zero-value ignored]
    [Stream index: 0]
    [Stream Packet Number: 1]
  ▸ [Timestamps]
    UDP payload (256 bytes)
▾ Data (256 bytes)
    Data […]: 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
    [Length: 256]
```

```
0000  00 21 45 00 01 1c 00 00  00 00 40 11 00 00 c0 a8   ·!E·····  ··@·····
0010  02 01 c0 a8 02 02 c0 01  00 3f 01 08 00 00 00 00   ········  ·?······
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········  ········
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00         ········  ······
```

```
Data (data.data), 256 bytes                                   Packets: 2
```

Figure 4. Wireshark simulation of client-server interaction of node 1.

Lessons Learnt

The objective of this first assignment is to become familiar with the ns-3 working environment and the simulation workflow. The first part of this assignment was to learn how to install and build ns-3 simulator then learn computer networking fundamental concepts, like the difference of UDP and TCP, in this case was Point to Point network (UDP), then were able to learn ns3 simulator workflow and command lines as explained in the experiment setup part. Using pointToPoint.EnablePcapAll () we got .pcpa files that we used to visualize our results in Wireshark.

Task 2

Experimental Setup

Now that we have become familiar with the ns-3 working environment and the simulation workflow, we needed to change the second.cc program in the ns-3.42/examples/tutorial path. Continuing our working environment with ns-3 and Wireshark, we needed to simulate a network of 3 nodes in the first shared bus under CSMA along with this same configuration for the second shared bus. Between the point-to-point link, there are 2 nodes (nodes 2 and 3) each with 1 network interface. The applications running the network are: UDP Echo Server on Node 1 (listening on port 21) and a UDP Echo Client at Node 5 (sending 2 UDP Echo packets to the server at times 4s and 7s. Packet tracing is enabled only in Node 2 in the point-to-point interface and at Node 4. Below represents the network infrastructure:

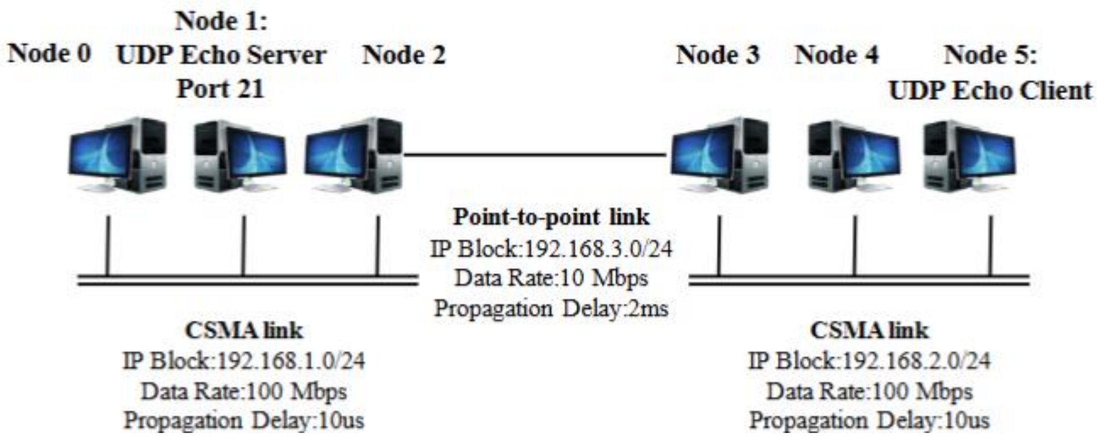
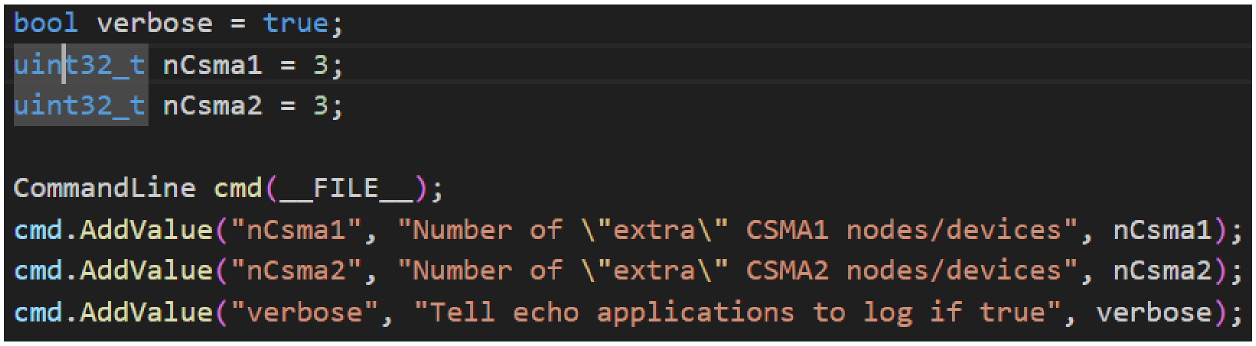


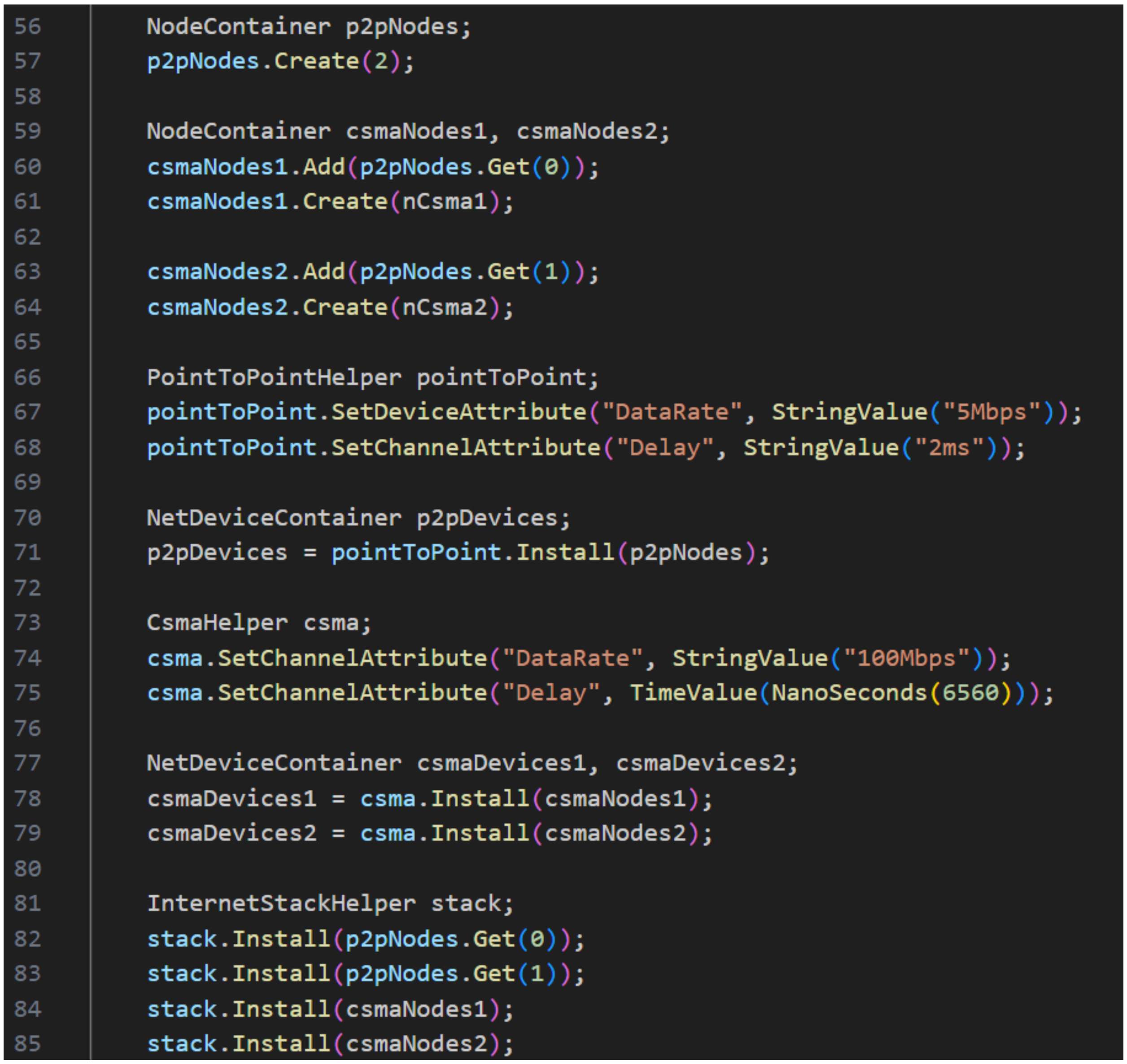
Figure 1. CSMA network Infrastructure.

The specific lines changed in second.cc were as follows:



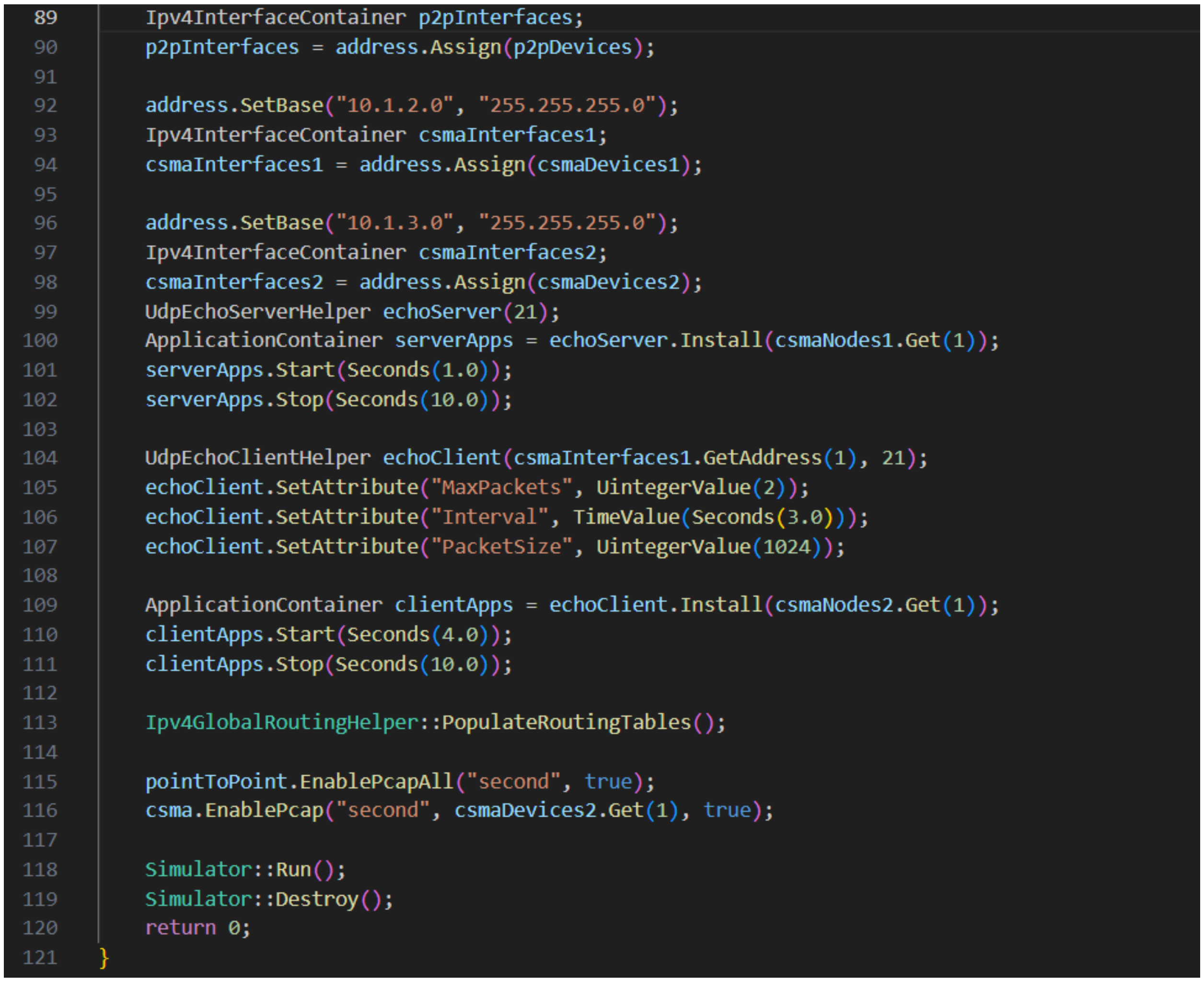
```
bool verbose = true;
uint32_t nCsma1 = 3;
uint32_t nCsma2 = 3;

CommandLine cmd(__FILE__);
cmd.AddValue("nCsma1", "Number of \"extra\" CSMA1 nodes/devices", nCsma1);
cmd.AddValue("nCsma2", "Number of \"extra\" CSMA2 nodes/devices", nCsma2);
cmd.AddValue("verbose", "Tell echo applications to log if true", verbose);
```

The two variables created to correspond to the CSMA between the first and second shared buses which needed 3 nodes each. This corresponded to addValue being changed for the cmd.



```
56      NodeContainer p2pNodes;
57      p2pNodes.Create(2);
58
59      NodeContainer csmaNodes1, csmaNodes2;
60      csmaNodes1.Add(p2pNodes.Get(0));
61      csmaNodes1.Create(nCsma1);
62
63      csmaNodes2.Add(p2pNodes.Get(1));
64      csmaNodes2.Create(nCsma2);
65
66      PointToPointHelper pointToPoint;
67      pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
68      pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
69
70      NetDeviceContainer p2pDevices;
71      p2pDevices = pointToPoint.Install(p2pNodes);
72
73      CsmaHelper csma;
74      csma.SetChannelAttribute("DataRate", StringValue("100Mbps"));
75      csma.SetChannelAttribute("Delay", TimeValue(NanoSeconds(6560)));
76
77      NetDeviceContainer csmaDevices1, csmaDevices2;
78      csmaDevices1 = csma.Install(csmaNodes1);
79      csmaDevices2 = csma.Install(csmaNodes2);
80
81      InternetStackHelper stack;
82      stack.Install(p2pNodes.Get(0));
83      stack.Install(p2pNodes.Get(1));
84      stack.Install(csmaNodes1);
85      stack.Install(csmaNodes2);
```

Because of this, variables csmaNodes1 and csmaNodes2 were also created (line 59). Undoubtedly, this cascade changes for the creation of nodes csma1 and csma2 (lines 60 to 64). The installation of point-to-point nodes remained the same but installing csmaNodes1 and csmaNodes2 into csmaDevices1 and csmaDevices2 were necessary. Line 83 was needed in addition to the stack. The installation of csmaNodes1 and csmaNodes2 into the stack is also necessary (lines 84 and 85).

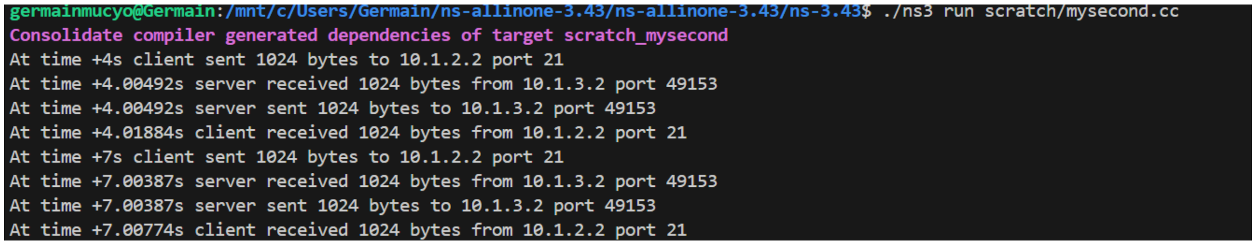


```
 89      Ipv4InterfaceContainer p2pInterfaces;
 90      p2pInterfaces = address.Assign(p2pDevices);
 91
 92      address.SetBase("10.1.2.0", "255.255.255.0");
 93      Ipv4InterfaceContainer csmaInterfaces1;
 94      csmaInterfaces1 = address.Assign(csmaDevices1);
 95
 96      address.SetBase("10.1.3.0", "255.255.255.0");
 97      Ipv4InterfaceContainer csmaInterfaces2;
 98      csmaInterfaces2 = address.Assign(csmaDevices2);
 99      UdpEchoServerHelper echoServer(21);
100      ApplicationContainer serverApps = echoServer.Install(csmaNodes1.Get(1));
101      serverApps.Start(Seconds(1.0));
102      serverApps.Stop(Seconds(10.0));
103
104      UdpEchoClientHelper echoClient(csmaInterfaces1.GetAddress(1), 21);
105      echoClient.SetAttribute("MaxPackets", UintegerValue(2));
106      echoClient.SetAttribute("Interval", TimeValue(Seconds(3.0)));
107      echoClient.SetAttribute("PacketSize", UintegerValue(1024));
108
109      ApplicationContainer clientApps = echoClient.Install(csmaNodes2.Get(1));
110      clientApps.Start(Seconds(4.0));
111      clientApps.Stop(Seconds(10.0));
112
113      Ipv4GlobalRoutingHelper::PopulateRoutingTables();
114
115      pointToPoint.EnablePcapAll("second", true);
116      csma.EnablePcap("second", csmaDevices2.Get(1), true);
117
118      Simulator::Run();
119      Simulator::Destroy();
120      return 0;
121  }
```

The interface container also needed this accommodation. This was necessary for both csmaDevices1 and csmaDevices2, which lead to csmaInterfaces1 and csmaInterfaces2 (lines 92 to 98). Line 100 to 102 shows that running the network, UDP Echo Server at node 1 needed to be listening on port 21. Lines 102 and 103 is an open window between 1 and 10 seconds such that the UDP Echo Client sends 2 packets at times 4 and 10 seconds (shown in lines 111 and 112). Line 117 enables packet capture, tracing on the second shared bus in the CSMA network for the csmaDevices2. The true argument indicates that all packets seen by this network interface will be captured.

Results

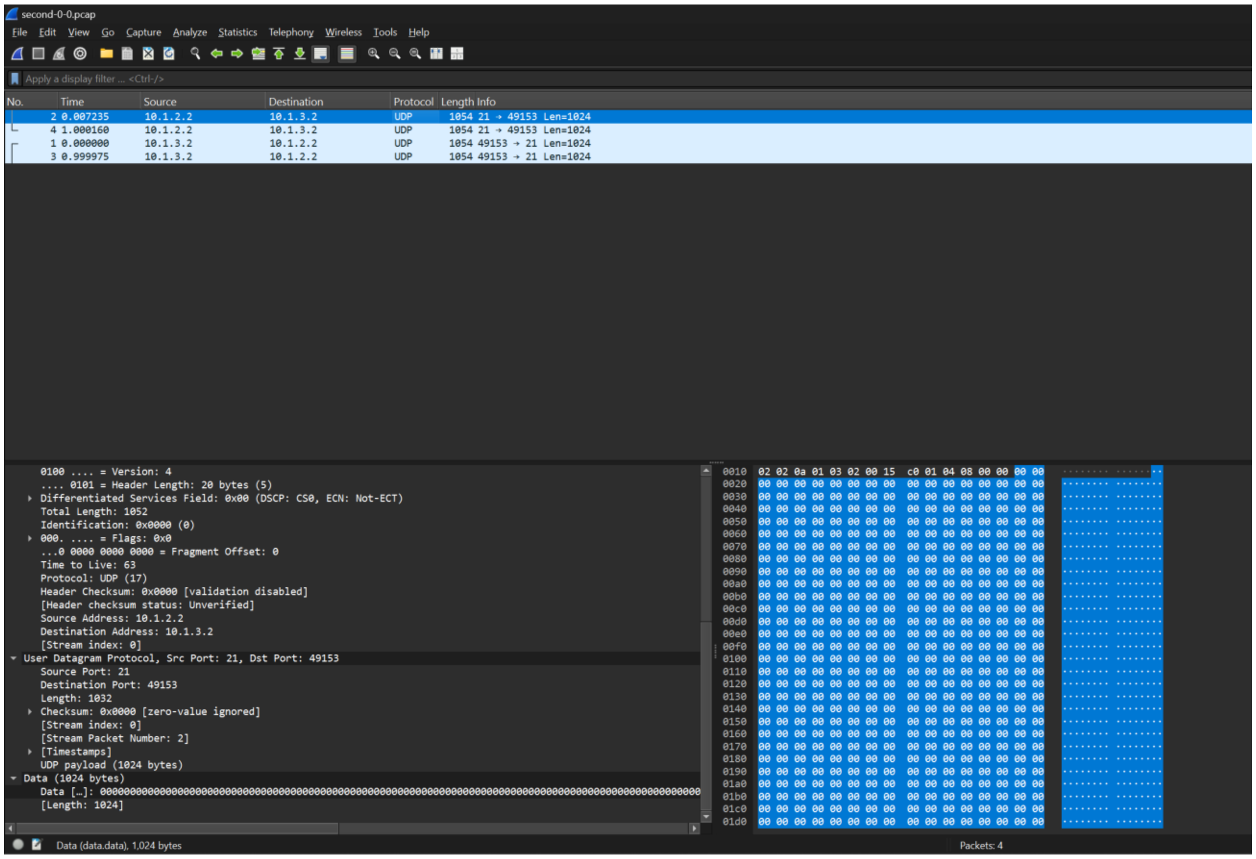
The program was then run with the output as follows:



```
germainmucyo@Germain:/mnt/c/Users/Germain/ns-allinone-3.43/ns-allinone-3.43/ns-3.43$ ./ns3 run scratch/mysecond.cc
Consolidate compiler generated dependencies of target scratch_mysecond
At time +4s client sent 1024 bytes to 10.1.2.2 port 21
At time +4.00492s server received 1024 bytes from 10.1.3.2 port 49153
At time +4.00492s server sent 1024 bytes to 10.1.3.2 port 49153
At time +4.01884s client received 1024 bytes from 10.1.2.2 port 21
At time +7s client sent 1024 bytes to 10.1.2.2 port 21
At time +7.00387s server received 1024 bytes from 10.1.3.2 port 49153
At time +7.00387s server sent 1024 bytes to 10.1.3.2 port 49153
At time +7.00774s client received 1024 bytes from 10.1.2.2 port 21
```

Figure 5. Output of UDP Echo Server-Client interaction on CSMA network interface.

The client sends packet via port 21 while the server receives this on their port (Germain's port). The server then sends a packet to the server via that same port it received, and the client receives this on their port, 21. The client then sends another packet to the server via port 21 and then eventually receives a packet from the server on port 21. Just like in task 1, we used pointToPoint.EnablePcapAll () to enable .pcpa tracing files that we used to visualize our results in Wireshark, see figure 6, 7 and 8.


second-0-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Packets: 4

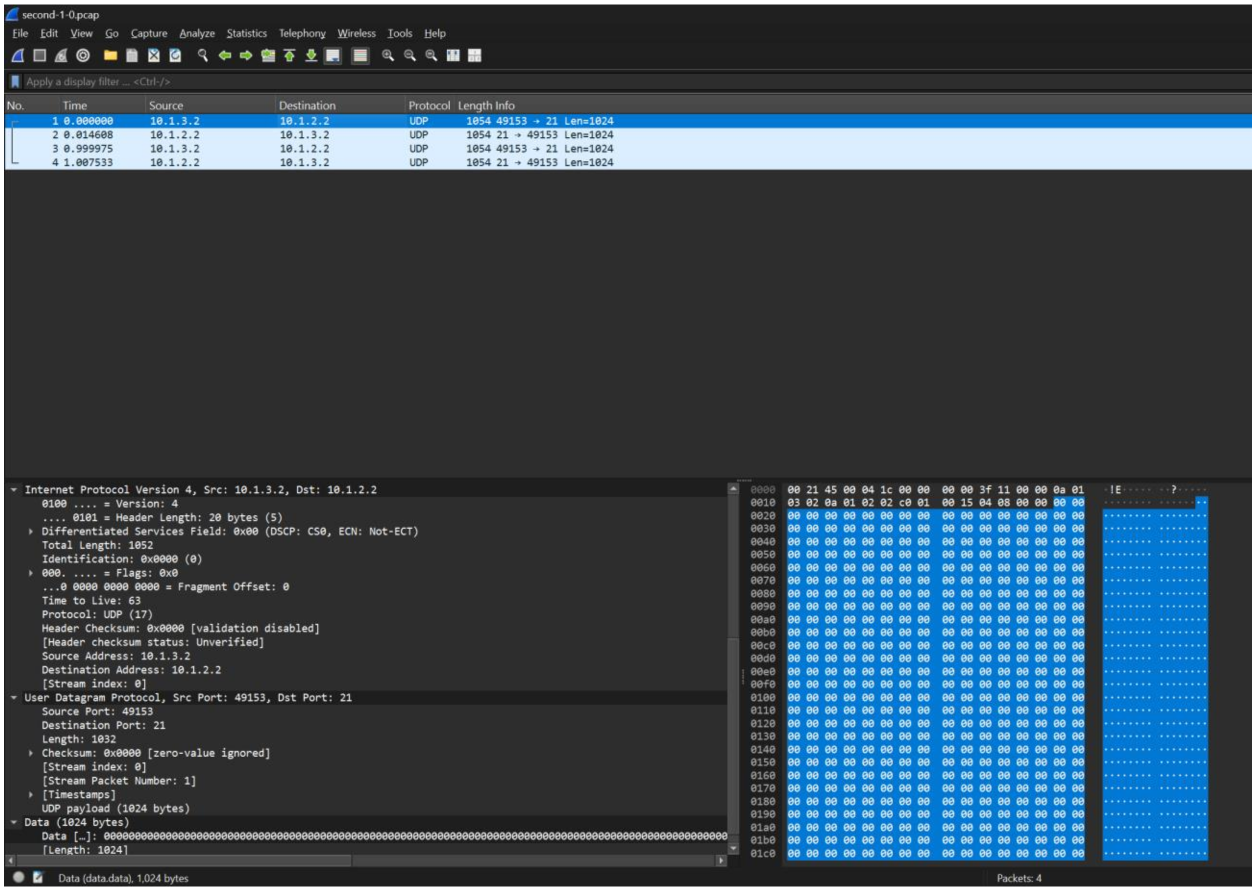
```
No.   Time       Source     Destination  Protocol  Length Info
  2 0.007235   10.1.2.2   10.1.3.2     UDP       1054 21 → 49153 Len=1024
  4 1.000160   10.1.2.2   10.1.3.2     UDP       1054 21 → 49153 Len=1024
  1 0.000000   10.1.3.2   10.1.2.2     UDP       1054 49153 → 21 Len=1024
  3 0.999975   10.1.3.2   10.1.2.2     UDP       1054 49153 → 21 Len=1024
```

```
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1052
    Identification: 0x0000 (0)
  ▸ 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 63
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.1.2.2
    Destination Address: 10.1.3.2
    [Stream index: 0]
▾ User Datagram Protocol, Src Port: 21, Dst Port: 49153
    Source Port: 21
    Destination Port: 49153
    Length: 1032
  ▸ Checksum: 0x0000 [zero-value ignored]
    [Stream index: 0]
    [Stream Packet Number: 2]
  ▸ [Timestamps]
    UDP payload (1024 bytes)
▾ Data (1024 bytes)
    Data […]: 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
    [Length: 1024]
```

```
0010  02 02 0a 01 03 02 00 15  c0 01 04 08 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Data (data.data), 1,024 bytes

Figure 6. Wireshark UDP Echo Server-Client interaction on CSMA Network of 3 nodes each in first and second shared bus.



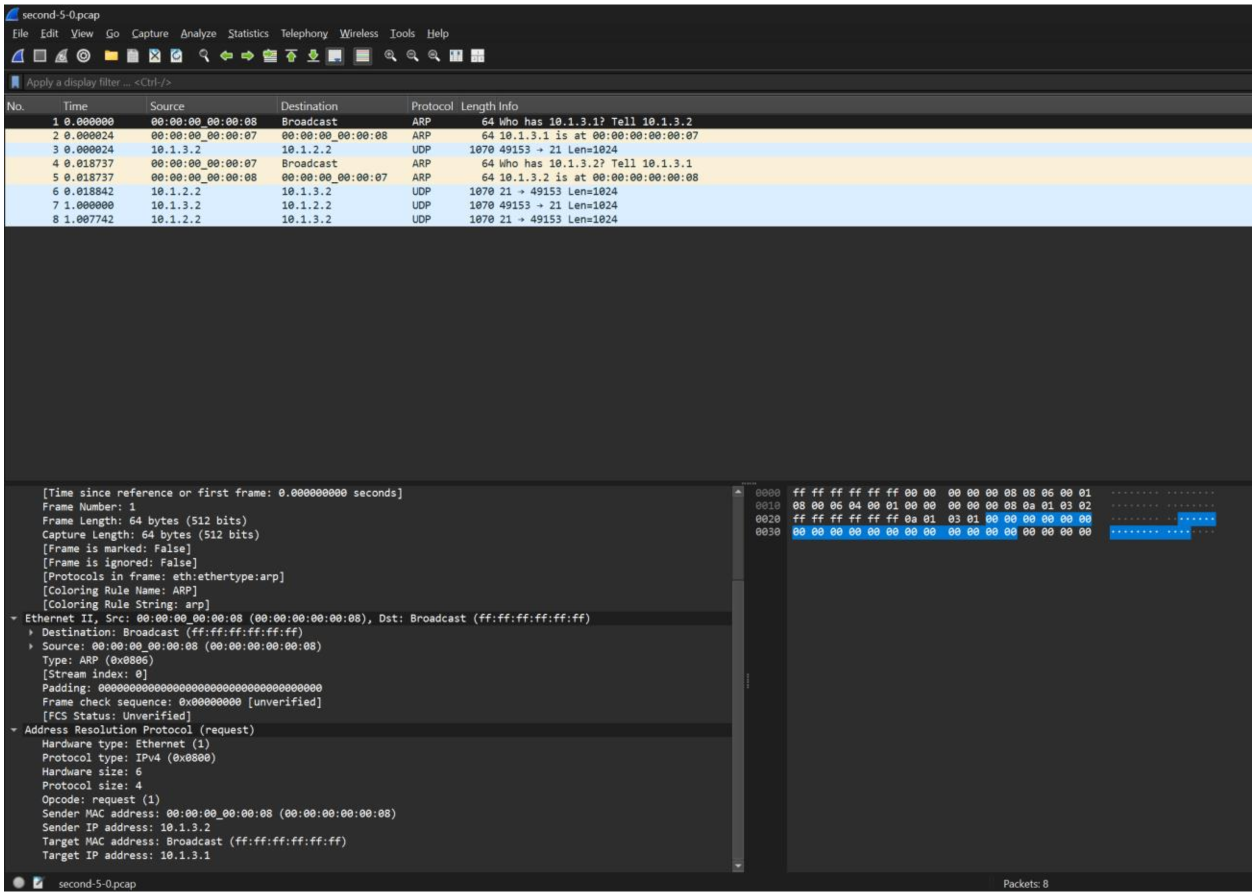
```
second-1-0.pcap

No.  Time      Source    Destination  Protocol  Length Info
   1 0.000000  10.1.3.2  10.1.2.2     UDP       1054 49153 → 21 Len=1024
   2 0.014608  10.1.2.2  10.1.3.2     UDP       1054 21 → 49153 Len=1024
   3 0.999975  10.1.3.2  10.1.2.2     UDP       1054 49153 → 21 Len=1024
   4 1.007533  10.1.2.2  10.1.3.2     UDP       1054 21 → 49153 Len=1024
```

```
Internet Protocol Version 4, Src: 10.1.3.2, Dst: 10.1.2.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1052
    Identification: 0x0000 (0)
    000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 63
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.1.3.2
    Destination Address: 10.1.2.2
    [Stream index: 0]
User Datagram Protocol, Src Port: 49153, Dst Port: 21
    Source Port: 49153
    Destination Port: 21
    Length: 1032
    Checksum: 0x0000 [zero-value ignored]
    [Stream index: 0]
    [Stream Packet Number: 1]
    [Timestamps]
    UDP payload (1024 bytes)
Data (1024 bytes)
    Data […]: 000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
    [Length: 1024]
```

```
0000  00 21 45 00 04 1c 00 00  00 00 3f 11 00 00 0a 01   ·!E····· ··?·····
0010  03 02 0a 01 02 02 c0 01  00 15 04 08 00 00 00 00   ········ ········
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

```
Data (data.data), 1,024 bytes                                  Packets: 4
```

Figure 7: Wireshark UDP Echo Server-Client interaction on CSMA Network of 3 nodes each in first and second shared bus.



```
No.  Time      Source               Destination          Protocol Length Info
   1 0.000000  00:00:00_00:00:08    Broadcast            ARP        64 Who has 10.1.3.1? Tell 10.1.3.2
   2 0.000024  00:00:00_00:00:07    00:00:00_00:00:08    ARP        64 10.1.3.1 is at 00:00:00:00:00:07
   3 0.000024  10.1.3.2             10.1.2.2             UDP      1070 49153 → 21 Len=1024
   4 0.018737  00:00:00_00:00:07    Broadcast            ARP        64 Who has 10.1.3.2? Tell 10.1.3.1
   5 0.018737  00:00:00_00:00:08    00:00:00_00:00:07    ARP        64 10.1.3.2 is at 00:00:00:00:00:08
   6 0.018842  10.1.2.2             10.1.3.2             UDP      1070 21 → 49153 Len=1024
   7 1.000000  10.1.3.2             10.1.2.2             UDP      1070 49153 → 21 Len=1024
   8 1.007742  10.1.2.2             10.1.3.2             UDP      1070 21 → 49153 Len=1024
```

```
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 64 bytes (512 bits)
    Capture Length: 64 bytes (512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
Ethernet II, Src: 00:00:00_00:00:08 (00:00:00:00:00:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    Source: 00:00:00_00:00:08 (00:00:00:00:00:08)
    Type: ARP (0x0806)
    [Stream index: 0]
    Padding: 000000000000000000000000000000000000
    Frame check sequence: 0x00000000 [unverified]
    [FCS Status: Unverified]
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 00:00:00_00:00:08 (00:00:00:00:00:08)
    Sender IP address: 10.1.3.2
    Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
    Target IP address: 10.1.3.1
```

```
0000  ff ff ff ff ff ff 00 00  00 00 00 08 08 06 00 01   ........ ........
0010  08 00 06 04 00 01 00 00  00 00 00 08 0a 01 03 02   ........ ........
0020  ff ff ff ff ff ff 0a 01  03 01 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

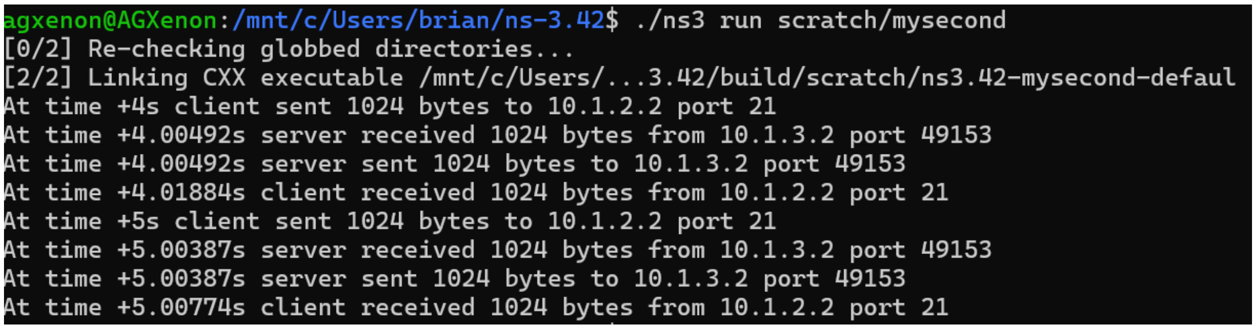
second-5-0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
second-5-0.pcap
Packets: 8

Figure 8. Wireshark UDP Echo Server-Client interaction on CSMA Network of 3 nodes each in first and second shared bus.

Lessons Learnt

The objective of this first assignment is to become familiar with the ns-3 working environment and the simulation workflow. More specifically, understanding the network topologies. CSMA and point-to-point networks were learned. The setup for a CSMA network was learned along with the communication between two nodes via point-to-point. We learned that nodes can be connected through multiple interfaces with Nodes 2 and 3 acting as the bridge between the CSMA buses with a point-to-point link.

By adjusting the time interval from 1 sec to 3sec echoClient.SetAttribute("Interval", TimeValue(Seconds(3.0))) and the stopping time from 7sec to 10 sec ‘clientApps.Stop(Seconds(10.0))’ . we were able to get desired results in Figure 5. Before, with 1sec and 7sec interval and stop time respectively, we had results shown outputs shown in Figure 9 (slightly wrong output on first attempt in editing program).



```
agxenon@AGXenon:/mnt/c/Users/brian/ns-3.42$ ./ns3 run scratch/mysecond
[0/2] Re-checking globbed directories...
[2/2] Linking CXX executable /mnt/c/Users/...3.42/build/scratch/ns3.42-mysecond-defaul
At time +4s client sent 1024 bytes to 10.1.2.2 port 21
At time +4.00492s server received 1024 bytes from 10.1.3.2 port 49153
At time +4.00492s server sent 1024 bytes to 10.1.3.2 port 49153
At time +4.01884s client received 1024 bytes from 10.1.2.2 port 21
At time +5s client sent 1024 bytes to 10.1.2.2 port 21
At time +5.00387s server received 1024 bytes from 10.1.3.2 port 49153
At time +5.00387s server sent 1024 bytes to 10.1.3.2 port 49153
At time +5.00774s client received 1024 bytes from 10.1.2.2 port 21
```

Figure 9: results with 1sec and 7sec interval and stop time respectively.

Overall, we were able to learn and practice more ns3 simulator workflow, by changing some parameters like time intervals and values. By generating the output in .pcap file format, we were able to visualize and analyze packet traffic details in Wireshark, offering an opportunity to revisit fundamental computer networking concepts. We got the chance to revisit Wireshark and recall its components, we believe this will help us to succeed in the coming classwork.