

PRACTICAL RECORD
of
ADVANCED COMPUTER NETWORKS LABORTATORY
(CSPE-352)
ACADEMIC YEAR: 2020-21



SUBMITTED BY:

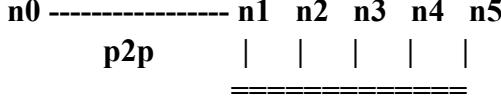
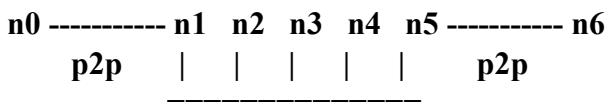
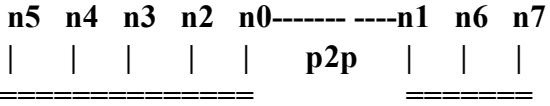
*RAJAN KATARIA
18103076
CSE / 6TH SEMESTER
GROUP: G4*

SUBMITTED TO:

*Dr. KUNWAR PAL
ASSISTANT PROFESSOR
CSE DEPARTMENT*

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
DR. B.R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY,
JALANDHAR - 144011

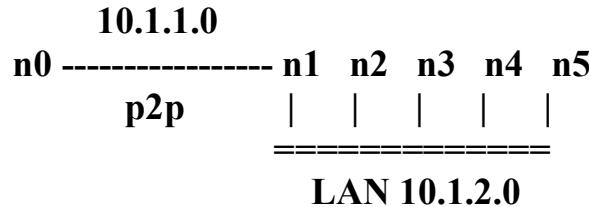
TABLE OF CONTENTS

Sr. No.	Description of the Practical	Date of Practical	Page No.	Signature /Remarks
7.	<p>i. Create a simple topology:</p> <p style="text-align: center;">10.1.1.0</p>  <p style="text-align: center;">LAN 10.1.2.0</p> <p>a. Create pcap file for each node. b. Analyze pcap file via Wireshark and tcpdump. c. Present the node structure and working using Network Animator. d. Create Ascii Trace file and execute analysis with Tracemetrics.</p>	24/02/2021	1-10	
	<p>ii. Create a simple topology:</p> <p style="text-align: center;">10.1.1.0 10.1.3.0</p>  <p style="text-align: center;">LAN 10.1.2.0</p> <p>a. Create pcap file for each node. b. Analyze pcap file via Wireshark and tcpdump. c. Present the node structure and working using Network Animator. d. Create Ascii Trace file and execute analysis with Tracemetrics.</p>	24/02/2021	11-19	
	<p>iii. Create a simple topology:</p> <p style="text-align: center;">10.1.1.0</p>  <p style="text-align: center;">LAN 10.1.2.0 LAN 10.1.3.0</p> <p>a. Create pcap file for each node. b. Analyze pcap file via Wireshark and tcpdump. c. Present the node structure and working using Network Animator. d. Create Ascii Trace file and execute analysis with Tracemetrics.</p>	24/02/2021	20-29	



Lab Assignment - 7

7.1 Create a simple topology:



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"
#include "ns3/netanim-module.h"

//      Network Topology
//
//      10.1.1.0
//      n0 ----- n1 n2 n3 n4 n5
//      p2p     | | | | |
//                  ===== LAN 10.1.2.0

// Client: n0, Server: n5

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("SecondScriptExample");

int
main (int argc, char *argv[])
{
    bool verbose = true;
    uint32_t nCsma = 4;

    CommandLine cmd (__FILE__);
    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;
p2pNodes.Create (2);

NodeContainer csmaNodes;
csmaNodes.Add (p2pNodes.Get (1));
csmaNodes.Create (nCsma);

//********************************************************************

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

NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);

CsmaHelper csma; // LAN has more bandwidth and lesser delay
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

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

//********************************************************************

InternetStackHelper stack;
stack.Install (p2pNodes.Get (0));
stack.Install (csmaNodes);

//********************************************************************

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);
```



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

/***********************/

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (nCsma));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

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));

/***********************/

//Populate all the routing tables to all the nodes.
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

/***********************/

pointToPoint.EnablePcapAll ("mysecond_p2p");
csma.EnablePcapAll ("mysecond_csma");

/***********************/

AnimationInterface anim("mysecond.xml");
anim.SetConstantPosition(p2pNodes.Get(0),10.0,10.0);
anim.SetConstantPosition(csmaNodes.Get(0),20.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(1),30.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(2),40.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(3),50.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(4),60.0,20.0);

/***********************/

AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("mysecond_p2p.tr"));
csma.EnableAsciiAll(ascii.CreateFileStream("mysecond_csma.tr"));
```



```
*****
Simulator::Run ();
Simulator::Destroy ();

*****
return 0;
}
```

Terminal Output:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/mysecond.cc
Waf: Entering directory `/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build'
[2885/2887] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/fd-net-device/ns3.32-tap-device-creator-debug
[2890/2891] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/fd-net-device/ns3.32-raw-sock-creator-debug
[2894/2894] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/tap-bridge/ns3.32-tap-creator-debug
Waf: Leaving directory `/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (40.397s)

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:5 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
AnimationInterface WARNING:Node:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
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
```

The following files were created after running the above program in ns-3.32 folder:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ ls
AUTHORS      README.md          build      mysecond_csmma-1-1.pcap   mysecond_csmma.tr      sri        utils.py    wutils.py
CHANGES.html  RELEASE_NOTES    config     mysecond_csmma-2-0.pcap   mysecond_p2p-0-0.pcap  test.py
CONTRIBUTING.md VERSION        examples   mysecond_csmma-3-0.pcap   mysecond_p2p-1-0.pcap  testpy-output
LICENSE       pycache          mysecond.xml mysecond_csmma-4-0.pcap   mysecond_p2p.tr      testpy.supp
Makefile      sindhangu        mysecond_csmma-5-0.pcap   mysecond_p2p.tr      waf

```

a. Create pcap file for each node.

The below written commands has been used in the above program to create pcap files for all the nodes.

```
106 ****
107
108 pointToPoint.EnablePcapAll ("mysecond_p2p");
109 csma.EnablePcapAll ("mysecond_csmma");
110 ****
```

b. Analyse pcap file via Wireshark and tcpdump.

To analyse the pcap files using Wireshark, write wireshark<space><pcap file name> in the terminal, and press enter (as shown below).



```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_csma-1-1.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_csma-1-1.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_csma-2-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_csma-3-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_csma-4-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_p2p-0-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark mysecond_p2p-0-1.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
```

Alternatively, you can simply write wireshark on the terminal and open Wireshark GUI and then, click File > Open File. And, then choose the file from the directory, and press enter.

The Wireshark window will show you different analysis of the respective pcap file, which includes Frame, Ethernet details and ARP for the nodes of CSMA ,and Frame, Point-To-Point Protocol, Internet Protocol, UDP details for the nodes of p2p, and data as shown for both the pcap files below.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:00:00_00:00:03	Broadcast	ARP	64	Who has 10.1.2.5? Tell 10.1.2.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.000024	00:00:00_00:00:07	00:00:00_00:00:03	ARP	64	10.1.2.5 is at 00:00:00:00:00:07 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
3	0.000024	10.1.1.1	10.1.2.5	UDP	1078	49153 - 9 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
4	0.009129	00:00:00_00:00:07	Broadcast	ARP	64	Who has 10.1.2.1? Tell 10.1.2.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
5	0.009129	00:00:00_00:00:03	00:00:00_00:00:07	ARP	64	10.1.2.1 is at 00:00:00:00:00:03 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
6	0.009235	10.1.2.5	10.1.1.1	UDP	1076	9 - 49153 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 00:00:00:00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 Source: 00:00:00_00:00:03 (00:00:00:00:00:03)
 Type: ARP (0x0806)
 Padding: 00000000000000000000000000000000
 Frame check sequence: 0x00000000 incorrect, should be 0x539fc5f7
 [FCS Status: Bad]
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_00:00:03 (00:00:00:00:00:03)
 Sender IP address: 10.1.2.1
 Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
 Target IP address: 10.1.2.5

0000 ff ff ff ff ff 00 00 00 00 00 03 00 06 00 01
 0010 08 00 06 04 00 01 00 00 00 00 03 00 01 02 01
 0020 ff ff ff ff ff 0a 01 02 05 00 00 00 00 00 00
 0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



mysecond_csma-2-0.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:00:00_00:00:03	Broadcast	ARP	64	Who has 10.1.2.5? Tell 10.1.2.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.009117	00:00:00_00:00:07	Broadcast	ARP	64	Who has 10.1.2.1? Tell 10.1.2.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 00:00:00_00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 Source: 00:00:00_00:00:03 (00:00:00:00:00:03)
 Type: ARP (0x0806)
 Padding: 00
 Frame check sequence: 0x00000000 incorrect, should be 0x539fc5f7
 [FCS Status: Bad]
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_00:00:03 (00:00:00:00:00:03)
 Sender IP address: 10.1.2.1
 Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
 Target IP address: 10.1.2.5

mysecond_csma-3-0.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:00:00_00:00:03	Broadcast	ARP	64	Who has 10.1.2.5? Tell 10.1.2.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.009117	00:00:00_00:00:07	Broadcast	ARP	64	Who has 10.1.2.1? Tell 10.1.2.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 00:00:00_00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 Source: 00:00:00_00:00:03 (00:00:00:00:00:03)
 Type: ARP (0x0806)
 Padding: 00
 Frame check sequence: 0x00000000 incorrect, should be 0x539fc5f7
 [FCS Status: Bad]
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_00:00:03 (00:00:00:00:00:03)
 Sender IP address: 10.1.2.1
 Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
 Target IP address: 10.1.2.5

mysecond_csma-4-0.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:00:00_00:00:03	Broadcast	ARP	64	Who has 10.1.2.5? Tell 10.1.2.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.009117	00:00:00_00:00:07	Broadcast	ARP	64	Who has 10.1.2.1? Tell 10.1.2.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 00:00:00_00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 Source: 00:00:00_00:00:03 (00:00:00:00:00:03)
 Type: ARP (0x0806)
 Padding: 00
 Frame check sequence: 0x00000000 incorrect, should be 0x539fc5f7
 [FCS Status: Bad]
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_00:00:03 (00:00:00:00:00:03)
 Sender IP address: 10.1.2.1
 Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
 Target IP address: 10.1.2.5



mysecond_csma-5.0.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	00:00:00:00:00:03	Broadcast	ARP	64 Who has 10.1.2.5? Tell 10.1.2.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.000000	00:00:00:00:00:07	00:00:00:00:00:03	ARP	64 10.1.2.5 is at 00:00:00:00:00:07 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
3	0.0000105	10.1.1.1	10.1.2.5	UDP	1070 49153 - 9 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
4	0.009105	00:00:00:00:00:07	Broadcast	ARP	64 Who has 10.1.2.1? Tell 10.1.2.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
5	0.009130	00:00:00:00:00:03	00:00:00:00:00:07	ARP	64 10.1.2.1 is at 00:00:00:00:00:03 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
6	0.009130	10.1.2.5	10.1.1.1	UDP	1070 9 - 49153 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)

Ethernet II, Src: 00:00:00:00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

- ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
- ▶ Source: 00:00:00_00:00:03 (00:00:00:00:00:03)
- Type: ARP (0x0806)
- Padding: 00000000000000000000000000000000
- ▶ Frame check sequence: 0x00000000 incorrect, should be 0x539fc5f7

[FCS Status: Bad]

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_00:00:03 (00:00:00:00:00:03)
Sender IP address: 10.1.2.1
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
Target IP address: 10.1.2.5

0000 ff ff ff ff ff 00 00 00 03 08 06 00 01
0010 08 00 06 04 00 01 00 00 00 03 00 01 02 01
0020 ff ff ff ff ff 0a 01 02 05 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00



Before the graphical user interface of Wireshark, the pcap files were analysed using tcpdump command as shown below:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_csma-1-1.pcap
[sudo] password for rajan:
reading from file mysecond_csma-1-1.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.5 (ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Reply 10.1.2.5 is-at 00:00:00:00:07, length 50
IP 10.1.1.1.49153 > 10.1.2.5.9: UDP, length 1024
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff) tell 10.1.2.5, length 50
ARP, Reply 10.1.2.1 is-at 00:00:00:00:03, length 50
IP 10.1.2.5.9 > 10.1.1.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_csma-2-0.pcap
reading from file mysecond_csma-2-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.5 (ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff) tell 10.1.2.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_csma-3-0.pcap
reading from file mysecond_csma-3-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.5 (ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff) tell 10.1.2.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_csma-4-0.pcap
reading from file mysecond_csma-4-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.5 (ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff) tell 10.1.2.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_csma-5-0.pcap
reading from file mysecond_csma-5-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.5 (ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Reply 10.1.2.5 is-at 00:00:00:00:07, length 50
IP 10.1.1.1.49153 > 10.1.2.5.9: UDP, length 1024
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff) tell 10.1.2.5, length 50
ARP, Reply 10.1.2.1 is-at 00:00:00:00:03, length 50
IP 10.1.2.5.9 > 10.1.1.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_p2p-0-0.pcap
reading from file mysecond_p2p-0-0.pcap, link-type PPP (PPP)
IP 10.1.1.1.49153 > 10.1.2.5.9: UDP, length 1024
IP 10.1.2.5.9 > 10.1.1.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r mysecond_p2p-1-0.pcap
reading from file mysecond_p2p-1-0.pcap, link-type PPP (PPP)
IP 10.1.1.1.49153 > 10.1.2.5.9: UDP, length 1024
IP 10.1.2.5.9 > 10.1.1.1.49153: UDP, length 1024
```

Note: If some error comes, try using sudo in front of tcpdump while writing the command on the terminal.

c. Present the node structure and working using Network Animator.

If you want to analyse the node structure using animation, in NetAnim (Network Animator), you need to make xml file for your C++ code in ns-3.

This can be formed using the below written code lines in end of the C++ program as shown. The arguments of SetConstantPosition function show the coordinates of nodes to be shown on the grid in the Network Animator.

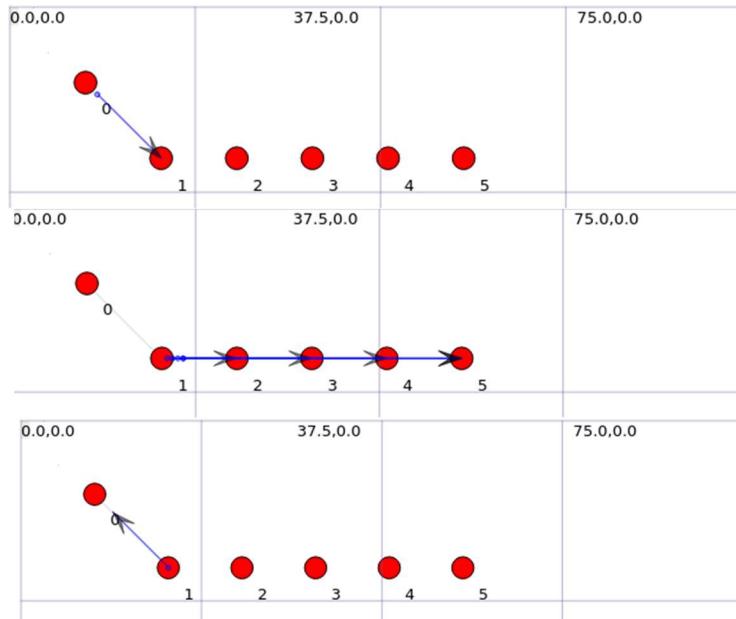
```
111 //*****
112
113 AnimationInterface anim("mysecond.xml");
114 anim.SetConstantPosition(p2pNodes.Get(0),10.0,10.0);
115 anim.SetConstantPosition(csmaNodes.Get(0),20.0,20.0);
116 anim.SetConstantPosition(csmaNodes.Get(1),30.0,20.0);
117 anim.SetConstantPosition(csmaNodes.Get(2),40.0,20.0);
118 anim.SetConstantPosition(csmaNodes.Get(3),50.0,20.0);
119 anim.SetConstantPosition(csmaNodes.Get(4),60.0,20.0);
120
121 //*****
```

Now to run xml file of your C++ program in NetAnim, follow the below written steps, i.e., go in the netanim-3.108 directory, and write ./NetAnim command as shown:



```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ cd ..
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32$ cd netanim-3.108
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/netanim-3.108$ ./NetAnim
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
```

The NetAnim GUI will open, just select your xml file from the directory, and press play button. The animation will play. The screenshots of node 0 (client) sending packet to the node 5 (server) and server sending acknowledgement back to the client are shown below.



d. Create Ascii Trace file and execute analysis with Tracemetrics.

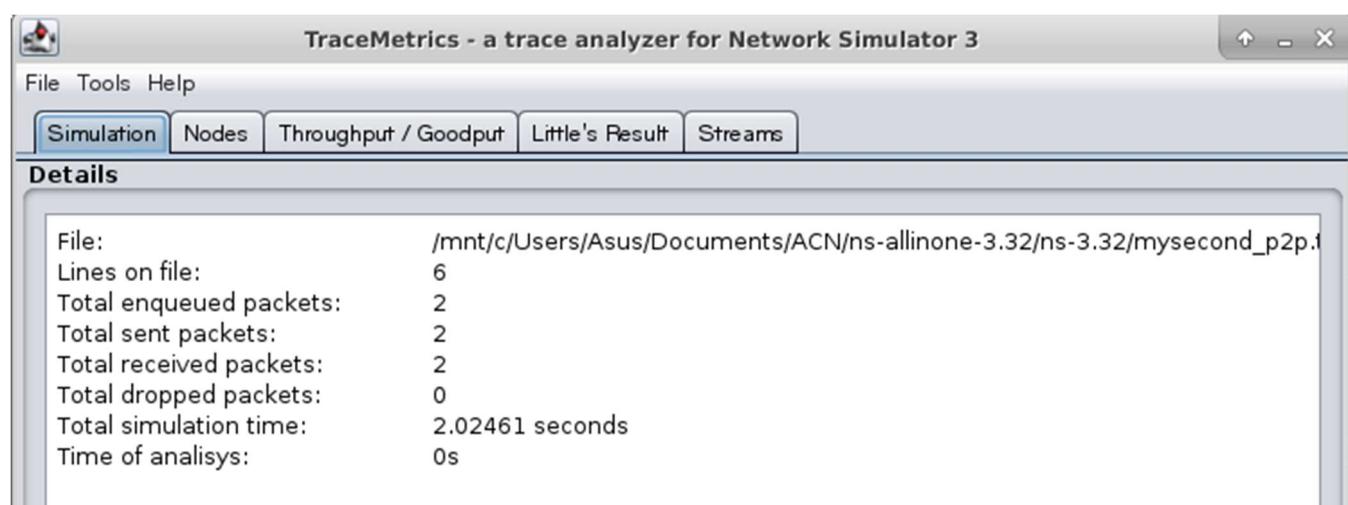
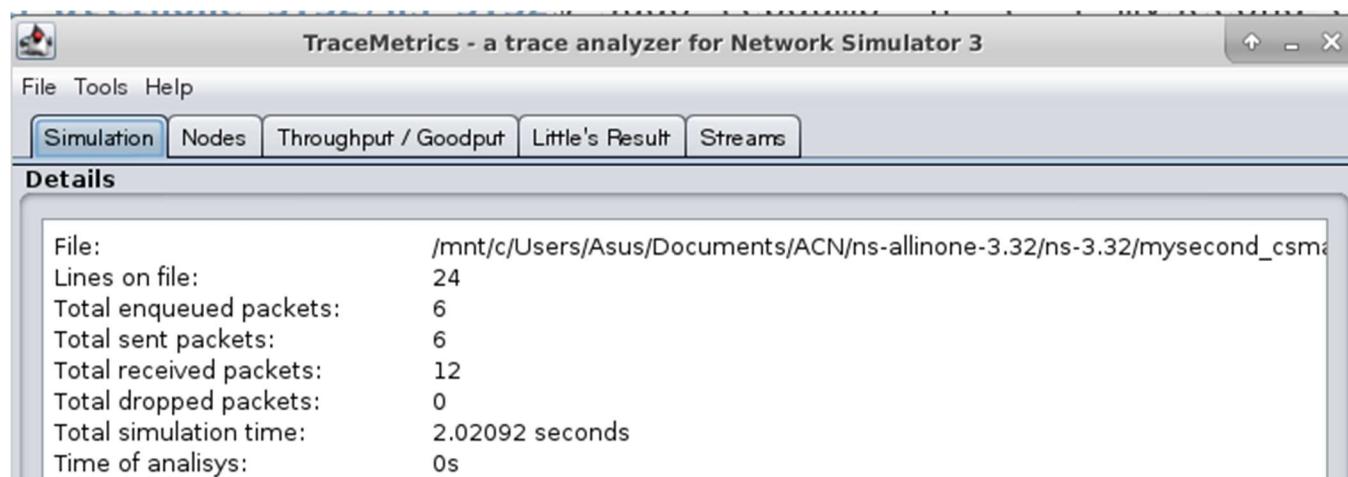
The ASCII trace file is made using the below mentioned command:

```
121 //*****
122 AsciiTraceHelper ascii;
123 pointToPoint.EnableAsciiAll(ascii.CreateFileStream("mysecond_p2p.tr"));
124 csma.EnableAsciiAll(ascii.CreateFileStream("mysecond_csma.tr"));
125 //*****
```

To run TraceMetrics - trace analyzer, run the following command in the directory where you have unzipped/extracted the tracemetrics.zip file.

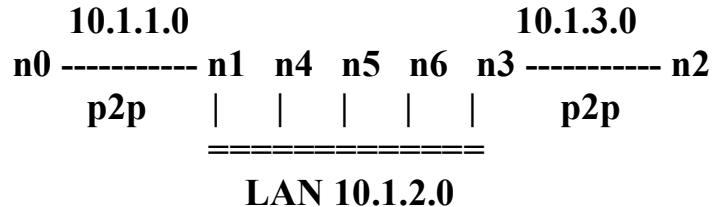
```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32$ java -jar tracemetrics.jar
TCP size: 0
```

The GUI of TraceMetrics - a trace analyzer for NS3 will open, you will have to select the trace file created by you in the program using File > Open from the Menu bar. And then, all the details of Simulation, Nodes, Throughput/Goodput Little's Result, and Streams will be available in the trace analyzer. You can view that by clicking on the respective button.





7.2 Create a simple topology:



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"
#include "ns3/netanim-module.h"

// Network Topology
//
//      10.1.1.0          10.1.3.0
//      n0 ----- n1  n4  n5  n6  n3 ----- n2
//      p2p    |    |    |    |    |    p2p
//      =====
//                      LAN 10.1.2.0

// Client: n2, Server: n0

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("SecondScriptExampleModified");

int
main (int argc, char *argv[])
{
    bool verbose = true;
    uint32_t nCsma = 3;

    CommandLine cmd (__FILE__);
    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;  
p2pNodes.Create (2);  
  
NodeContainer p2pNodes1;  
p2pNodes1.Create (2);  
  
NodeContainer csmaNodes;  
csmaNodes.Add (p2pNodes.Get (1));  
csmaNodes.Add (p2pNodes1.Get (1));  
csmaNodes.Create (nCsma);  
  
/******************************************/  
  
PointToPointHelper pointToPoint;  
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));  
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));  
  
NetDeviceContainer p2pDevices;  
p2pDevices = pointToPoint.Install (p2pNodes);  
  
PointToPointHelper pointToPoint1;  
pointToPoint1.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));  
pointToPoint1.SetChannelAttribute ("Delay", StringValue ("2ms"));  
  
NetDeviceContainer p2pDevices1;  
p2pDevices1 = pointToPoint1.Install (p2pNodes1);  
  
CsmaHelper csma; // LAN has more bandwidth and lesser delay  
csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));  
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));  
  
NetDeviceContainer csmaDevices;  
csmaDevices = csma.Install (csmaNodes);  
  
/******************************************/  
  
InternetStackHelper stack;  
stack.Install (p2pNodes.Get (0));
```



```
stack.Install (p2pNodes1.Get (0));
stack.Install (csmaNodes);

/***********************/

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces1;
p2pInterfaces1 = address.Assign (p2pDevices1);

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

/***********************/

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (p2pNodes.Get (0));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (p2pInterfaces.GetAddress (0), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (p2pNodes1.Get (0));

clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (4.0));

/***********************/

//Populate all the routing tables to all the nodes.
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

/***********************/

pointToPoint.EnablePcapAll ("lab7_2_p2p");
pointToPoint1.EnablePcapAll ("lab7_2_p2p1");
csma.EnablePcapAll ("lab7_2_csma");
```



```
*****
```

```
AnimationInterface anim("lab7_2.xml");
anim.SetConstantPosition(p2pNodes.Get(0),10.0,10.0);
anim.SetConstantPosition(p2pNodes.Get(1),20.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(2),30.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(3),40.0,20.0);
anim.SetConstantPosition(csmaNodes.Get(4),50.0,20.0);
anim.SetConstantPosition(p2pNodes1.Get(1),60.0,20.0);
anim.SetConstantPosition(p2pNodes1.Get(0),70.0,10.0);
```

```
*****
```

```
AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("lab7_2_p2p.tr"));
csma.EnableAsciiAll(ascii.CreateFileStream("lab7_2_csma.tr"));
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("lab7_2_p2p1.tr"));
```

```
*****
```

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

```
*****
```

```
return 0;
}
```

Terminal Output:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/lab7_2.cc
Waf: Entering directory `/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build'
[2885/2887] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/fd-net-device/ns3.32-tap-device-creator-debug
[sudo] password for rajan:
[2890/2891] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/fd-net-device/ns3.32-raw-sock-creator-debug
[2894/2894] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/tap-bridge/ns3.32-tap-creator-debug
Waf: Leaving directory `/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1m29.842s)

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:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:6 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
AnimationInterface WARNING:Node:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:6 Does not have a mobility model. Use SetConstantPosition if it is stationary
At time +2s client sent 1024 bytes to 10.1.1.1 port 9
At time +2.01449s server received 1024 bytes from 10.1.3.1 port 49153
At time +2.01449s server sent 1024 bytes to 10.1.3.1 port 49153
At time +2.02498s client received 1024 bytes from 10.1.1.1 port 9
```



The following files were created after running the above program in ns-3.32 folder:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ ls
AUTHORS          RELEASE_NOTES    doc           lab7_2_csma-5-0.pcap   lab7_2_p2p-3-0.pcap   lab7_2_p2p1.tr    util      wutils.py
CHANGES.html      VERSION        examples       lab7_2_csma-6-0.pcap   lab7_2_p2p.tr     scratch    utils.py
CONTRIBUTING.md   _pycache_      lab7_2.xml     lab7_2_csma.tr      lab7_2_p2p1-0-0.pcap  scratch    waf
LICENSE          bindings       lab7_2_csma-1-1.pcap  lab7_2_p2p-0-0.pcap  lab7_2_p2p1-1-0.pcap  test.py    waf-tools
Makefile          build         lab7_2_csma-3-1.pcap  lab7_2_p2p-1-0.pcap  lab7_2_p2p1-2-0.pcap  test-output  waf.bat
README.md         config        lab7_2_csma-4-0.pcap  lab7_2_p2p-2-0.pcap  lab7_2_p2p1-3-0.pcap  testpy.supp  wscript
```

a. Create pcap file for each node.

The below written command has been used in the above program to create pcap files for all the nodes.

```
122 //*****
123
124 pointToPoint.EnablePcapAll ("lab7_2_p2p");
125 pointToPoint1.EnablePcapAll ("lab7_2_p2p1");
126 csma.EnablePcapAll ("lab7_2_csma");
127
128 //*****
```

b. Analyse pcap file via Wireshark and tcpdump.

To analyse the pcap files using Wireshark, write `wireshark <space> <pcap file name>` in the terminal, and press enter (as shown below).

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark lab7_2_csma-1-1.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark lab7_2_p2p-0-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark lab7_2_p2p1-0-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
```

Alternatively, you can simply write wireshark on the terminal and open Wireshark GUI and then, click File > Open File. And, then choose the file from the directory, and press enter.

The Wireshark window will show you different analysis of the respective pcap file, which includes Frame, Ethernet details and ARP for the nodes of CSMA, and Frame, Point-To-Point Protocol, Internet Protocol, UDP details for the nodes of p2p, as shown below for some of the pcap files.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	00:00:00_00:00:06	Broadcast	ARP	64 Who has 10.1.2.1? Tell 10.1.2.2 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.000000	00:00:00_00:00:05	00:00:00_00:00:06	ARP	64 10.1.2.1 is at 00:00:00:00:00:05 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
3	0.000185	10.1.3.1	10.1.1.1	UDP	1070 49153 - 9 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
4	0.004787	00:00:00_00:00:05	Broadcast	ARP	64 Who has 10.1.2.2? Tell 10.1.2.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
5	0.005053	00:00:00_00:00:06	00:00:00_00:00:05	ARP	64 10.1.2.2 is at 00:00:00:00:00:06 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
6	0.005063	10.1.1.1	10.1.3.1	UDP	1070 9 - 49153 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 00:00:00_00:00:06 (00:00:00:00:00:06), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: 00:00:00_00:00:06 (00:00:00:00:00:06)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
Frame check sequence: 0x00000000 incorrect, should be 0x7eb33689
[FCS Status: Bad]
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_00:00:06 (00:00:00:00:00:06)
Sender IP address: 10.1.2.2
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
Target IP address: 10.1.2.1



lab7_2_p2p-0-0.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.3.1	10.1.1.1	UDP	1054	49153 → 9 Len=1024
2	0.000000	10.1.1.1	10.1.3.1	UDP	1054	9 → 49153 Len=1024

Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)
 ▾ Point-to-Point Protocol
 Protocol: Internet Protocol version 4 (0x0021)
 ▶ Internet Protocol Version 4, Src: 10.1.3.1, Dst: 10.1.1.1
 ▾ User Datagram Protocol, Src Port: 49153, Dst Port: 9
 Source Port: 49153
 Destination Port: 9
 Length: 1032
 [Checksum: [missing]]
 [Checksum Status: Not present]
 [Stream index: 0]
 ▾ Data (1024 bytes)
 Data: 00...
 [Length: 1024]

lab7_2_p2p1-0-0.pcap

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.3.1	10.1.1.1	UDP	1054	49153 → 9 Len=1024
2	0.000000	10.1.1.1	10.1.3.1	UDP	1054	9 → 49153 Len=1024

Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)
 ▾ Point-to-Point Protocol
 ▾ Internet Protocol Version 4, Src: 10.1.3.1, Dst: 10.1.1.1
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 1052
 Identification: 0x0000 (0)
 ▶ Flags: 0x0000
 Time to live: 62
 Protocol: UDP (17)
 Header checksum: 0x0000 [validation disabled]
 [Header checksum status: Unverified]
 Source: 10.1.3.1
 Destination: 10.1.1.1
 ▾ User Datagram Protocol, Src Port: 49153, Dst Port: 9
 ▾ Data (1024 bytes)

Before the graphical user interface of Wireshark, the pcap files were analysed using tcpdump command as shown below:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_csma-1-1.pcap
reading from file lab7_2_csma-1-1.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.2, length 50
ARP, Reply 10.1.2.1 is-at 00:00:00:00:00:05, length 50
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
ARP, Request who-has 10.1.2.2 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Reply 10.1.2.2 is-at 00:00:00:00:00:06, length 50
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_csma-3-1.pcap
reading from file lab7_2_csma-3-1.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.2, length 50
ARP, Reply 10.1.2.1 is-at 00:00:00:00:00:05, length 50
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
ARP, Request who-has 10.1.2.2 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
ARP, Reply 10.1.2.2 is-at 00:00:00:00:00:06, length 50
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_csma-4-0.pcap
reading from file lab7_2_csma-4-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.2, length 50
ARP, Request who-has 10.1.2.2 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_csma-5-0.pcap
reading from file lab7_2_csma-5-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.2, length 50
ARP, Request who-has 10.1.2.2 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_csma-6-0.pcap
reading from file lab7_2_csma-6-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.2, length 50
ARP, Request who-has 10.1.2.2 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
```



```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p-0-0.pcap
reading from file lab7_2_p2p-0-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p-1-0.pcap
reading from file lab7_2_p2p-1-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p-2-0.pcap
reading from file lab7_2_p2p-2-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p-3-0.pcap
reading from file lab7_2_p2p-3-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p1-0-0.pcap
reading from file lab7_2_p2p1-0-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p1-1-0.pcap
reading from file lab7_2_p2p1-1-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p1-2-0.pcap
reading from file lab7_2_p2p1-2-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_2_p2p1-3-0.pcap
reading from file lab7_2_p2p1-3-0.pcap, link-type PPP (PPP)
IP 10.1.3.1.49153 > 10.1.1.1.9: UDP, length 1024
IP 10.1.1.1.9 > 10.1.3.1.49153: UDP, length 1024
```

Note: If some error comes, try using sudo in front of tcpdump while writing the command on the terminal.

c. Present the node structure and working using Network Animator.

If you want to analyse the node structure using animation, in NetAnim (Network Animator), you need to make xml file for your C++ code in ns-3.

This can be formed using the below written code lines in end of the C++ program as shown. The arguments of SetConstantPosition function show the coordinates of nodes to be shown on the grid in the Network Animator.

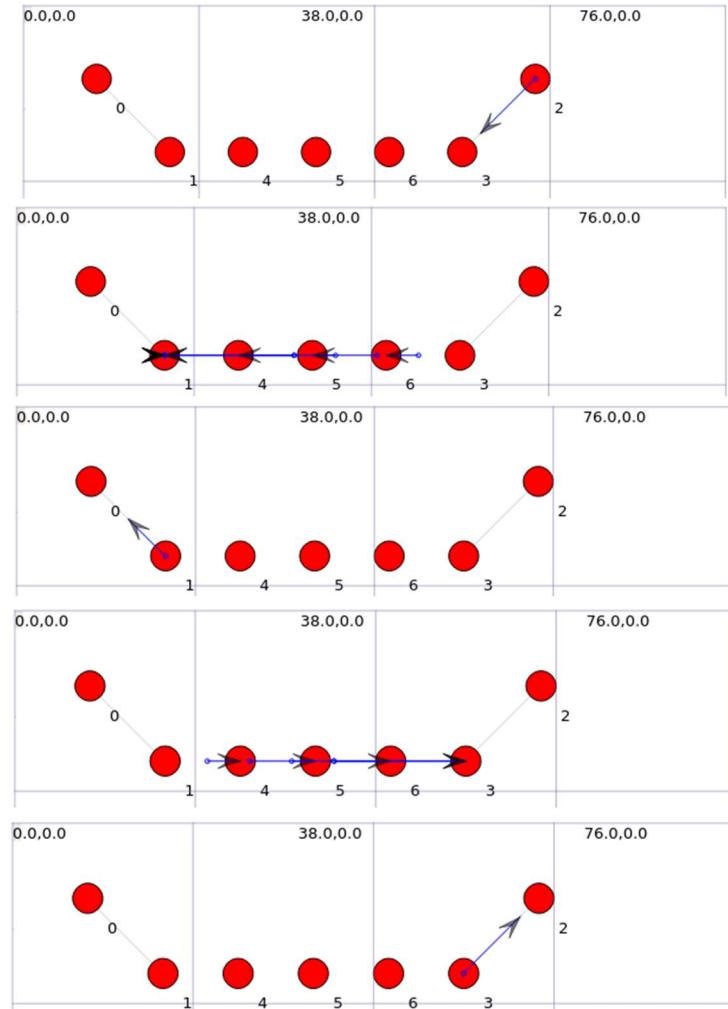
```
128 //*****
129
130 AnimationInterface anim("lab7_2.xml");
131 anim.SetConstantPosition(p2pNodes.Get(0),10.0,10.0);
132 anim.SetConstantPosition(p2pNodes.Get(1),20.0,20.0);
133 anim.SetConstantPosition(csmaNodes.Get(2),30.0,20.0);
134 anim.SetConstantPosition(csmaNodes.Get(3),40.0,20.0);
135 anim.SetConstantPosition(csmaNodes.Get(4),50.0,20.0);
136 anim.SetConstantPosition(p2pNodes1.Get(1),60.0,20.0);
137 anim.SetConstantPosition(p2pNodes1.Get(0),70.0,10.0);
138
139 //*****
```

Now to run xml file of your C++ program in NetAnim, follow the below written steps, i.e., go in the netanim-3.108 directory, and write ./NetAnim command as shown:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ cd ..
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32$ cd netanim-3.108
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/netanim-3.108$ ./NetAnim
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
```



The NetAnim GUI will open, just select your xml file from the directory, and press play button. The animation will play. The screenshots of node 2 (client) sending packet to the node 0 (server) and server sending acknowledgement back to the client are shown below.



d. Create Ascii Trace file and execute analysis with Tracemetrics.

The ASCII trace file is made using the below mentioned command:

```

139 //*****
140
141 AsciiTraceHelper ascii;
142 pointToPoint.EnableAsciiAll(ascii.CreateFileStream("lab7_2_p2p.tr"));
143 csma.EnableAsciiAll(ascii.CreateFileStream("lab7_2_csma.tr"));
144 pointToPoint.EnableAsciiAll(ascii.CreateFileStream("lab7_2_p2p1.tr"));
145
146 //*****

```

To run TraceMetrics - trace analyzer, run the following command in the directory where you have unzipped/extracted the tracemetrics.zip file.



```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32$ java -jar tracemetrics.jar
TCP size: 0
```

The GUI of TraceMetrics - a trace analyzer for NS3 will open, you will have to select the trace file created by you in the program using File > Open from the Menu bar. And then, all the details of Simulation, Nodes, Throughput/Goodput Little's Result, and Streams will be available in the trace analyzer. You can view that that by clicking on the respective button.

The figure consists of three vertically stacked windows of the 'TraceMetrics - a trace analyzer for Network Simulator 3' application. Each window has a title bar, a menu bar (File, Tools, Help), and a tab bar with tabs: Simulation (selected), Nodes, Throughput / Goodput, Little's Result, and Streams. Below the tabs is a 'Details' section containing a table of simulation statistics.

Top Window (lab7_2_csma.tr):

File:	/mnt/c/Users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/lab7_2_csma.tr
Lines on file:	24
Total enqueued packets:	6
Total sent packets:	6
Total received packets:	12
Total dropped packets:	0
Total simulation time:	2.02129 seconds
Time of analysis:	0s

Middle Window (lab7_2_p2p.tr):

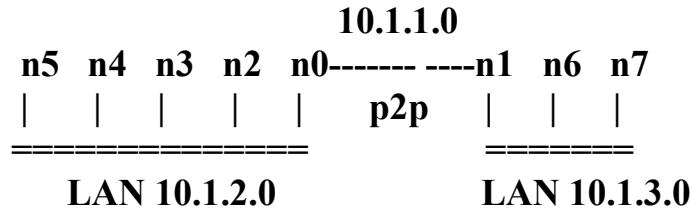
File:	/mnt/c/Users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/lab7_2_p2p.tr
Lines on file:	12
Total enqueued packets:	4
Total sent packets:	4
Total received packets:	4
Total dropped packets:	0
Total simulation time:	2.02498 seconds
Time of analysis:	0s

Bottom Window (lab7_2_p2p1.tr):

File:	/mnt/c/Users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/lab7_2_p2p1.tr
Lines on file:	12
Total enqueued packets:	4
Total sent packets:	4
Total received packets:	4
Total dropped packets:	0
Total simulation time:	2.02498 seconds
Time of analysis:	0s



7.3 Create a simple topology:



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"
#include "ns3/netanim-module.h"

//           Network Topology
//           10.1.1.0
//           n5  n4  n3  n2  n0-----n1  n6  n7
//           |      |      |      |      |
//           =====   =====   =====
//           LAN 10.1.2.0          LAN 10.1.3.0

// Client: n7, Server: n5

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("SecondScriptExample");

int
main (int argc, char *argv[])
{
    bool verbose = true;
    uint32_t nCsma1 = 4;
    uint32_t nCsma2 = 2;

    CommandLine cmd (__FILE__);
    cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma1);
    cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma2);
    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);
}

/***********************/

nCsma1 = nCsma1 == 0 ? 1 : nCsma1;
nCsma2 = nCsma2 == 0 ? 1 : nCsma2;

/***********************/

NodeContainer p2pNodes;
p2pNodes.Create (2);

NodeContainer csmaNodes1;
csmaNodes1.Add (p2pNodes.Get (0));
csmaNodes1.Create (nCsma1);

NodeContainer csmaNodes2;
csmaNodes2.Add (p2pNodes.Get (1));
csmaNodes2.Create (nCsma2);

/***********************/

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

NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);

CsmaHelper csma1; // LAN has more bandwidth and lesser delay
csma1.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma1.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

NetDeviceContainer csmaDevices1;
csmaDevices1 = csma1.Install (csmaNodes1);

CsmaHelper csma2; // LAN has more bandwidth and lesser delay
csma2.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csma2.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

NetDeviceContainer csmaDevices2;
csmaDevices2 = csma2.Install (csmaNodes2);
```



```
*****
```

```
InternetStackHelper stack;  
stack.Install (csmaNodes1);  
stack.Install (csmaNodes2);
```

```
*****
```

```
Ipv4AddressHelper address;  
address.SetBase ("10.1.2.0", "255.255.255.0");  
Ipv4InterfaceContainer p2pInterfaces;  
p2pInterfaces = address.Assign (p2pDevices);  
  
address.SetBase ("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer csmaInterfaces1;  
csmaInterfaces1 = address.Assign (csmaDevices1);  
  
address.SetBase ("10.1.3.0", "255.255.255.0");  
Ipv4InterfaceContainer csmaInterfaces2;  
csmaInterfaces2 = address.Assign (csmaDevices2);
```

```
*****
```

```
UdpEchoServerHelper echoServer (9);  
ApplicationContainer serverApps = echoServer.Install (csmaNodes1.Get (nCsma1));  
serverApps.Start (Seconds (1.0));  
serverApps.Stop (Seconds (10.0));
```

```
UdpEchoClientHelper echoClient (csmaInterfaces1.GetAddress (nCsma1), 9);  
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));  
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));  
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));
```

```
ApplicationContainer clientApps = echoClient.Install (csmaNodes2.Get (nCsma2));
```

```
clientApps.Start (Seconds (2.0));  
clientApps.Stop (Seconds (4.0));
```

```
*****
```

```
//Populate all the routing tables to all the nodes.  
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

```
*****
```

```
pointToPoint.EnablePcapAll ("lab7_3_p2p");  
csma1.EnablePcapAll ("lab7_3_csma1");
```



```
csma2.EnablePcapAll ("lab7_2_csma2");

/***********************/

AnimationInterface anim("lab7_3.xml");
anim.SetConstantPosition(csmaNodes1.Get(0),50.0,30.0);
anim.SetConstantPosition(csmaNodes1.Get(1),40.0,30.0);
anim.SetConstantPosition(csmaNodes1.Get(2),30.0,30.0);
anim.SetConstantPosition(csmaNodes1.Get(3),20.0,30.0);
anim.SetConstantPosition(csmaNodes1.Get(4),10.0,30.0);

anim.SetConstantPosition(csmaNodes2.Get(0),60.0,30.0);
anim.SetConstantPosition(csmaNodes2.Get(1),70.0,30.0);
anim.SetConstantPosition(csmaNodes2.Get(2),80.0,30.0);

/***********************/

AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("lab7_3_p2p.tr"));
csma1.EnableAsciiAll(ascii.CreateFileStream("lab7_3_csma1.tr"));
csma2.EnableAsciiAll(ascii.CreateFileStream("lab7_3_csma2.tr"));

/***********************/

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

/***********************/

return 0;
}
```



Terminal Output:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ ./waf --run scratch/lab7_3.cc
Waf: Entering directory '/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build'
[2885/2887] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/fd-net-device/ns3.32-tap-device-creator-debug
[sudo] password for rajan:
[2890/2891] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/fd-net-device/ns3.32-raw-sock-creator-debug
[2894/2894] Running SuidBuild_task
setting uid bit on executable /mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build/src/tap-bridge/ns3.32-tap-creator-debug
Waf: Leaving directory '/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (44.910s)

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:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:6 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:7 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
AnimationInterface WARNING:Node:5 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:6 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:7 Does not have a mobility model. Use SetConstantPosition if it is stationary
At time +2s client sent 1024 bytes to 10.1.1.5 port 9
At time +2.00992s server received 1024 bytes from 10.1.3.3 port 49153
At time +2.00992s server sent 1024 bytes to 10.1.3.3 port 49153
At time +2.02884s client received 1024 bytes from 10.1.1.5 port 9
```

The following files were created after running the above program in ns-3.32 folder:

AUTHORS	VERSION	lab7_3.xml	lab7_3_csmal-0-1.pcap	lab7_3_csmal-6-0.pcap	lab7_3_csmal-7-0.pcap	lab7_3_csmal-4-0.pcap	lab7_3_csmal-5-0.pcap	lab7_3_p2p.tr	utils.py
CHANGES.html	pycache								waf
CONTRIBUTING.md	bindings		lab7_3_csmal-1-1.pcap	lab7_3_csmal.tr	lab7_3_csmal-2-0.pcap	lab7_3_csmal-3-0.pcap	lab7_3_csmal-4-0.pcap	test.py	distutils
LICENSE	build		lab7_3_csmal-2-0.pcap	lab7_3_csmal-0-1.pcap	lab7_3_csmal-1-1.pcap	lab7_3_csmal-2-1.pcap	lab7_3_csmal-2-2-0.pcap	testpy.supp	waf.bat
Makefile	control		lab7_3_csmal-3-0.pcap	lab7_3_csmal-1-1.pcap	lab7_3_csmal-2-1.pcap	lab7_3_csmal-3-0.pcap	lab7_3_csmal-2-0-0.pcap	testpy	wscript
README.md	docs		lab7_3_csmal-4-0.pcap	lab7_3_csmal-2-0.pcap	lab7_3_csmal-2-1.pcap	lab7_3_csmal-4-0.pcap	lab7_3_p2p-0-0.pcap	testpy.supp	wutils.py
RELEASE_NOTES	examples		lab7_3_csmal-5-0.pcap	lab7_3_csmal-3-0.pcap	lab7_3_csmal-4-0.pcap	lab7_3_csmal-5-0.pcap	lab7_3_p2p-1-0.pcap	testpy	

a. Create pcap file for each node.

The below written command has been used in the above program o create pcap files for all the nodes.

```
129 //*****
130
131 pointToPoint.EnablePcapAll ("lab7_3_p2p");
132 csmal.EnablePcapAll ("lab7_3_csmal");
133 csma2.EnablePcapAll ("lab7_3_csma2");
134
135 //*****
```

b. Analyse pcap file via Wireshark and tcpdump.

To analyse the pcap files using Wireshark, write wireshark<space><pcap file name> in the terminal, and press enter (as shown below).

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark lab7_3_csmal-0-1.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark lab7_3_csma2-0-1.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ wireshark lab7_3_p2p-0-0.pcap
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
nl80211 not found.
```

Alternatively, you can simply write wireshark on the terminal and open Wireshark GUI and then, click File > Open File. And, then choose the file from the directory, and press enter.



The Wireshark window will show you different analysis of the respective pcap file, which includes Frame, Ethernet details and ARP for the nodes of CSMA, and Frame, Point-To-Point Protocol, Internet Protocol, UDP details for the nodes of p2p, as shown below for some of the pcap files.

lab7_3_csm1-0-1.pcap

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	00:00:00 00:00:03	Broadcast	ARP	64 Who has 10.1.1.5? Tell 10.1.1.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.000025	00:00:00 00:00:07	00:00:00 00:00:03	ARP	64 10.1.1.5 is at 00:00:00:00:00:07 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
3	0.000025	10.1.3.3	10.1.1.5	UDP	1070 49153 - 9 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
4	0.006130	00:00:00 00:00:07	Broadcast	ARP	64 Who has 10.1.1.1? Tell 10.1.1.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
5	0.006130	00:00:00 00:00:03	00:00:00 00:00:07	ARP	64 10.1.1.1 is at 00:00:00:00:00:03 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
6	0.006235	10.1.1.5	10.1.3.3	UDP	1070 9 - 49153 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
 Encapsulation type: Ethernet (1)
 Arrival Time: Jan 1, 1970 05:30:02.009803000 IST
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 2.009803000 seconds
 [Time delta from previous captured frame: 0.000000000 seconds]
 [Time delta from previous displayed frame: 0.000000000 seconds]
 [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 00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ► Address Resolution Protocol (request)

lab7_3_csm2-0-1.pcap

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	00:00:00 00:00:03	Broadcast	ARP	64 Who has 10.1.1.5? Tell 10.1.1.1 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
2	0.000025	00:00:00 00:00:07	00:00:00 00:00:03	ARP	64 10.1.1.5 is at 00:00:00:00:00:07 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
3	0.000025	10.1.3.3	10.1.1.5	UDP	1070 49153 - 9 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
4	0.006130	00:00:00 00:00:07	Broadcast	ARP	64 Who has 10.1.1.1? Tell 10.1.1.5 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
5	0.006130	00:00:00 00:00:03	00:00:00 00:00:07	ARP	64 10.1.1.1 is at 00:00:00:00:00:03 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]
6	0.006235	10.1.1.5	10.1.3.3	UDP	1070 9 - 49153 Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORRECT]

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
 ► Ethernet II, Src: 00:00:00 00:00:03 (00:00:00:00:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ► Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 ► Source: 00:00:00 00:00:03 (00:00:00:00:00:03)
 Type: ARP (0x0806)
 Padding: 00000000000000000000000000000000
 ► Frame check sequence: 0x00000000 incorrect, should be 0xB1f7cdic
 [FCS Status: Bad]
 ► 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 00:00:03 (00:00:00:00:00:03)
 Sender IP address: 10.1.1.1
 Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
 Target IP address: 10.1.1.5

lab7_3_p2p-0-0.pcap

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000	10.1.3.3	10.1.1.5	UDP	1054 49153 - 9 Len=1024
2	0.007235	10.1.1.5	10.1.3.3	UDP	1054 9 - 49153 Len=1024

Frame 1: 1054 bytes on wire (8432 bits), 1054 bytes captured (8432 bits)
 Point-to-Point Protocol
 ► Internet Protocol Version 4, Src: 10.1.3.3, Dst: 10.1.1.5
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 ► Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 1052
 Identification: 0x0000 (0)
 ► Flags: 0x0000
 Time to live: 63
 Protocol: UDP (17)
 Header checksum: 0x0000 [validation disabled]
 [Header checksum status: Unverified]
 Source: 10.1.3.3
 Destination: 10.1.1.5
 ► User Datagram Protocol, Src Port: 49153, Dst Port: 9
 Data (1024 bytes)



Before the graphical user interface of Wireshark, the pcap files were analysed using tcpdump command as shown below:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-0-1.pcap
reading from file lab7_3_csmal-0-1.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Reply 10.1.1.5 is-at 00:00:00:00:00:07, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
ARP, Reply 10.1.1.1 is-at 00:00:00:00:00:03, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-1-1.pcap
reading from file lab7_3_csmal-1-1.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.3.1 (ff:ff:ff:ff:ff:ff) tell 10.1.3.3, length 50
ARP, Reply 10.1.3.1 is-at 00:00:00:00:00:08, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.3.3 (ff:ff:ff:ff:ff:ff) tell 10.1.3.1, length 50
ARP, Reply 10.1.3.3 is-at 00:00:00:00:00:0a, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-2-0.pcap
reading from file lab7_3_csmal-2-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-3-0.pcap
reading from file lab7_3_csmal-3-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-4-0.pcap
reading from file lab7_3_csmal-4-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-5-0.pcap
reading from file lab7_3_csmal-5-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Reply 10.1.1.5 is-at 00:00:00:00:00:07, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
ARP, Reply 10.1.1.1 is-at 00:00:00:00:00:03, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-6-0.pcap
reading from file lab7_3_csmal-6-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.3.1 (ff:ff:ff:ff:ff:ff) tell 10.1.3.3, length 50
ARP, Request who-has 10.1.3.3 (ff:ff:ff:ff:ff:ff) tell 10.1.3.1, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csmal-7-0.pcap
reading from file lab7_3_csmal-7-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.3.1 (ff:ff:ff:ff:ff:ff) tell 10.1.3.3, length 50
ARP, Reply 10.1.3.1 is-at 00:00:00:00:00:08, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.3.3 (ff:ff:ff:ff:ff:ff) tell 10.1.3.1, length 50
ARP, Reply 10.1.3.3 is-at 00:00:00:00:00:0a, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-1-1.pcap
reading from file lab7_3_csma2-1-1.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.3.1 (ff:ff:ff:ff:ff:ff) tell 10.1.3.3, length 50
ARP, Reply 10.1.3.1 is-at 00:00:00:00:00:08, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.3.3 (ff:ff:ff:ff:ff:ff) tell 10.1.3.1, length 50
ARP, Reply 10.1.3.3 is-at 00:00:00:00:00:0a, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-2-0.pcap
reading from file lab7_3_csma2-2-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-3-0.pcap
reading from file lab7_3_csma2-3-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-4-0.pcap
reading from file lab7_3_csma2-4-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-5-0.pcap
reading from file lab7_3_csma2-5-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.1.5 (ff:ff:ff:ff:ff:ff) tell 10.1.1.1, length 50
ARP, Reply 10.1.1.5 is-at 00:00:00:00:00:07, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.1.1 (ff:ff:ff:ff:ff:ff) tell 10.1.1.5, length 50
ARP, Reply 10.1.1.1 is-at 00:00:00:00:00:03, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-6-0.pcap
reading from file lab7_3_csma2-6-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.3.1 (ff:ff:ff:ff:ff:ff) tell 10.1.3.3, length 50
ARP, Request who-has 10.1.3.3 (ff:ff:ff:ff:ff:ff) tell 10.1.3.1, length 50
```



```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_csma2-7-0.pcap
reading from file lab7_3_csma2-7-0.pcap, link-type EN10MB (Ethernet)
ARP, Request who-has 10.1.3.1 (ff:ff:ff:ff:ff:ff) tell 10.1.3.3, length 50
ARP, 10.1.3.1 is-at 00:00:00:00:00:08, length 50
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
ARP, Request who-has 10.1.3.3 (ff:ff:ff:ff:ff:ff) tell 10.1.3.1, length 50
ARP, Reply 10.1.3.3 is-at 00:00:00:00:00:0a, length 50
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_p2p-0-0.pcap
reading from file lab7_3_p2p-0-0.pcap, link-type PPP (PPP)
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ sudo tcpdump -n -t -r lab7_3_p2p-1-0.pcap
reading from file lab7_3_p2p-1-0.pcap, link-type PPP (PPP)
IP 10.1.3.3.49153 > 10.1.1.5.9: UDP, length 1024
IP 10.1.1.5.9 > 10.1.3.3.49153: UDP, length 1024
```

Note: If some error comes, try using sudo in front of tcpdump while writing the command on the terminal.

c. Present the node structure and working using Network Animator.

If you want to analyse the node structure using animation, in NetAnim (Network Animator), you need to make xml file for your C++ code in ns-3.

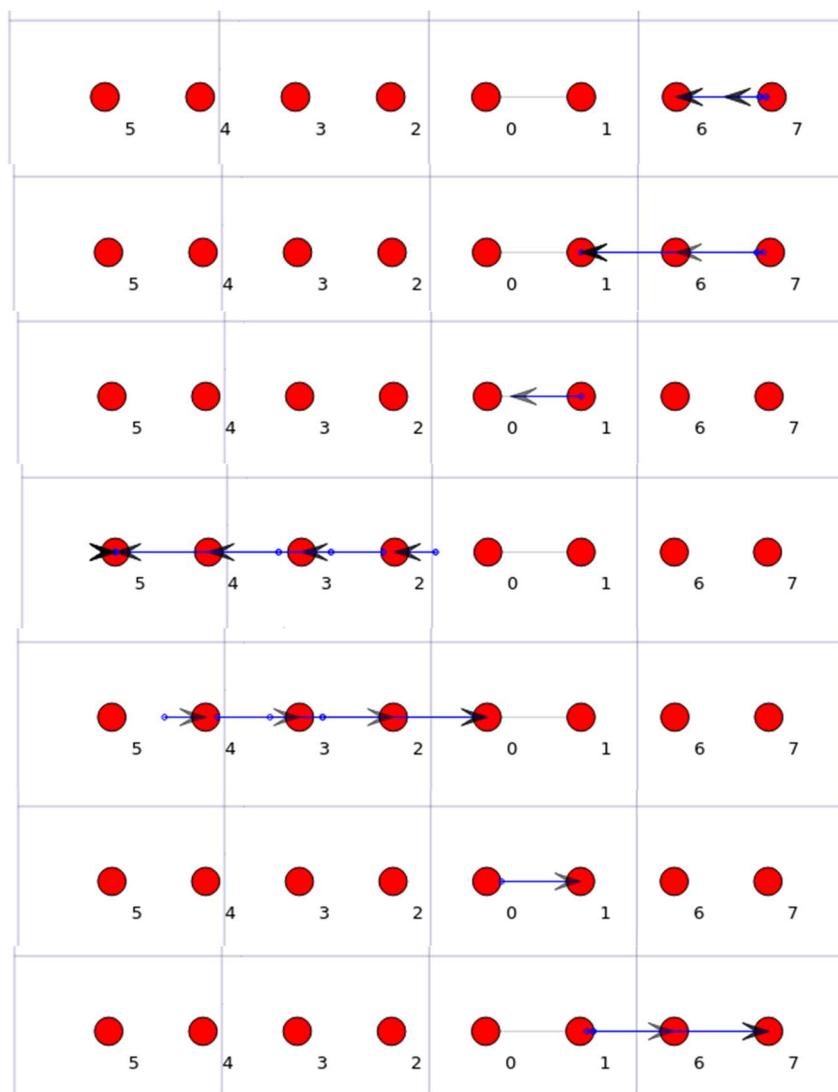
This can be formed using the below written code lines in end of the C++ program as shown. The arguments of SetConstantPosition function show the coordinates of nodes to be shown on the grid in the Network Animator.

```
135 //*****
136
137 AnimationInterface anim("lab7_3.xml");
138 anim.SetConstantPosition(csmaNodes1.Get(0),50.0,30.0);
139 anim.SetConstantPosition(csmaNodes1.Get(1),40.0,30.0);
140 anim.SetConstantPosition(csmaNodes1.Get(2),30.0,30.0);
141 anim.SetConstantPosition(csmaNodes1.Get(3),20.0,30.0);
142 anim.SetConstantPosition(csmaNodes1.Get(4),10.0,30.0);
143
144 anim.SetConstantPosition(csmaNodes2.Get(0),60.0,30.0);
145 anim.SetConstantPosition(csmaNodes2.Get(1),70.0,30.0);
146 anim.SetConstantPosition(csmaNodes2.Get(2),80.0,30.0);
147
148 //*****
```

Now to run xml file of your C++ program in NetAnim, follow the below written steps, i.e., go in the netanim-3.108 directory, and write ./NetAnim command as shown:

```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32$ cd ..
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32$ cd netanim-3.108
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32/netanim-3.108$ ./NetAnim
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rajan'
```

The NetAnim GUI will open, just select your xml file from the directory, and press play button. The animation will play. The screenshots of node 7 (client) sending packet to the node 5 (server) and server sending acknowledgement back to the client are shown below.



d. Create Ascii Trace file and execute analysis with Tracemetrics.

The ASCII trace file is made using the below mentioned command:

```

148 //*****
149
150 AsciiTraceHelper ascii;
151 pointToPoint.EnableAsciiAll(ascii.CreateFileStream("lab7_3_p2p.tr"));
152 csma1.EnableAsciiAll(ascii.CreateFileStream("lab7_3_csma1.tr"));
153 csma2.EnableAsciiAll(ascii.CreateFileStream("lab7_3_csma2.tr"));
154 //*****
155 
```

To run TraceMetrics - trace analyzer, run the following command in the directory where you have unzipped/extracted the tracemetrics.zip file.



```
rajan@RAJAN:/mnt/c/users/Asus/Documents/ACN/ns-allinone-3.32$ java -jar tracemetrics.jar  
TCP size: 0
```

The GUI of TraceMetrics - a trace analyzer for NS3 will open, you will have to select the trace file created by you in the program using File > Open from the Menu bar. And then, all the details of Simulation, Nodes, Throughput/Goodput Little's Result, and Streams will be available in the trace analyzer. You can view that. The screenshots for the above file are attached.

The screenshots show the TraceMetrics interface with the following details:

lab7_3_csmal1.tr Analysis:

File:	/mnt/c/Users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/lab7_3_csmal1.tr
Lines on file:	44
Total enqueued packets:	12
Total sent packets:	12
Total received packets:	20
Total dropped packets:	0
Total simulation time:	2.02884 seconds
Time of analysis:	0s

lab7_3_csma2.tr Analysis:

File:	/mnt/c/Users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/lab7_3_csma2.tr
Lines on file:	44
Total enqueued packets:	12
Total sent packets:	12
Total received packets:	20
Total dropped packets:	0
Total simulation time:	2.02884 seconds
Time of analysis:	0s

lab7_3_p2p.tr Analysis:

File:	/mnt/c/Users/Asus/Documents/ACN/ns-allinone-3.32/ns-3.32/lab7_3_p2p.tr
Lines on file:	6
Total enqueued packets:	2
Total sent packets:	2
Total received packets:	2
Total dropped packets:	0
Total simulation time:	2.01973 seconds
Time of analysis:	0s