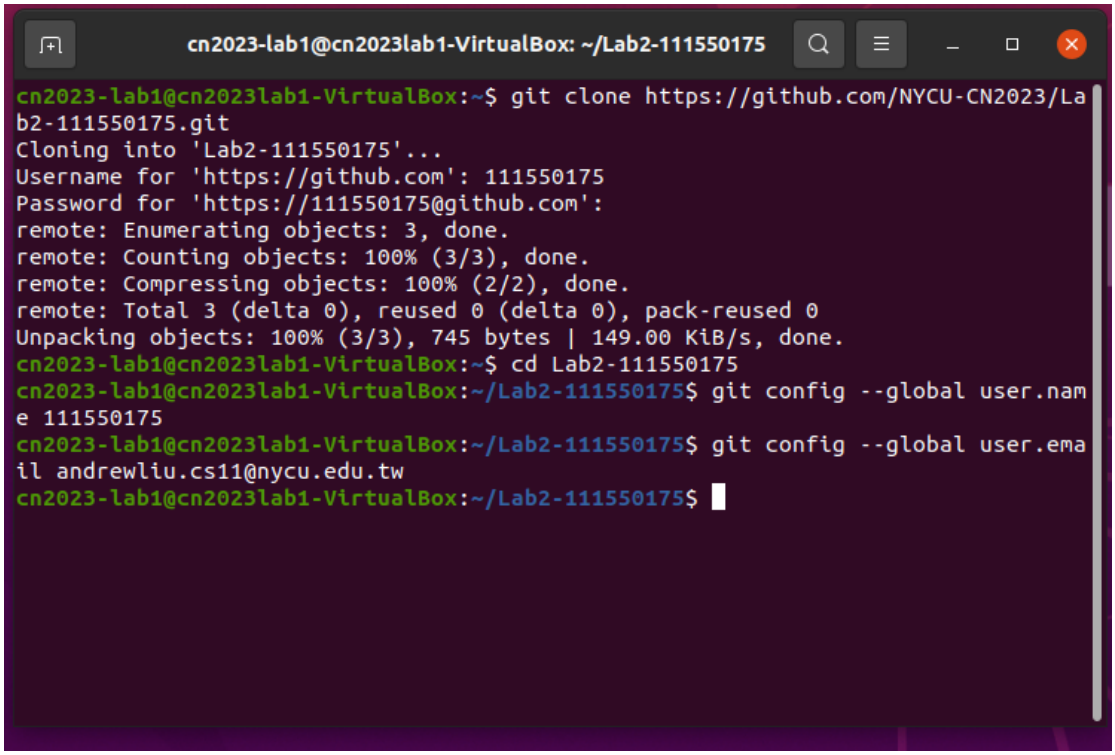
Computer Network Lab2 Report

1.Describe each step and how to run your program:

Task 1:

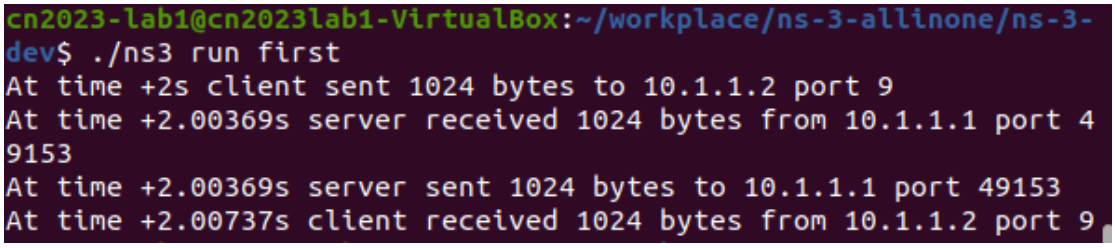
(1) Set up the environment:



(2) Follow the editorial to install NS-3

Task 2:

(1) Run the example code:



Task 3:

(1) Change the topology:

```
42
43      NodeContainer nodes;
44      nodes.Create(3);
45
46      PointToPointHelper pointToPoint;
47      pointToPoint.SetDeviceAttribute("DataRate", StringValue("2Mbps"));
48      pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
49      PointToPointHelper pointToPoint1;
50      pointToPoint1.SetDeviceAttribute("DataRate", StringValue("3Mbps"));
51      pointToPoint1.SetChannelAttribute("Delay", StringValue("2ms"));
52
53      NetDeviceContainer devices;
54      devices = pointToPoint.Install(nodes.Get(0), nodes.Get(1));
55      NetDeviceContainer devices1;
56      devices1= pointToPoint1.Install(nodes.Get(0), nodes.Get(2));
57
58      InternetStackHelper stack;
59      stack.Install(nodes);
60
61      Ipv4AddressHelper address;
62      address.SetBase("10.0.1.0", "255.255.255.0");
63      Ipv4AddressHelper address1;
64      address1.SetBase("10.0.2.0", "255.255.255.0");
65
66      Ipv4InterfaceContainer interfaces = address.Assign(devices);
67      Ipv4InterfaceContainer interfaces1 = address1.Assign(devices1);
68
69      UdpEchoServerHelper echoServer(99);
70      UdpEchoServerHelper echoServer1(98);
```

Task 4:

(1) The .cc file should include one client and two servers, with each of two flows sending four UDP packets:

```
72      ApplicationContainer serverApps = echoServer.Install(nodes.Get(1));
73      serverApps.Start(Seconds(1.0));
74      serverApps.Stop(Seconds(10.0));
75      ApplicationContainer serverApps1 = echoServer1.Install(nodes.Get(2));
76      serverApps1.Start(Seconds(1.0));
77      serverApps1.Stop(Seconds(10.0));
78
79      UdpEchoClientHelper echoClient(interfaces.GetAddress(1), 99);
80      echoClient.SetAttribute("MaxPackets", UintegerValue(4));
81      echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));
82      echoClient.SetAttribute("PacketSize", UintegerValue(1024));
83      UdpEchoClientHelper echoClient1(interfaces1.GetAddress(1), 98);
84      echoClient1.SetAttribute("MaxPackets", UintegerValue(4));
85      echoClient1.SetAttribute("Interval", TimeValue(Seconds(1.0)));
86      echoClient1.SetAttribute("PacketSize", UintegerValue(1024));
87
88      ApplicationContainer clientApps = echoClient.Install(nodes.Get(0));
89      clientApps.Start(Seconds(2.0));
90      clientApps.Stop(Seconds(10.0));
91      ApplicationContainer clientApps1 = echoClient1.Install(nodes.Get(0));
92      clientApps1.Start(Seconds(2.0));
93      clientApps1.Stop(Seconds(10.0));
94
```

(2) Output:

```
cn2023-lab1@cn2023lab1-VirtualBox:~/workplace/ns-3-allinone/ns-3-dev$ ./ns3 run 111550175
Scanning dependencies of target scratch_111550175
[  0%] Building CXX object scratch/CMakeFiles/scratch_111550175.dir/111550175.cc.o
[  0%] Linking CXX executable ../../build/scratch/ns3-dev-111550175-default
At time +2s client sent 1024 bytes to 10.0.1.2 port 99
At time +2s client sent 1024 bytes to 10.0.2.2 port 98
At time +2.00481s server received 1024 bytes from 10.0.2.1 port 49154
At time +2.00481s server sent 1024 bytes to 10.0.2.1 port 49154
At time +2.00622s server received 1024 bytes from 10.0.1.1 port 49153
At time +2.00622s server sent 1024 bytes to 10.0.1.1 port 49153
At time +2.00962s client received 1024 bytes from 10.0.2.2 port 98
At time +2.01243s client received 1024 bytes from 10.0.1.2 port 99
At time +3s client sent 1024 bytes to 10.0.1.2 port 99
At time +3s client sent 1024 bytes to 10.0.2.2 port 98
At time +3.00481s server received 1024 bytes from 10.0.2.1 port 49154
At time +3.00481s server sent 1024 bytes to 10.0.2.1 port 49154
At time +3.00622s server received 1024 bytes from 10.0.1.1 port 49153
At time +3.00622s server sent 1024 bytes to 10.0.1.1 port 49153
At time +3.00962s client received 1024 bytes from 10.0.2.2 port 98
At time +3.01243s client received 1024 bytes from 10.0.1.2 port 99
At time +4s client sent 1024 bytes to 10.0.1.2 port 99
At time +4s client sent 1024 bytes to 10.0.2.2 port 98
At time +4.00481s server received 1024 bytes from 10.0.2.1 port 49154
At time +4.00481s server sent 1024 bytes to 10.0.2.1 port 49154
At time +4.00622s server received 1024 bytes from 10.0.1.1 port 49153
At time +4.00622s server sent 1024 bytes to 10.0.1.1 port 49153
At time +4.00962s client received 1024 bytes from 10.0.2.2 port 98
At time +4.01243s client received 1024 bytes from 10.0.1.2 port 99
At time +5s client sent 1024 bytes to 10.0.1.2 port 99
At time +5s client sent 1024 bytes to 10.0.2.2 port 98
At time +5.00481s server received 1024 bytes from 10.0.2.1 port 49154
At time +5.00481s server sent 1024 bytes to 10.0.2.1 port 49154
At time +5.00622s server received 1024 bytes from 10.0.1.1 port 49153
At time +5.00622s server sent 1024 bytes to 10.0.1.1 port 49153
At time +5.00962s client received 1024 bytes from 10.0.2.2 port 98
At time +5.01243s client received 1024 bytes from 10.0.1.2 port 99
```

QA:

(1) What is the different between network simulation and emulation?

Network simulation usually only simulate the network through the computer, that is, only use one end-system to construct a virtual network and calculate the output. But network emulation often uses real hardware such as routers, switches, etc. to construct a similar network. Then, calculate the output based on the emulated network.

(2) Generally, in NS-3, if you don’t change the code, the output will be always the same every time you run, even if you set some probabilistic parameter like error rate, why?

By default, NS-3 simulations use a fixed seed and deterministic random number generator, so with the fixed seed, the deterministic random number generator with generate the same output.

(3) Following the previous question, how to deal with this problem?

Because the randomness has been restricted by the fixed seed, so we can set a new seed to generate the different output.

Bonus:

(1) What have you learned from this lab?

I have learned how to simulate a simple topology network by using NS-3 and c++, also some basic command or operation on Linux operating system.

(2) What difficulty have you met in this lab?

Actually, I think this lab is much more easier than lab1; however, I still met one difficulty in this lab. That difficulty is to write the code that can simulate the topology. Though I have written c++, it is still a little hard for me to understand the editorial and write the code. Moreover, I had went on the internet to search some other reference and watch some lecture videoes to completely understand how to write the code.