Task 1:-

Modifications:-

In order to obtain point to point link with data rate of 20Mbps & delay 2ms following change was made .

```
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("20Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

Ip address assignment was done using the below code .

```
Ipv4AddressHelper address;
address.SetBase ("190.120.2.0", "255.255.255.0");
```

Port address of echo server was made 63 using the below code.

```
UdpEchoServerHelper echoServer (63);
```

Also, the following change was made to get the ip address of the server.

```
UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 63);
```

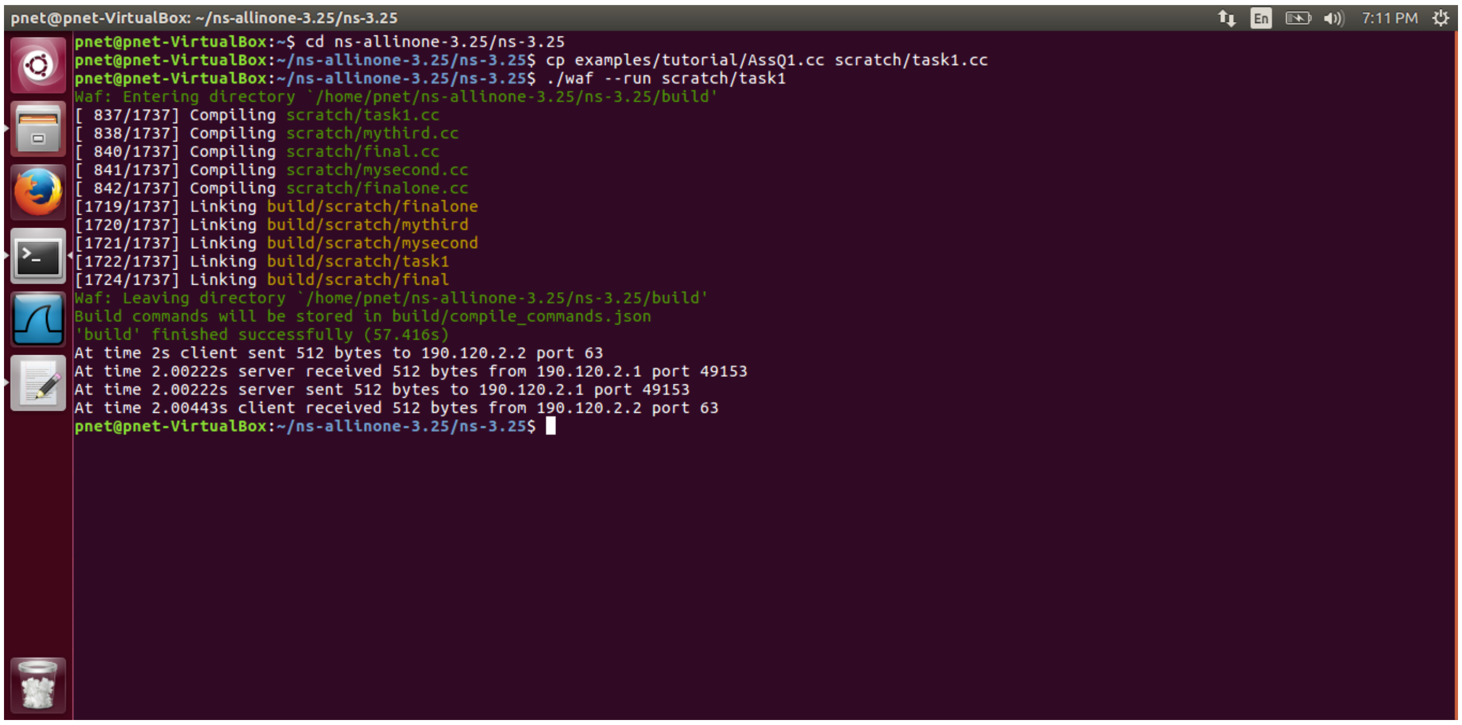
In order to create a packet size of 512 bytes, following code change was made.

```
echoClient.SetAttribute ("PacketSize", UintegerValue (512));
```

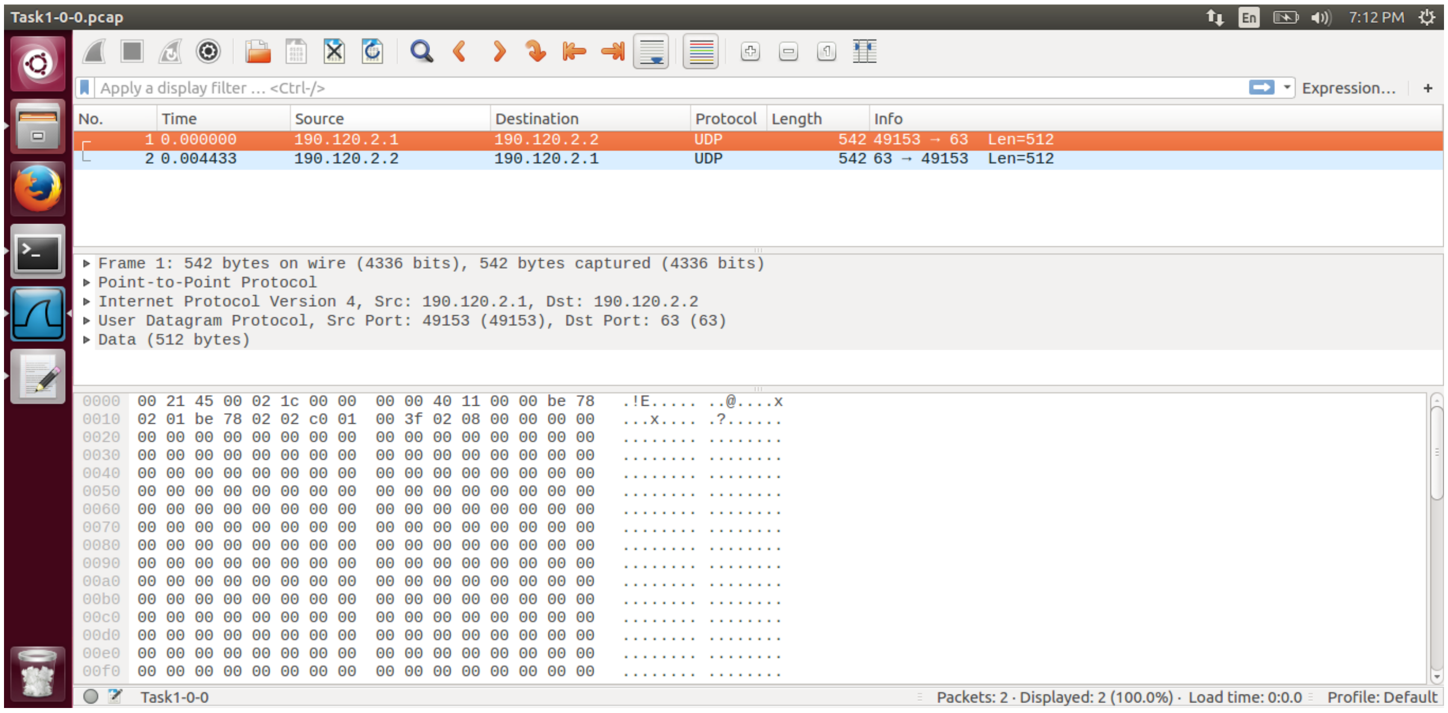
Also, in order to generate pcap file, we have added the following code.

```
pointToPoint.EnablePcapAll("Task1");
```

Result & Learning:-



```
pnet@pnet-VirtualBox:~$ cd ns-allinone-3.25/ns-3.25
pnet@pnet-VirtualBox:~/ns-allinone-3.25/ns-3.25$ cp examples/tutorial/AssQ1.cc scratch/task1.cc
pnet@pnet-VirtualBox:~/ns-allinone-3.25/ns-3.25$ ./waf --run scratch/task1
Waf: Entering directory `/home/pnet/ns-allinone-3.25/ns-3.25/build'
[ 837/1737] Compiling scratch/task1.cc
[ 838/1737] Compiling scratch/mythird.cc
[ 840/1737] Compiling scratch/final.cc
[ 841/1737] Compiling scratch/mysecond.cc
[ 842/1737] Compiling scratch/finalone.cc
[1719/1737] Linking build/scratch/finalone
[1720/1737] Linking build/scratch/mythird
[1721/1737] Linking build/scratch/mysecond
[1722/1737] Linking build/scratch/task1
[1724/1737] Linking build/scratch/final
Waf: Leaving directory `/home/pnet/ns-allinone-3.25/ns-3.25/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (57.416s)
At time 2s client sent 512 bytes to 190.120.2.2 port 63
At time 2.00222s server received 512 bytes from 190.120.2.1 port 49153
At time 2.00222s server sent 512 bytes to 190.120.2.1 port 49153
At time 2.00443s client received 512 bytes from 190.120.2.2 port 63
pnet@pnet-VirtualBox:~/ns-allinone-3.25/ns-3.25$
```



```
Task1-0-0.pcap                                                              7:12 PM

No.   Time        Source        Destination   Protocol Length Info
    1 0.000000    190.120.2.1   190.120.2.2   UDP         542 49153 → 63  Len=512
    2 0.004433    190.120.2.2   190.120.2.1   UDP         542 63 → 49153  Len=512

▸ Frame 1: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits)
▸ Point-to-Point Protocol
▸ Internet Protocol Version 4, Src: 190.120.2.1, Dst: 190.120.2.2
▸ User Datagram Protocol, Src Port: 49153 (49153), Dst Port: 63 (63)
▸ Data (512 bytes)

0000  00 21 45 00 02 1c 00 00  00 00 40 11 00 00 be 78   .!E..... ..@....x
0010  02 01 be 78 02 02 c0 01  00 3f 02 08 00 00 00 00   ...x.... .?......
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   ........ ........

Task1-0-0      Packets: 2 · Displayed: 2 (100.0%) · Load time: 0:0.0      Profile: Default
```

1) From the above screen shots, we can see that at time 2 seconds client sent 512 bytes of data to the server port 63. After 2.22 ms server received the data sent by the client and echoes back the same packet. The client receives back the packet again after another 2.22 ms.
2) We can also see from the wire shark pcap files of the client and server that the data in both the packets is exactly the same.
3) The data is sent using user datagram protocol.
4) The total delay is about 2.22 ms. This delay will reduce if we shorten the packet size. We have tried sending the packet with size 1024 bytes and the delay has increased.
5) Since there was only a point to point connection no ARP request was broadcasted.
6) Through this task we learnt how to make UDP echo application.

Task 2:-

Modifications:-

We created point to point nodes using the code.

```
NodeContainer p2pNodes;
p2pNodes.Create (2);
```

For creating the csma nodes we used the following code.

```
NodeContainer csmaNodes;
csmaNodes.Add (p2pNodes.Get (0));
csmaNodes.Create (2);

NodeContainer csmaNodes1;
csmaNodes1.Add (p2pNodes.Get (1));
csmaNodes1.Create (2);
```

In the above code two different objects were created because we have to give different base ip address in the future. Also, we have made the point to point nodes created earlier as csma.

Point to point device & channel attributes are set through the following code.

```
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("20Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

Installed Network card in the point to point devices using

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

Csma channel attributes are set using

```
CsmaHelper csma;
csma.SetChannelAttribute ("DataRate", StringValue ("200Mbps"));
csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (10000)));
```

Installed Network card in the Csma devices using

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

NetDeviceContainer csmaDevices1;
csmaDevices1 = csma.Install (csmaNodes1);
```

Note: -** **Now we have two network cards in the point to point devices. Also, we have installed separately the network cards using two objects. This is because of the same reason of different base IP address.

After installing the protocols in the devices, we used the below code for assigning the base IP address.

```
Ipv4AddressHelper address;
address.SetBase ("192.120.2.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("192.120.1.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign (csmaDevices);

address.SetBase ("192.120.3.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces1;
csmaInterfaces1 = address.Assign (csmaDevices1);
```

<u>Note: -</u> Point to point devices have been given the base ip address as 192.120.2.0. One set of csma nodes (csmaNodes object) has been given the base IP address as 192.120.1.0 while the other(csmaNodes1 object) has been given 192.120.3.0 as the base IP address.

Node 1 has been made server along with the assignment of port using

```
UdpEchoServerHelper echoServer (21);
ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (1));
```

Node 5 has requested the address of server using

```
UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (1), 21);
```

For sending 2 packets at time 4 sec and 7 sec, we have made the following changes

```
echoClient.SetAttribute ("MaxPackets", UintegerValue (2));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (3.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (csmaNodes1.Get (2));
clientApps.Start (Seconds (4.0));
clientApps.Stop (Seconds (10.0));
```

<u>Note:-</u> We have also made the Node 5 as client using the above code.

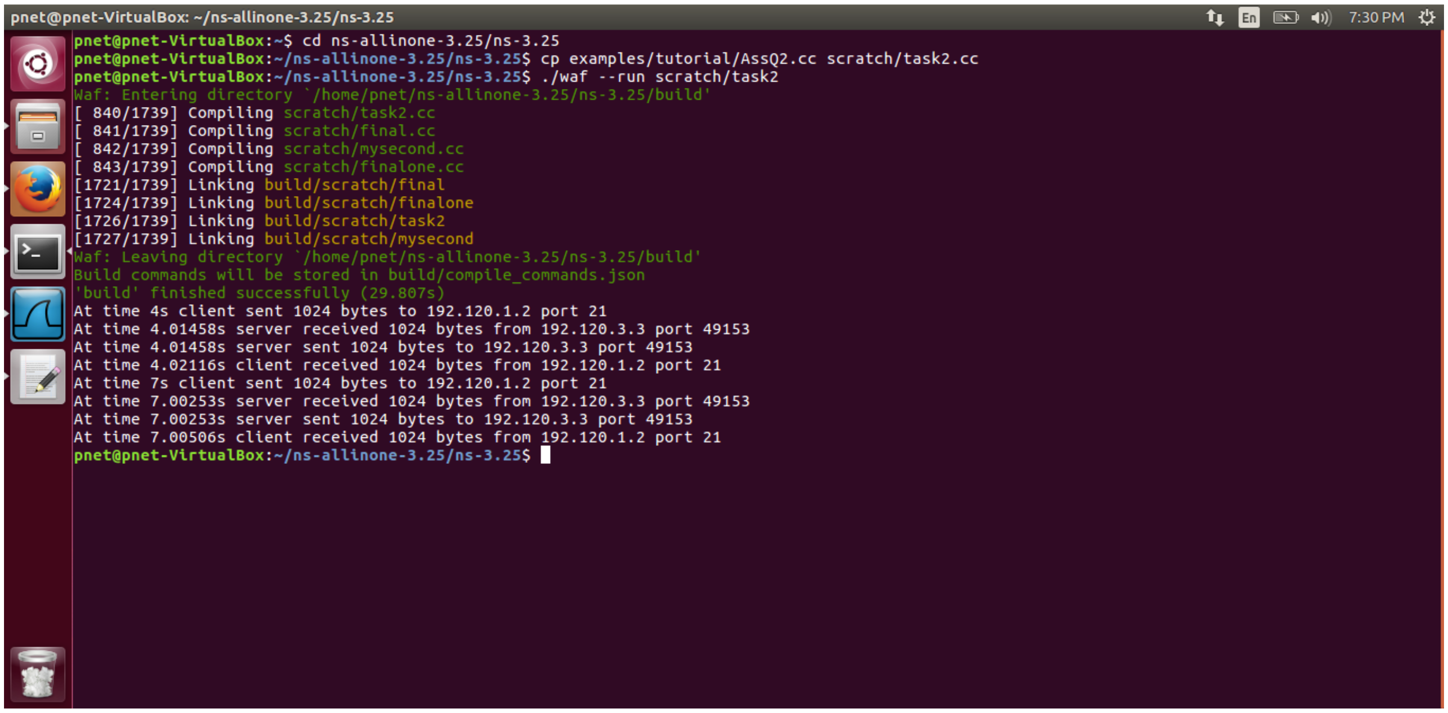
For enabling the PCAP files we used the below code:-

```
pointToPoint.EnablePcapAll ("Task2-P2P");
csma.EnablePcap ("Task2-Server", csmaDevices.Get (1), true);
csma.EnablePcap ("Task2-Client", csmaDevices1.Get (2), true);
csma.EnablePcap ("Task2-nonprom", csmaDevices1.Get (1), false);
```

<u>Note:-</u> The false parameter was used for a non-promiscuous mode packet trace for node 4.

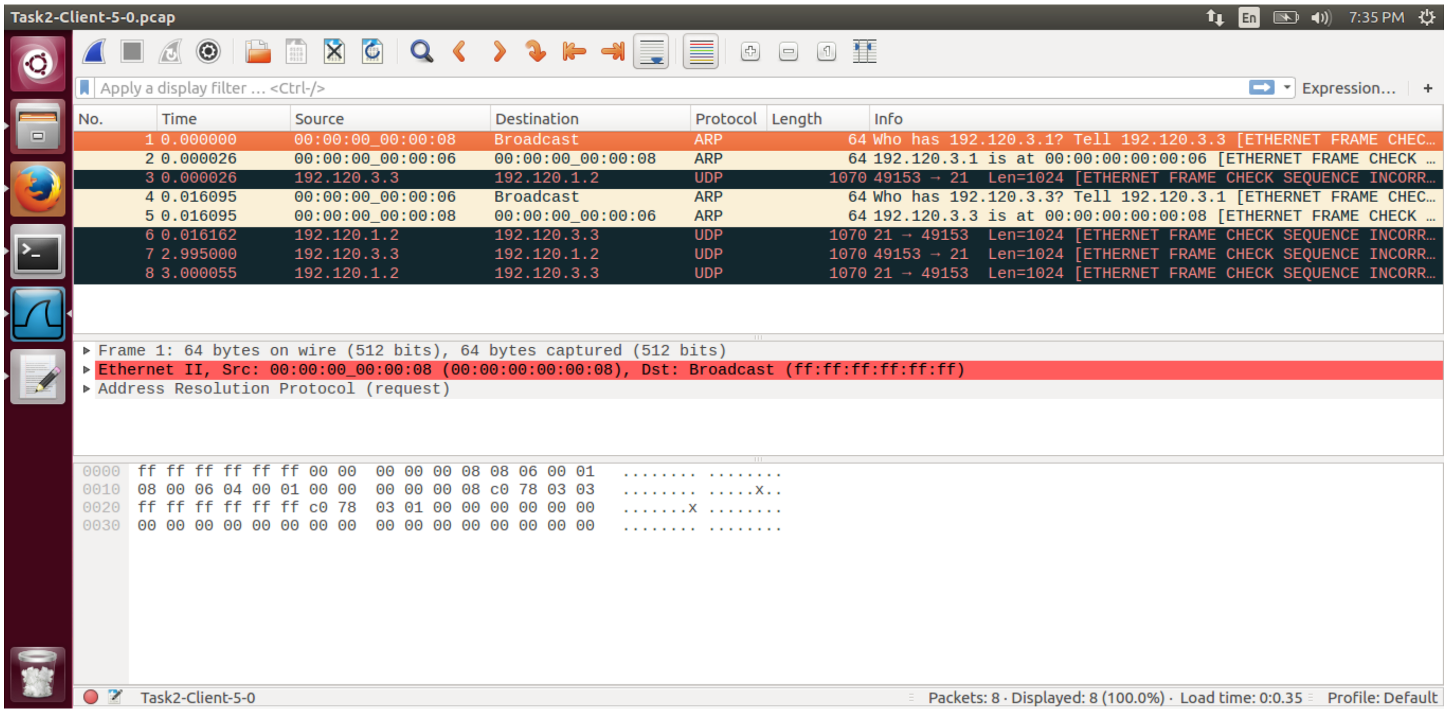
Result & Learning:-

Screenshot of the output



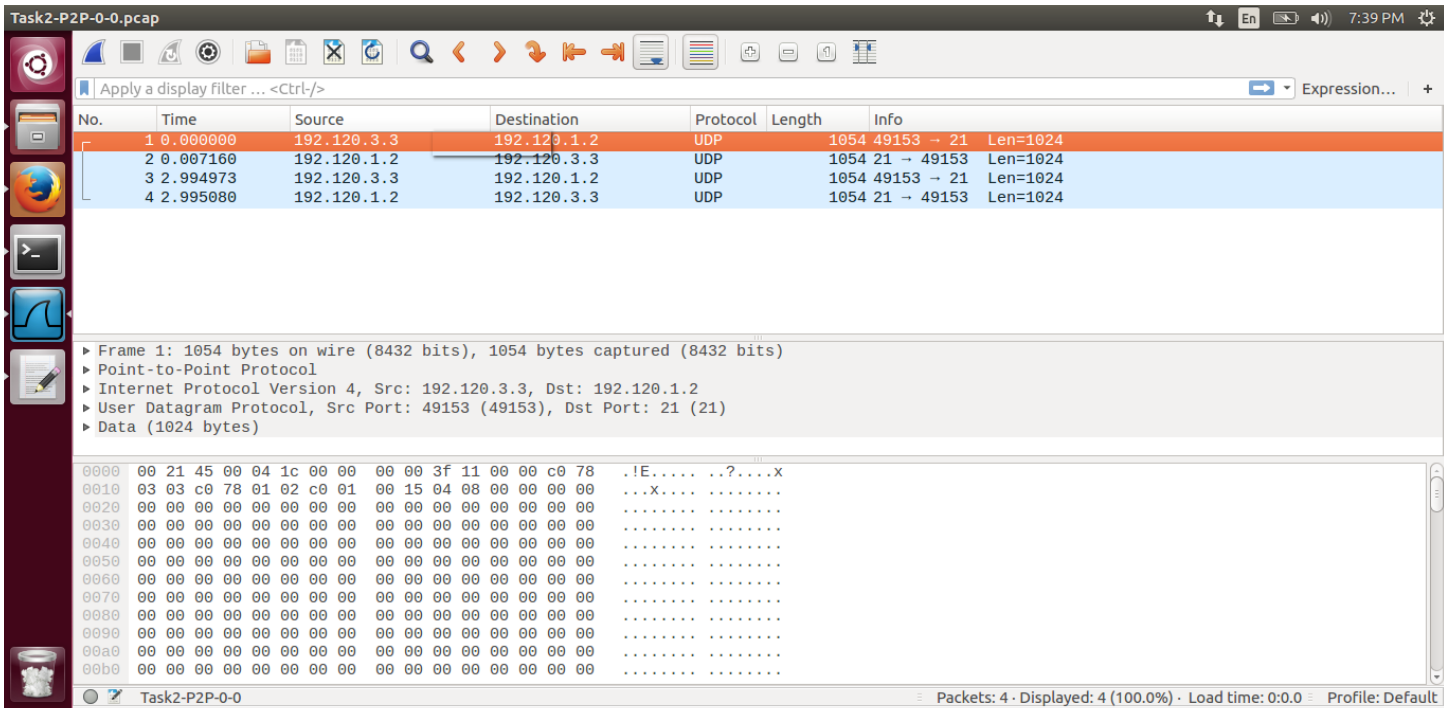
```
pnet@pnet-VirtualBox: ~/ns-allinone-3.25/ns-3.25
pnet@pnet-VirtualBox:~$ cd ns-allinone-3.25/ns-3.25
pnet@pnet-VirtualBox:~/ns-allinone-3.25/ns-3.25$ cp examples/tutorial/AssQ2.cc scratch/task2.cc
pnet@pnet-VirtualBox:~/ns-allinone-3.25/ns-3.25$ ./waf --run scratch/task2
Waf: Entering directory `/home/pnet/ns-allinone-3.25/ns-3.25/build'
[ 840/1739] Compiling scratch/task2.cc
[ 841/1739] Compiling scratch/final.cc
[ 842/1739] Compiling scratch/mysecond.cc
[ 843/1739] Compiling scratch/finalone.cc
[1721/1739] Linking build/scratch/final
[1724/1739] Linking build/scratch/finalone
[1726/1739] Linking build/scratch/task2
[1727/1739] Linking build/scratch/mysecond
Waf: Leaving directory `/home/pnet/ns-allinone-3.25/ns-3.25/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (29.807s)
At time 4s client sent 1024 bytes to 192.120.1.2 port 21
At time 4.01458s server received 1024 bytes from 192.120.3.3 port 49153
At time 4.01458s server sent 1024 bytes to 192.120.3.3 port 49153
At time 4.02116s client received 1024 bytes from 192.120.1.2 port 21
At time 7s client sent 1024 bytes to 192.120.1.2 port 21
At time 7.00253s server received 1024 bytes from 192.120.3.3 port 49153
At time 7.00253s server sent 1024 bytes to 192.120.3.3 port 49153
At time 7.00506s client received 1024 bytes from 192.120.1.2 port 21
pnet@pnet-VirtualBox:~/ns-allinone-3.25/ns-3.25$
```

Screenshot of the Client wire shark pcap



```
Task2-Client-5-0.pcap

No.  Time       Source             Destination        Protocol Length Info
   1 0.000000   00:00:00_00:00:08  Broadcast          ARP        64 Who has 192.120.3.1? Tell 192.120.3.3 [ETHERNET FRAME CHEC…
   2 0.000026   00:00:00_00:00:06  00:00:00_00:00:08  ARP        64 192.120.3.1 is at 00:00:00:00:00:06 [ETHERNET FRAME CHECK …
   3 0.000026   192.120.3.3        192.120.1.2        UDP      1070 49153 → 21  Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORR…
   4 0.016095   00:00:00_00:00:06  Broadcast          ARP        64 Who has 192.120.3.3? Tell 192.120.3.1 [ETHERNET FRAME CHEC…
   5 0.016095   00:00:00_00:00:08  00:00:00_00:00:06  ARP        64 192.120.3.3 is at 00:00:00:00:00:08 [ETHERNET FRAME CHECK …
   6 0.016162   192.120.1.2        192.120.3.3        UDP      1070 21 → 49153  Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORR…
   7 2.995000   192.120.3.3        192.120.1.2        UDP      1070 49153 → 21  Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORR…
   8 3.000055   192.120.1.2        192.120.3.3        UDP      1070 21 → 49153  Len=1024 [ETHERNET FRAME CHECK SEQUENCE INCORR…

▸ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
▸ Ethernet II, Src: 00:00:00_00:00:08 (00:00:00:00:00:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)

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 c0 78 03 03   ........ .....x..
0020  ff ff ff ff ff ff c0 78  03 01 00 00 00 00 00 00   .......x ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........

Task2-Client-5-0        Packets: 8 · Displayed: 8 (100.0%) · Load time: 0:0.35   Profile: Default
```

Screenshot of the point to point wire shark pcap



Task2-P2P-0-0.pcap

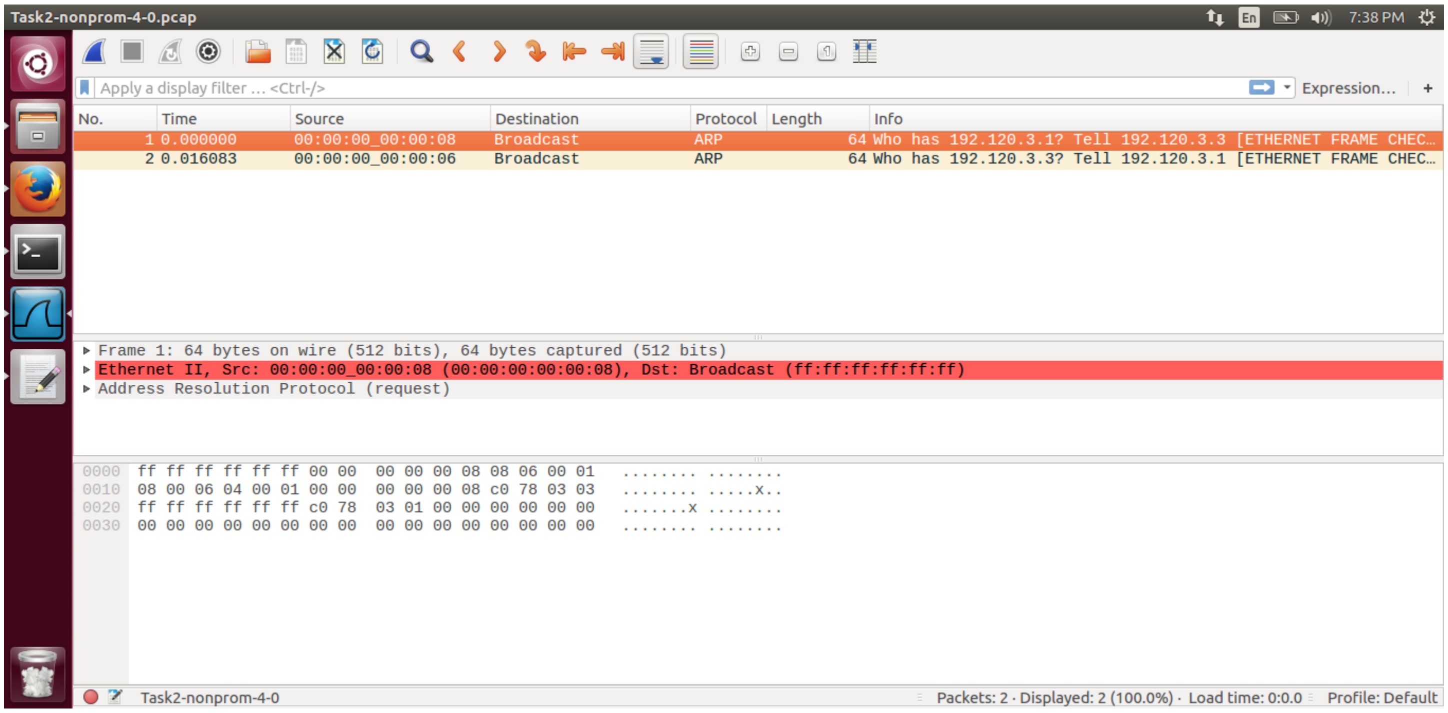
```
No.  Time        Source         Destination    Protocol Length Info
   1 0.000000    192.120.3.3    192.120.1.2    UDP        1054 49153 → 21  Len=1024
   2 0.007160    192.120.1.2    192.120.3.3    UDP        1054 21 → 49153  Len=1024
   3 2.994973    192.120.3.3    192.120.1.2    UDP        1054 49153 → 21  Len=1024
   4 2.995080    192.120.1.2    192.120.3.3    UDP        1054 21 → 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: 192.120.3.3, Dst: 192.120.1.2
▸ User Datagram Protocol, Src Port: 49153 (49153), Dst Port: 21 (21)
▸ Data (1024 bytes)
```

```
0000  00 21 45 00 04 1c 00 00  00 00 3f 11 00 00 c0 78   .!E..... ..?....x
0010  03 03 c0 78 01 02 c0 01  00 15 04 08 00 00 00 00   ...x.... ........
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   ........ ........
```

Task2-P2P-0-0 Packets: 4 · Displayed: 4 (100.0%) · Load time: 0:0.0 Profile: Default

Screenshot of the node 4 (non – promiscuous mode)



Task2-nonprom-4-0.pcap

```
No.  Time        Source              Destination  Protocol Length Info
   1 0.000000    00:00:00_00:00:08   Broadcast    ARP         64 Who has 192.120.3.1? Tell 192.120.3.3 [ETHERNET FRAME CHEC…
   2 0.016083    00:00:00_00:00:06   Broadcast    ARP         64 Who has 192.120.3.3? Tell 192.120.3.1 [ETHERNET FRAME CHEC…
```

```
▸ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
▸ Ethernet II, Src: 00:00:00_00:00:08 (00:00:00:00:00:08), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸ Address Resolution Protocol (request)
```

```
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 c0 78 03 03   ........ .....x..
0020  ff ff ff ff ff ff c0 78  03 01 00 00 00 00 00 00   .......x ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Task2-nonprom-4-0 Packets: 2 · Displayed: 2 (100.0%) · Load time: 0:0.0 Profile: Default

1) From the output screen shot, we can easily see that 2 packets are being sent. One at 4 second while the other at 7 second.
2) From the client wire shark pcap file we can see that in order to know the MAC address of the destination ARP request is broadcasted. This ARP request is broadcasted twice. One for the client to know the address of the server and the other for the server to know the address of the client. We can see ARP 4 times in the pcap file as the 2 and 4 are the response by the destination.
3) We can also see that for the second packet to the same destination no ARP request was broadcasted. This is because now the client and server know the MAC address of each other.
4) **Important: -** For the first packet we can see that there is a delay of about 14.5 ms for the data to reach from client to server. However, for the second packet there is a delay of only about 2.53 ms. This is because we do not have to send ARP request for the second time.
5) Since, this is echoing so we can see from the wire shark file that the same data is transmitted and received back.
6) From the wire shark file of node 4, we can see that it only received the ARP broadcast packets. It did not receive (did not accept) the data packets flowing between server and client. This is because it is working in non-promiscuous mode.
7) Through this task, we learned on how to make UDP echo application for client and server present in different networks.

Q1)

Ans :- Node 2 and Node 3 are part of two networks. They are connected to both CSMA link as well as Point to Point link. They cannot connect to both the links using the same network card. That's why they need two network adapters.

Q2)

Ans:- There are 3 networks in Figure 1. Since the subnet mask is /24 so the first 3 octets will decide whether they are in the same network or different. Here we have base IP addresses as 192.120.1.0, 192.120.2.0 & 192.120.3.0. Since the first 3 octets of all three base IP addresses are different so there are 3 networks.

Q3)

Ans:- Promiscuous mode is the mode which allows the interface to receive all the packet whether they are addressed to it or not. The node in the promiscuous mode can read the packets even if they are not addressed to it. However, in the non-promiscuous mode the node will discard the packets not addressed to it.

Group 4

Pulkit Hanswal

Roshan Baby Reji