# Experiment 4 - Building Topologies in NS3

Muhammad Atif Farooq L144392 EE

26th September 2019

## 1 Introduction

Carrier-Sense multiples access with collision avoidance or **CSMA** is a multiple accesss channel method, where a multitude of nodes are connected to a common transmission medium. The nodes then listen in on the traffic of the channel, trasmitting only when the the channel is idle, in addition when they do transmit there packet load in full. It is a protocol that appears in the Data Link Layer of the OSI model.

## 2 Objective

Build a network by making use of the `CsmaHelper` class in NS3.

## 3 Procedure

For the lab task we were required to implement the following topology using the `CsmaHelper` class in NS3. We discuss the procedure by remarking first on the topology in question and then discuss the stratedy we employed to realize this topology in NS3.

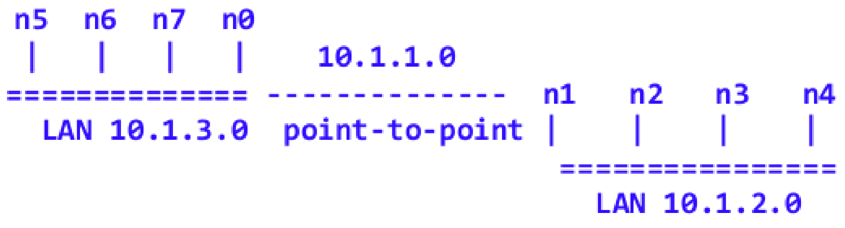


Figure 1: topologyLabQuestion.png

## 3.1 The Topology

As illustrated in the above figure we were required to implement two csma channels i.e. bus topologies with there terminal nodes being connected by a point to point link. nodes six and nodes two were meant to house the client and the server respectively, with the server being configured to listen in on port 93.

## 3.2 The Stratedgy Employed

We started by first creating a node container to house the terminal nodes for the point to point link we then created two other node container containing three nodes each, prepending one and appending the second to the terminal nodes. The remaining task was the conventional NS3 procedure of defining the channel attributes for the csma and point to point links, installing net devices and protocols stack, assigning IPs, and installing the the server and client at the appropriate nodes. We then enabled pcap tracing using the `EnablePcap()` method of the `CsmaHelper` class to determine which of the nodes in the topology were involved in the routing process.

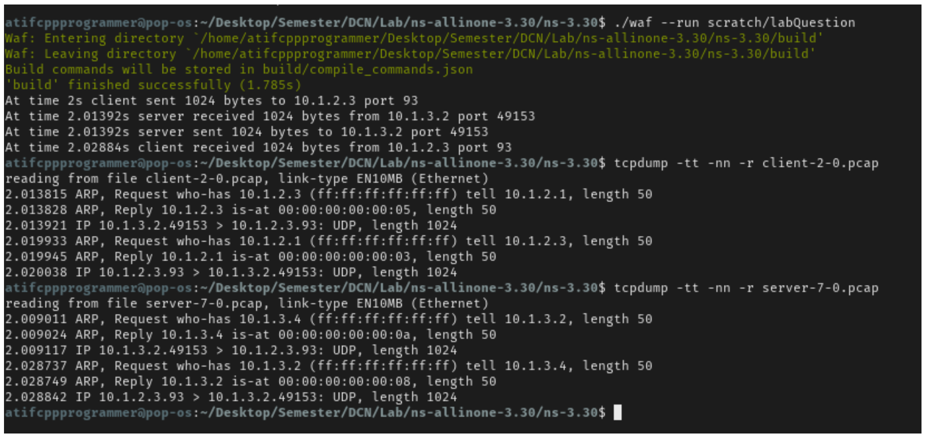
The required code and program output including the pcap traces are presented below.

```
#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"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("lab_question_script");

int
main (int argc, char *argv[])
{
  // For verbose output and number of csma nodes.
  bool verbose = true;
  uint32_t nCsma = 3;

  CommandLine cmd;
  cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
  cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);

  cmd.Parse (argc,argv);
```

```
if (verbose)
  {
    LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
  }

nCsma = nCsma == 0 ? 1 : nCsma;

// Creating Node Container to House Nodes-0,1.
NodeContainer p2pNodes;
p2pNodes.Create(2);

// Creating Node Container to House Nodes-1,2,3,4.
NodeContainer csmaNodesOne;
csmaNodesOne.Add(p2pNodes.Get(1));
csmaNodesOne.Create(nCsma);

// Creating Node Container to House Nodes-5,6,7,0.
NodeContainer csmaNodesTwo;
csmaNodesTwo.Create(nCsma);
csmaNodesTwo.Add(p2pNodes.Get(0));

// Specifying Point to Point Link Attributes.
PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

// Installing Net Devices.
NetDeviceContainer p2pDevices;
p2pDevices = pointToPoint.Install (p2pNodes);

// Specifying csma channel Attributes.
CsmaHelper csmaOne;
csmaOne.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));
csmaOne.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

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

// Installing NetDevices.
NetDeviceContainer csmaDevicesOne,csmaDevicesTwo;
csmaDevicesOne = csmaOne.Install(csmaNodesOne);
csmaDevicesTwo = csmaTwo.Install(csmaNodesTwo);
```

```
// Installing Protocol Stack on p2pNodes and csma nodes.
InternetStackHelper stack;

stack.Install(csmaNodesOne);
stack.Install(csmaNodesTwo);

// Specifying IP Assignments for point to point nodes.
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

// Specifying IP Assignments for csma nodes.
address.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfacesOne;
csmaInterfacesOne = address.Assign (csmaDevicesOne);

address.SetBase ("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfacesTwo;
csmaInterfacesTwo = address.Assign (csmaDevicesTwo);

// Creating Server on Node-3 with port 93 and Installing Server Apps.
UdpEchoServerHelper echoServer (93);

ApplicationContainer serverApps = echoServer.Install(csmaNodesOne.Get(nCsma-1));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

// Creating Client on Node-2
UdpEchoClientHelper echoClient (csmaInterfacesOne.GetAddress (nCsma-1), 93);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (csmaNodesTwo.Get (1));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

// To Enable Routing.
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// For tcp-dump traces of client and server.
csmaOne.EnablePcap ("client", csmaDevicesOne.Get (1), true);
csmaTwo.EnablePcap ("server", csmaDevicesTwo.Get (2), true);
```

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



```
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ ./waf --run scratch/labQuestion
Waf: Entering directory `/home/atifcppprogrammer/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30/build'
Waf: Leaving directory `/home/atifcppprogrammer/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.785s)
At time 2s client sent 1024 bytes to 10.1.2.3 port 93
At time 2.01392s server received 1024 bytes from 10.1.3.2 port 49153
At time 2.01392s server sent 1024 bytes to 10.1.3.2 port 49153
At time 2.02884s client received 1024 bytes from 10.1.2.3 port 93
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ tcpdump -tt -nn -r client-2-0.pcap
reading from file client-2-0.pcap, link-type EN10MB (Ethernet)
2.013815 ARP, Request who-has 10.1.2.3 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
2.013828 ARP, Reply 10.1.2.3 is-at 00:00:00:00:00:05, length 50
2.013921 IP 10.1.3.2.49153 > 10.1.2.3.93: UDP, length 1024
2.019933 ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.3, length 50
2.019945 ARP, Reply 10.1.2.1 is-at 00:00:00:00:00:03, length 50
2.020038 IP 10.1.2.3.93 > 10.1.3.2.49153: UDP, length 1024
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ tcpdump -tt -nn -r server-7-0.pcap
reading from file server-7-0.pcap, link-type EN10MB (Ethernet)
2.009011 ARP, Request who-has 10.1.3.4 (ff:ff:ff:ff:ff:ff) tell 10.1.3.2, length 50
2.009024 ARP, Reply 10.1.3.4 is-at 00:00:00:00:00:0a, length 50
2.009117 IP 10.1.3.2.49153 > 10.1.2.3.93: UDP, length 1024
2.028737 ARP, Request who-has 10.1.3.2 (ff:ff:ff:ff:ff:ff) tell 10.1.3.4, length 50
2.028749 ARP, Reply 10.1.3.2 is-at 00:00:00:00:00:08, length 50
2.028842 IP 10.1.2.3.93 > 10.1.3.2.49153: UDP, length 1024
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$
```

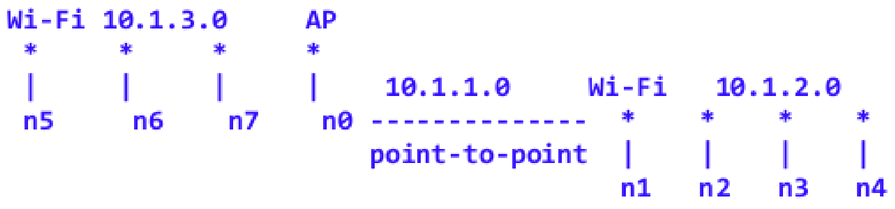
Figure 2: labQuestion.cc

## 4 Conclusions

The `CsmaHelper` class can used to create complex topologies that incorporate bus topologies in NS3.

## 5 Post-Lab Question

For the post lab we werer required to implement the following topology.



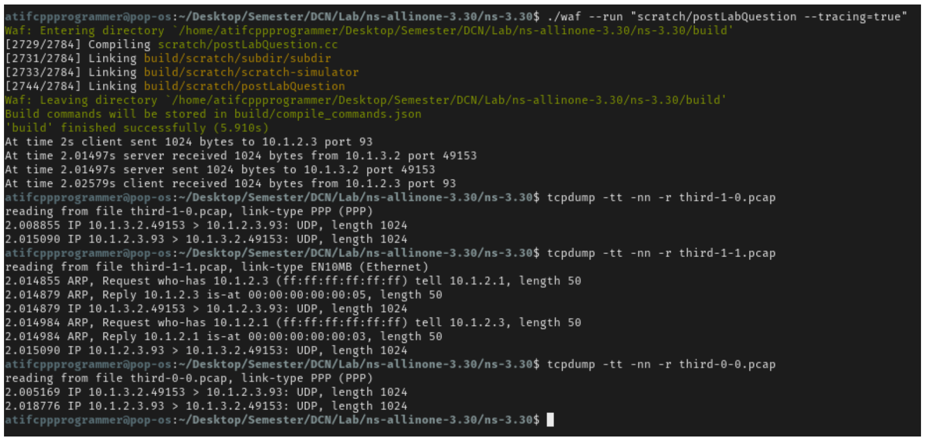
```
 Wi-Fi 10.1.3.0      AP
  *      *     *      *
  |      |     |      |    10.1.1.0      Wi-Fi    10.1.2.0
  n5     n6    n7     n0 --------------    *     *     *     *
                         point-to-point    |     |     |     |
                                           n1    n2    n3    n4
```

Figure 3: topologyPostLabQuestion.png

The code required to implement the topology in question is presented in `examples/tutorial/third.cc` therefore we only provide the code required to meet stipulations set in the statement of post lab question.

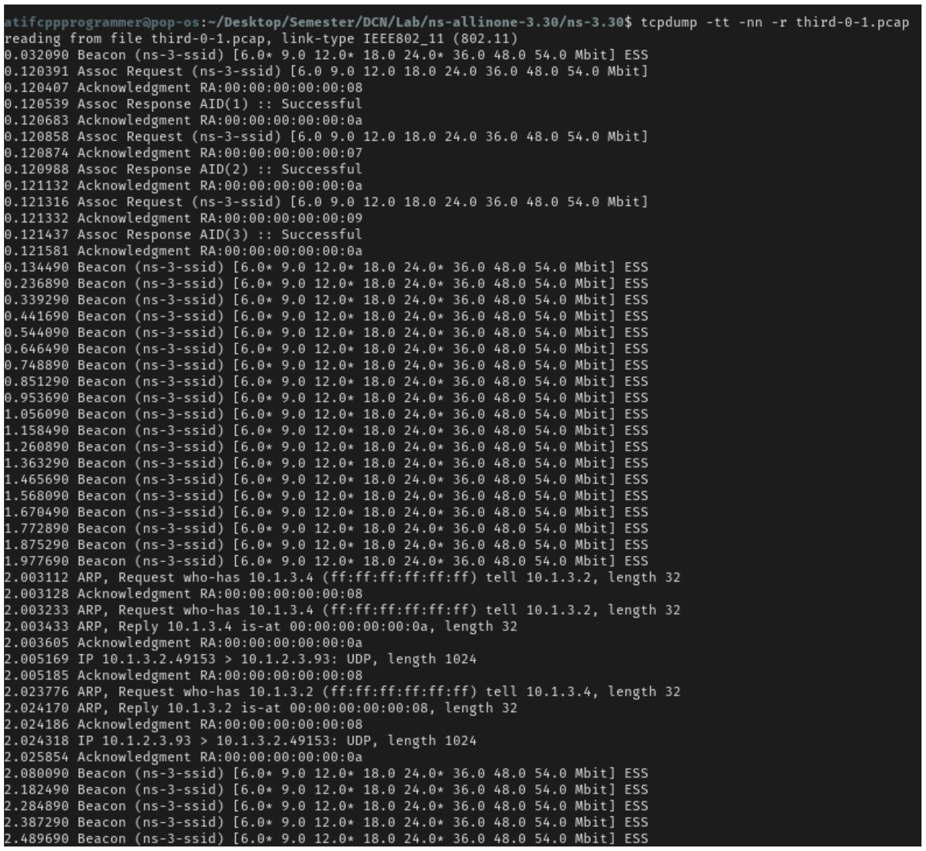
```
UdpEchoServerHelper echoServer (93);

ApplicationContainer serverApps = echoServer.Install (csmaNodes.Get (2));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (csmaInterfaces.GetAddress (2), 93);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps =
  echoClient.Install (wifiStaNodes.Get (1));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));
```

What follows are the script output in addition to the pcap traces of the point to point nodes of the topology in question.



```
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ ./waf --run "scratch/postLabQuestion --tracing=true"
Waf: Entering directory `/home/atifcppprogrammer/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30/build'
[2729/2784] Compiling scratch/postLabQuestion.cc
[2731/2784] Linking build/scratch/subdir/subdir
[2733/2784] Linking build/scratch/scratch-simulator
[2744/2784] Linking build/scratch/postLabQuestion
Waf: Leaving directory `/home/atifcppprogrammer/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.910s)
At time 2s client sent 1024 bytes to 10.1.2.3 port 93
At time 2.01497s server received 1024 bytes from 10.1.3.2 port 49153
At time 2.01497s server sent 1024 bytes to 10.1.3.2 port 49153
At time 2.02579s client received 1024 bytes from 10.1.2.3 port 93
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ tcpdump -tt -nn -r third-1-0.pcap
reading from file third-1-0.pcap, link-type PPP (PPP)
2.008855 IP 10.1.3.2.49153 > 10.1.2.3.93: UDP, length 1024
2.015090 IP 10.1.2.3.93 > 10.1.3.2.49153: UDP, length 1024
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ tcpdump -tt -nn -r third-1-1.pcap
reading from file third-1-1.pcap, link-type EN10MB (Ethernet)
2.014855 ARP, Request who-has 10.1.2.3 (ff:ff:ff:ff:ff:ff) tell 10.1.2.1, length 50
2.014879 ARP, Reply 10.1.2.3 is-at 00:00:00:00:00:05, length 50
2.014879 IP 10.1.3.2.49153 > 10.1.2.3.93: UDP, length 1024
2.014984 ARP, Request who-has 10.1.2.1 (ff:ff:ff:ff:ff:ff) tell 10.1.2.3, length 50
2.014984 ARP, Reply 10.1.2.1 is-at 00:00:00:00:00:03, length 50
2.015090 IP 10.1.2.3.93 > 10.1.3.2.49153: UDP, length 1024
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ tcpdump -tt -nn -r third-0-0.pcap
reading from file third-0-0.pcap, link-type PPP (PPP)
2.005169 IP 10.1.3.2.49153 > 10.1.2.3.93: UDP, length 1024
2.018776 IP 10.1.2.3.93 > 10.1.3.2.49153: UDP, length 1024
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$
```

Figure 4: postLabQuestionOne.png



```
atifcppprogrammer@pop-os:~/Desktop/Semester/DCN/Lab/ns-allinone-3.30/ns-3.30$ tcpdump -tt -nn -r third-0-1.pcap
reading from file third-0-1.pcap, link-type IEEE802_11 (802.11)
0.032090 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.120391 Assoc Request (ns-3-ssid) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
0.120407 Acknowledgment RA:00:00:00:00:00:08
0.120539 Assoc Response AID(1) :: Successful
0.120683 Acknowledgment RA:00:00:00:00:00:0a
0.120858 Assoc Request (ns-3-ssid) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
0.120874 Acknowledgment RA:00:00:00:00:00:07
0.120988 Assoc Response AID(2) :: Successful
0.121132 Acknowledgment RA:00:00:00:00:00:0a
0.121316 Assoc Request (ns-3-ssid) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
0.121332 Acknowledgment RA:00:00:00:00:00:09
0.121437 Assoc Response AID(3) :: Successful
0.121581 Acknowledgment RA:00:00:00:00:00:0a
0.134490 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.236890 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.339290 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.441690 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.544090 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.646490 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.748890 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.851290 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
0.953690 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.056090 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.158490 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.260890 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.363290 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.465690 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.568090 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.670490 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.772890 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.875290 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
1.977690 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
2.003112 ARP, Request who-has 10.1.3.4 (ff:ff:ff:ff:ff:ff) tell 10.1.3.2, length 32
2.003128 Acknowledgment RA:00:00:00:00:00:08
2.003233 ARP, Request who-has 10.1.3.4 (ff:ff:ff:ff:ff:ff) tell 10.1.3.2, length 32
2.003433 ARP, Reply 10.1.3.4 is-at 00:00:00:00:00:0a, length 32
2.003605 Acknowledgment RA:00:00:00:00:00:0a
2.005169 IP 10.1.3.2.49153 > 10.1.2.3.93: UDP, length 1024
2.005185 Acknowledgment RA:00:00:00:00:00:08
2.023776 ARP, Request who-has 10.1.3.2 (ff:ff:ff:ff:ff:ff) tell 10.1.3.4, length 32
2.024170 ARP, Reply 10.1.3.2 is-at 00:00:00:00:00:08, length 32
2.024186 Acknowledgment RA:00:00:00:00:00:08
2.024318 IP 10.1.2.3.93 > 10.1.3.2.49153: UDP, length 1024
2.025854 Acknowledgment RA:00:00:00:00:00:0a
2.080090 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
2.182490 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
2.284890 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
2.387290 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
2.489690 Beacon (ns-3-ssid) [6.0* 9.0 12.0* 18.0 24.0* 36.0 48.0 54.0 Mbit] ESS
```

Figure 5: postLabQuestionOne.png