Assignment 3

Group number: 12

Students' Names:

Ahmed Ahmed Alwasefy

Ibrahim Mostafa Elshenhapy

Mohamed Salah Gabr

Date:

2023/06/22

Task 3

1-1- Modifications in code

1. Please provide images of modifications in code. Highlight the parts you have changed.

```
#include "ns3/core-module.h"
#include "ns3/point-to-point-module.h
#include "ns3/network-module.h"
#include "ns3/applications-module.h"
#include "ns3/mobility-module.h"
#include "ns3/csma-module.h"
#include "ns3/internet-module.h"
#include "ns3/yans-wifi-helper.h"
#include "ns3/ssid.h"

#include "ns3/wifi-module.h" //
#include "ns3/olsr-helper.h" //
#include "ns3/netanim-module.h" //
```

The first node in point-to-point topology is connected to an Adhoc WIFI network with 10 nodes.
The second node in point-to-point topology is connected to a LAN network with CSMA topology with 8 nodes.

```
int
main (int argc, char *argv[])
{
  bool verbose = true;
  uint32_t nCsma = 7; //8 nodes
  uint32_t nWifi = 9; //9 nodes
  bool tracing = true;
```

This part to Generating log of applications. (Part C) [Placed at the end of the code]

```
  if (tracing)
    {
      phy.SetPcapDataLinkType (WifiPhyHelper::DLT_IEEE802_11_RADIO);
      pointToPoint.EnablePcapAll ("third");
      phy.EnablePcap ("third", adDevices.Get (0));
      csma.EnablePcap ("third", csmaDevices.Get (0), true);
    }
```

Define your nodes.
10Mbps for data rate and 5ms for Delay in point-to-point topology.
Use related ns3 helpers to configure the required topology on the nodes.

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

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

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

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);
```

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

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

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

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

NetDeviceContainer adDevices;
adDevices = adWifi.Install (phy, adMac, wifiadNodes);
```

Install mobility on the nodes.
For point-to-point and CSMA parts, used Constant Position mobility models.
For Adhoc WIFI part, use Random Walk mobility model.
Install protocol stack 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 (wifiadNodes);

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

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

Install IP on the nodes.

- IP range of 20.1.1.0 (with subnet mask 255.255.255.0) for p2p nodes
- IP range of 20.1.2.0 (with subnet mask 255.255.255.0) for Adhoc WIFI nodes
- 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 ("20.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign (csmaDevices);

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

Use UDP Echo Client application and UDP Echo Server application to send and receive packets between client and server.

- The server (receiver) would be installed on the last node of WIFI adhoc part.
- The client would be installed on the last node of CSMA part.

- The maximum number of packets the application will send is 100.
- The time to wait between packets is 1 second.
- Size of echo data in the packets is 1024 bytes.
- Port number is 9.
- Run your applications for 10 seconds.

```
UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (wifiadNodes.Get (nWifi));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (11.0));

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 (12.0));

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

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

Generate NetAnim xml file

```
  AnimationInterface anim("/home/shenhapy/Downloads/ns-allinone-3.35/mythird.xml");

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

2. A technical reflection on the assignment: Write a brief about my approach and the challenges you faced during this assignment. Reflect on the following:

   • Difficulties encountered and solutions implemented while setting up the UDP Echo Client and Server application.

   While setting up the UDP Echo Client and Server application in the network simulation, one of the difficulties arose from the complex network topology, involving three networks: adhoc wifi, csma, and p2p. The task of defining the points for the p2p nodes, considering their connection with different csma network and wif network. However, by carefully analyzing the network structure and adjusting, we were able to overcome this challenge and successfully define the points.

   • Our insights about the interplay between P2P ,CSMA and wifi network topologies, gained through this assignment.

   Regarding the experience and learning gained from adding the animation module to the script, it was an insightful process. We already used NetAnim in last task and then integrated it into the script of current task. The simulation assisted in identifying and rectifying configuration errors in the network setup, such as inaccurately specifying the number of nodes for one of the csma network and the server for sending packets. NetAnim proved to be a powerful tool for visualizing the network and aided in rearranging the nodes to achieve better visualization.

   • Experience and learning about network simulation from the process of adding the animation module to your script.

   Through this assignment, I gained insights into the interplay between Adhoc Wifi, P2P and CSMA network topologies. The P2P topology exhibited minimal delay and a lower data rate since it established a dedicated, high-speed connection between two nodes. The CSMA networks represented shared Ethernet-like environments, resulting in higher delay but a higher data rate as the communication was extended to multiple points. The Wifi which will be represented as some random walk network members having many random states which is challenging to adjust.

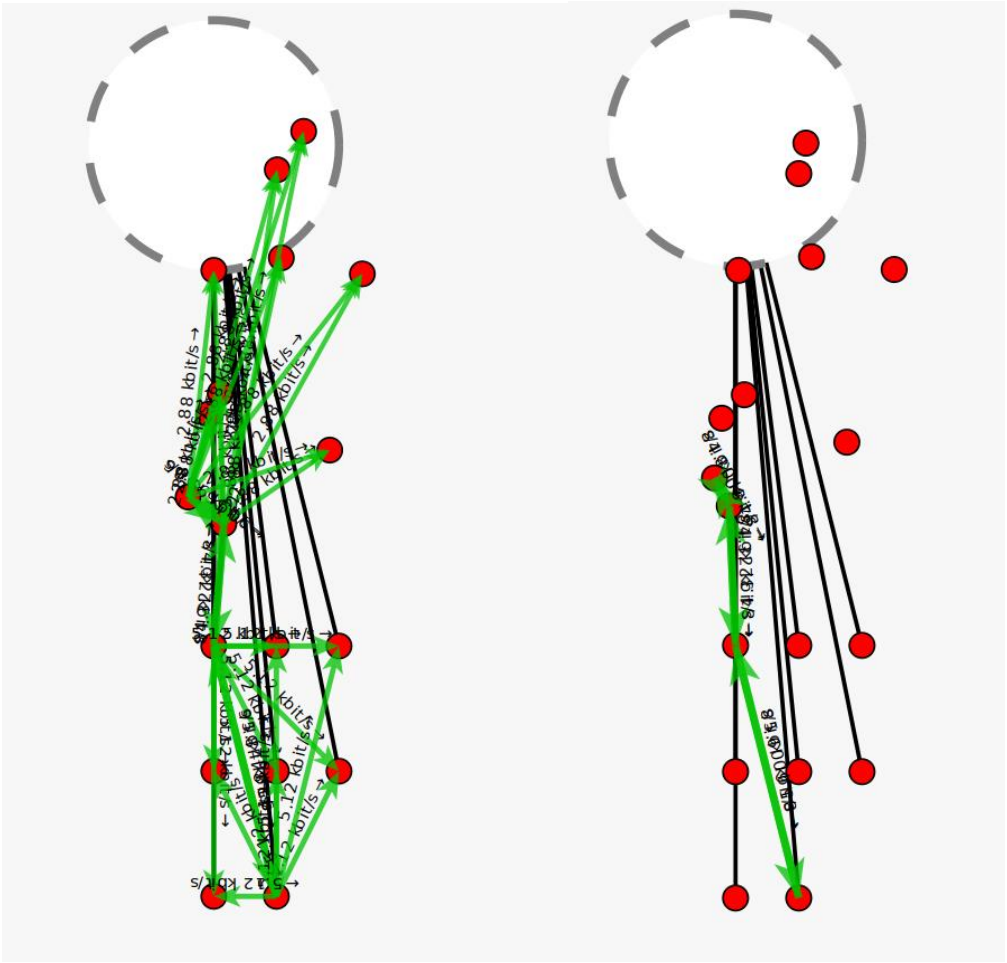
1-2- Results

Please provide your output (figures, texts, …):

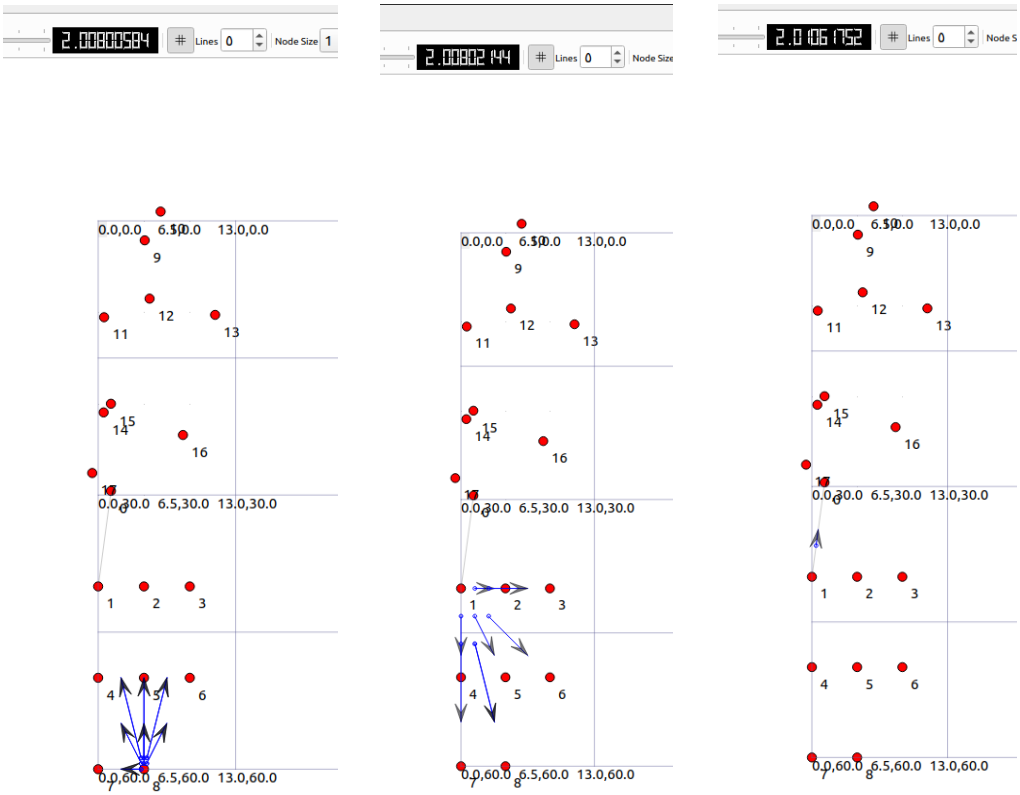
1. A snapshot of the command line output showing the log of sent and received packets

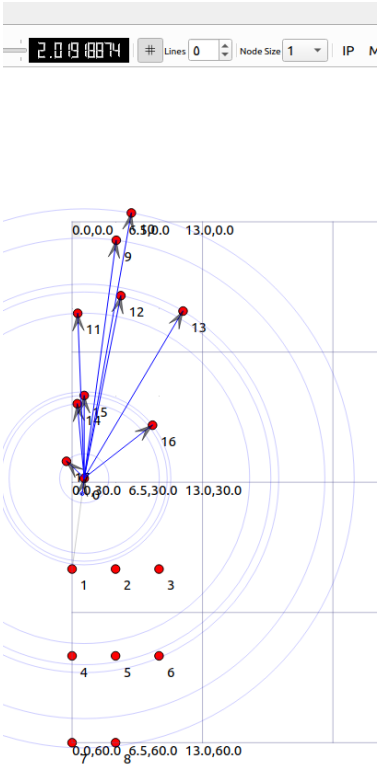
```
Waf: Entering directory `/home/shenhapy/Downloads/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/shenhapy/Downloads/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.367s)
At time +2s client sent 1024 bytes to 20.1.2.10 port 9
At time +2.01947s server received 1024 bytes from 20.1.3.8 port 49153
At time +2.01947s server sent 1024 bytes to 20.1.3.8 port 49153
At time +2.037s client received 1024 bytes from 20.1.2.10 port 9
At time +3s client sent 1024 bytes to 20.1.2.10 port 9
At time +3.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +3.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +3.01235s client received 1024 bytes from 20.1.2.10 port 9
At time +4s client sent 1024 bytes to 20.1.2.10 port 9
At time +4.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +4.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +4.01235s client received 1024 bytes from 20.1.2.10 port 9
At time +5s client sent 1024 bytes to 20.1.2.10 port 9
At time +5.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +5.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +5.01235s client received 1024 bytes from 20.1.2.10 port 9
At time +6s client sent 1024 bytes to 20.1.2.10 port 9
At time +6.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +6.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +6.01235s client received 1024 bytes from 20.1.2.10 port 9
At time +7s client sent 1024 bytes to 20.1.2.10 port 9
At time +7.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +7.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +7.01235s client received 1024 bytes from 20.1.2.10 port 9
At time +8s client sent 1024 bytes to 20.1.2.10 port 9
At time +8.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +8.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +8.01235s client received 1024 bytes from 20.1.2.10 port 9
At time +9s client sent 1024 bytes to 20.1.2.10 port 9
At time +9.00615s server received 1024 bytes from 20.1.3.8 port 49153
At time +9.00615s server sent 1024 bytes to 20.1.3.8 port 49153
At time +9.01235s client received 1024 bytes from 20.1.2.10 port 9
shenhapy@shenhapy:~/Downloads/ns-allinone-3.35/ns-3.35$
```

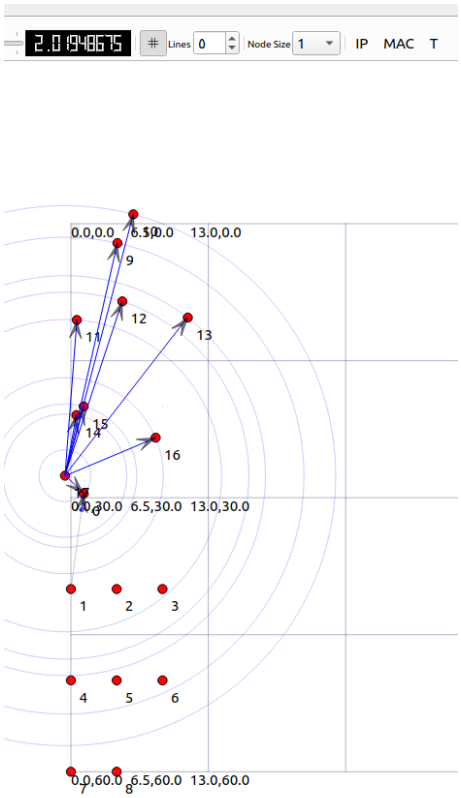
2. A snapshot of the visualization using Vis

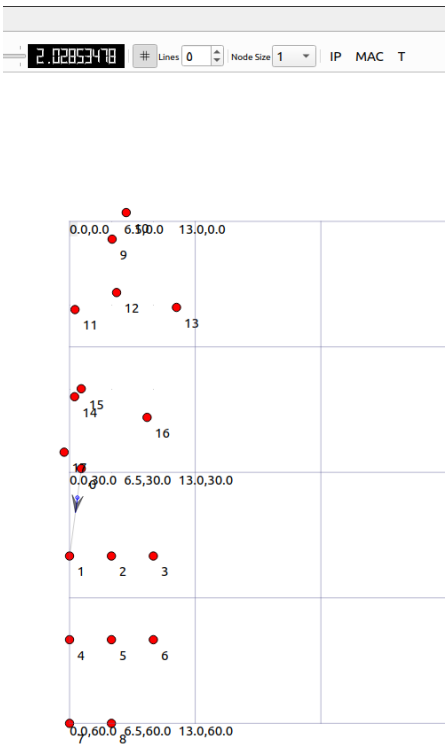


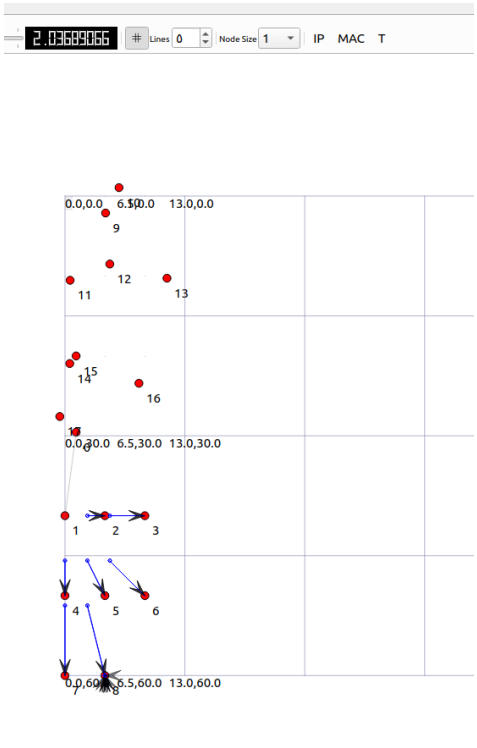
3. A snapshot of the visualization using NetAnim

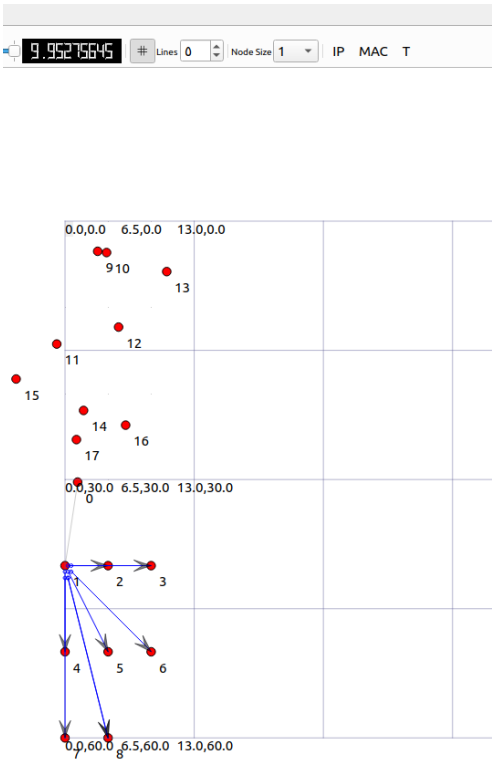



2.0194874
Lines 0
Node Size 1
IP M
0.0,0.0 6.5,0.0 13.0,0.0
0.0,30.0 6.5,30.0 13.0,30.0
0.0,60.0 6.5,60.0 13.0,60.0


2.0194875
Lines 0
Node Size 1
IP MAC T
0.0,0.0 6.5,0.0 13.0,0.0
0.0,30.0 6.5,30.0 13.0,30.0
0.0,60.0 6.5,60.0 13.0,60.0


2.02853478
Lines 0
Node Size 1
IP MAC T
0.0,0.0 6.5,0.0 13.0,0.0
0.0,30.0 6.5,30.0 13.0,30.0
0.0,60.0 6.5,60.0 13.0,60.0


2.03689066
Lines 0
Node Size 1
IP MAC T
0.0,0.0 6.5,0.0 13.0,0.0
0.0,30.0 6.5,30.0 13.0,30.0
0.0,60.0 6.5,60.0 13.0,60.0


9.95275645
Lines 0
Node Size 1
IP MAC T
0.0,0.0 6.5,0.0 13.0,0.0
0.0,30.0 6.5,30.0 13.0,30.0
0.0,60.0 6.5,60.0 13.0,60.0

1-3- List of files

Please mention the name of the files you have attached in Brightspace. Also, attach this file and modified code file in Brightspace.

1-Code_GP12.cpp

2-Report_GP12.pdf

mythird.xml

third-0-0.pcap

third-0-1.pcap

third-1-0.pcap

third-1-1.pcap