Assignment 3

Students' Names:

Aya Metwally

Heba Mostafa

Amira Abu Issa

Date:

6/21/2023

Professor: Dr. Wail Mardini

Teaching Assistant: Murat Arda Onsu

The Report: Part D

Part A (Creating the topologies):

- Define nodes:

```
int
main (int argc, char *argv[])
{
  bool verbose = true;
  uint32_t nCsma = 7;
  uint32_t nWifi = 9;
  bool tracing = true;

  CommandLine cmd (__FILE__);
  cmd.AddValue ("nCsma", "Number of \"extra\" CSMA nodes/devices", nCsma);
  cmd.AddValue ("nWifi", "Number of wifi STA devices", nWifi);
  cmd.AddValue ("verbose", "Tell echo applications to log if true", verbose);
  cmd.AddValue ("tracing", "Enable pcap tracing", tracing);

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

- Point-to-point part: Which is two nodes connected via point-to-point topology. Please consider 10Mbps for data rate and 5ms for Delay in point-to-point topology.

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

PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("10Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("5ms"));

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

- Use related ns3 helpers to configure the required topology on the nodes.

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

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

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

- Create Adhoc wifi

```
NodeContainer adhocNodes;
adhocNodes.Add (p2pNodes.Get (0));
adhocNodes.Create (nWifi);

YansWifiChannelHelper channel = YansWifiChannelHelper::Default ();
YansWifiPhyHelper phy;
phy.SetChannel (channel.Create ());

WifiHelper adhocWifi;
adhocWifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode", StringValue ("OfdmRate54Mbps"));

WifiMacHelper adhocMac;
adhocMac.SetType ("ns3::AdhocWifiMac");

NetDeviceContainer adhocDevices ;
adhocDevices = adhocWifi.Install (phy, adhocMac, adhocNodes);
```

- Install protocol stack on the nodes. (Please consider not to define protocol stack on any of the nodes twice)

```
InternetStackHelper stack;
stack.Install (csmaNodes);
stack.Install (adhocNodes);
```

- Install IP on the nodes.
  1. IP range of 20.1.1.0 (with subnet mask 255.255.255.0) for p2p nodes
  2. IP range of 20.1.2.0 (with subnet mask 255.255.255.0) for Adhoc WIFI nodes
  3. IP range of 20.1.3.0 (with subnet mask 255.255.255.0) for CSMA nodes

```
Ipv4AddressHelper address;

address.SetBase ("20.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("20.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer wifiInterfaces;
wifiInterfaces = address.Assign (adhocDevices);

address.SetBase ("20.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign (csmaDevices);
```

- Install mobility on the nodes

```
MobilityHelper mobility;

mobility.SetPositionAllocator ("ns3::GridPositionAllocator",
                               "MinX", DoubleValue (0.0),
                               "MinY", DoubleValue (0.0),
                               "DeltaX", DoubleValue (5.0),
                               "DeltaY", DoubleValue (10.0),
                               "GridWidth", UintegerValue (3),
                               "LayoutType", StringValue ("RowFirst"));

mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel", "Bounds", RectangleValue (Rectangle (-50, 50, -50, 50)));
mobility.Install (adhocNodes);

mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (p2pNodes);
mobility.Install (csmaNodes);
```

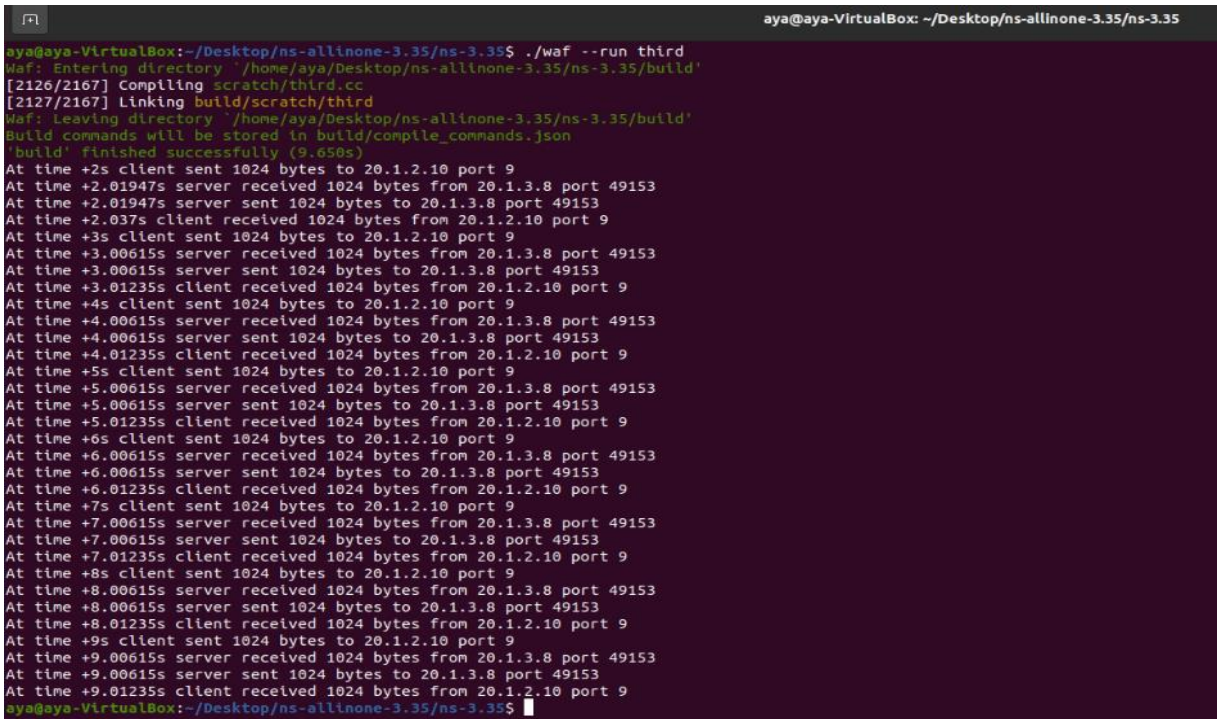
Part B (Creating the application):

- Use UDP Echo Client application and UDP Echo Server application

```
UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (adhocNodes.Get (nWifi));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

UdpEchoClientHelper echoClient (wifiInterfaces.GetAddress (nWifi), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (100));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (csmaNodes.Get (nCsma));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

Simulator::Stop (Seconds (10.0));
```

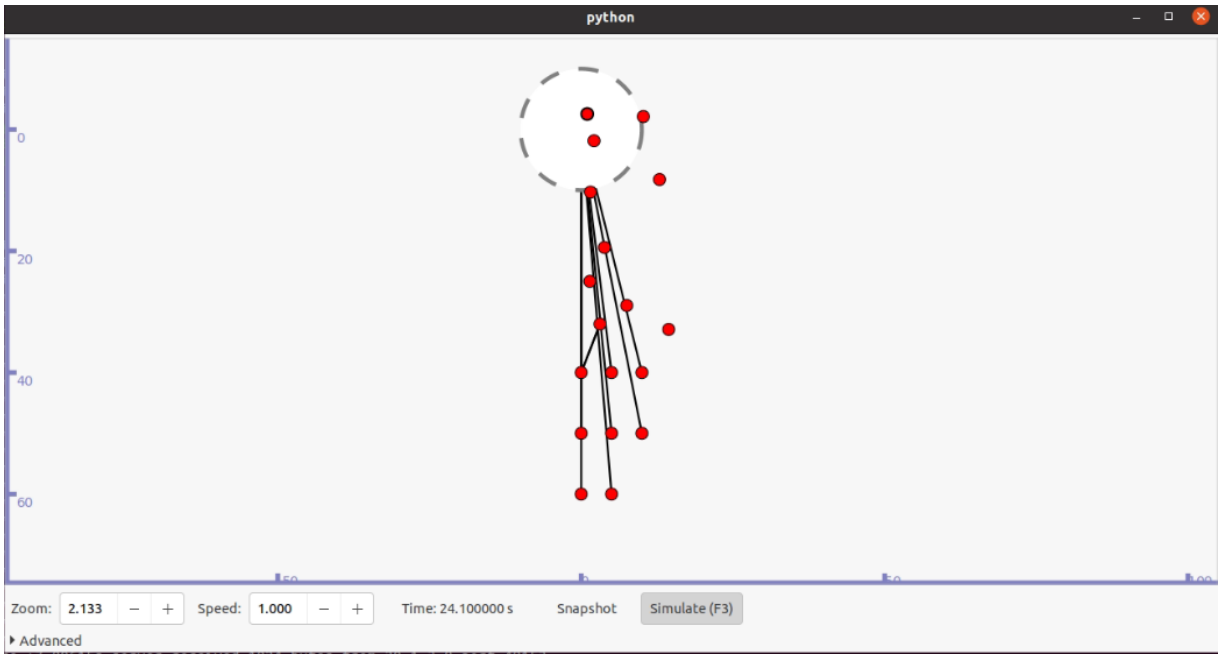
Part C (log of the applications):

- A snapshot of the command line output showing the log of sent and received packets

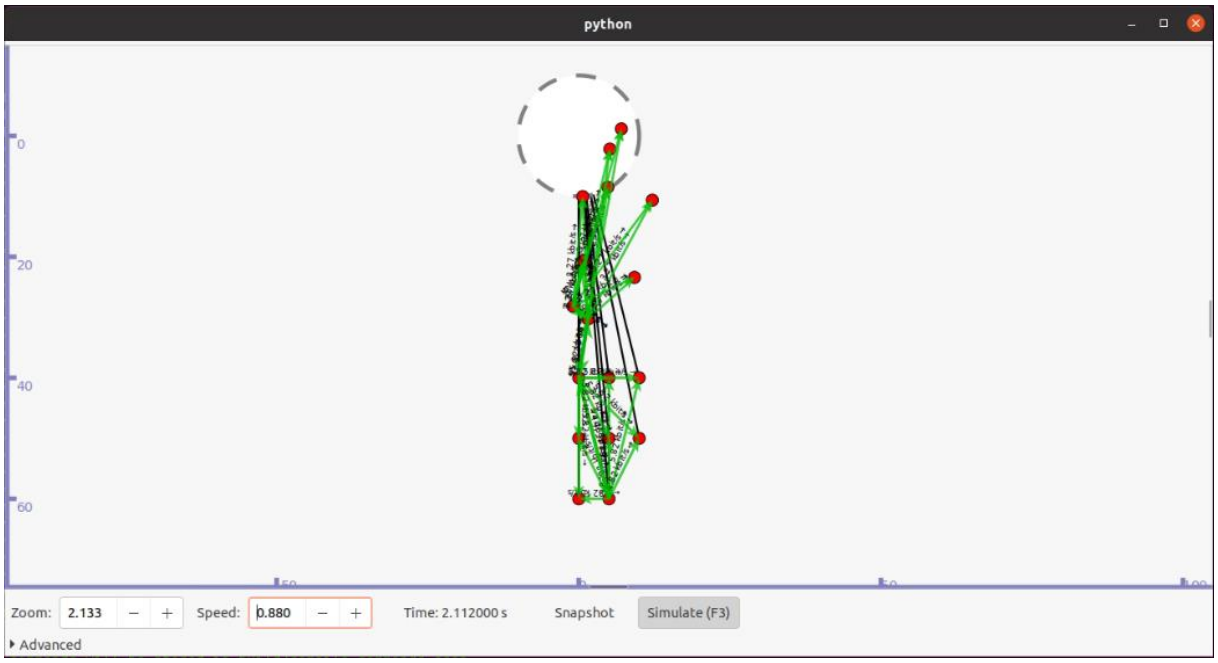


Results

- A snapshot of the visualization using vis



- After simulation



Conclusion

In this report, we have shown the creation of a topology in NS3, which includes three parts: point-to-point, Adhoc WIFI, and CSMA networks. We have used various NS3 helpers to configure the required topology on the nodes and installed the Internet stack and assigned IP addresses to the nodes. We have also installed the UDP Echo Server and Client applications on the appropriate nodes and set the required attributes for the applications.

List of files

The name of the files that attached in Brightspace:

1- Report-Assignment3-NS3.pdf
2- third.cc
3- third -0-0.pcap
4- third -0-1.pcap
5- third-1-0.pcap
6- third -1-1.pcap