111652017 廖修誼 Report

1. Describe each step and how to run your program

Task 1. Environment Setup

- I just followed the homework guidence step by step

Task 2. Trace Example Code

- I will illustrate this part with the following description.

Task 3. Create a New Topology

First, we create three nodes and constructe a point to point link. (modified by the example code)

```
    // create three nodes
    nodes.Create(3); // modify from 2 to 3
    // constructe a point to point link, and set their
    // DataRate and delay.
    PointToPointHelper pointToPoint1;
    pointToPoint1.SetDeviceAttribute("DataRate", StringValue("2Mbps"));
    pointToPoint1.SetChannelAttribute("Delay", StringValue("2ms"));
    NetDeviceContainer devices1 =
        pointToPoint1.Install(nodes.Get(0), nodes.Get(1));
        // flow 1

    PointToPointHelper pointToPoint3;
    pointToPoint3.SetDeviceAttribute("DataRate", StringValue("3Mbps"));
    pointToPoint3.SetChannelAttribute("Delay", StringValue("2ms"));
    NetDeviceContainer devices3 =
        pointToPoint3.Install(nodes.Get(0), nodes.Get(2));
        // flow 2 // modify
```

Second, we set up protocol stacks installed on our nodes.

```
    InternetStackHelper stack;
    stack.Install(nodes);
```

Third, we assign the IP address of all nodes. (modified by the example code)

```
    Ipv4AddressHelper address1;
    //  it should begin allocating IP addresses from the network
    //  10.0.1.0 using the mask 255.255.255.0 to define the
    //  allocatable bits.
    address1.SetBase("10.0.1.0", "255.255.255.0");
    Ipv4InterfaceContainer interfaces1 = address1.Assign(devices1);

    Ipv4AddressHelper address2;
    //  it should begin allocating IP addresses from the network
    //  10.0.2.0 using the mask 255.255.255.0 to define the
    //  allocatable bits.
    address2.SetBase("10.0.2.0", "255.255.255.0");
    Ipv4InterfaceContainer interfaces2 = address2.Assign(devices3);
```

Task 4. Create New Flows (Cont.)

First, we create two ApplicationContainer, they can help us to generate traffic, and we set up the starting and stopping time. And when I use them, I need to know what port and what IP I use. This part is also modified by example code.

```
    UdpEchoServerHelper echoServer1(99); // modify the port number
    ApplicationContainer serverApps1 = echoServer1.Install(nodes.Get(1));
    serverApps1.Start(Seconds(1.0));
    serverApps1.Stop(Seconds(10.0));
```

Second, we set up some setting about the flows. This part is also modified by example code.

```
    UdpEchoClientHelper echoClient1(interfaces1.GetAddress(1), 99);
    // modify the port number
    echoClient1.SetAttribute("MaxPackets", UintegerValue(4));
    // since we send '4' UDP packets
    echoClient1.SetAttribute("Interval", TimeValue(Seconds(1.0)));
    echoClient1.SetAttribute("PacketSize", UintegerValue(1024));
    ApplicationContainer clientApps1 = echoClient1.Install(nodes.Get(0));
    clientApps1.Start(Seconds(2.0));
    clientApps1.Stop(Seconds(10.0));
```

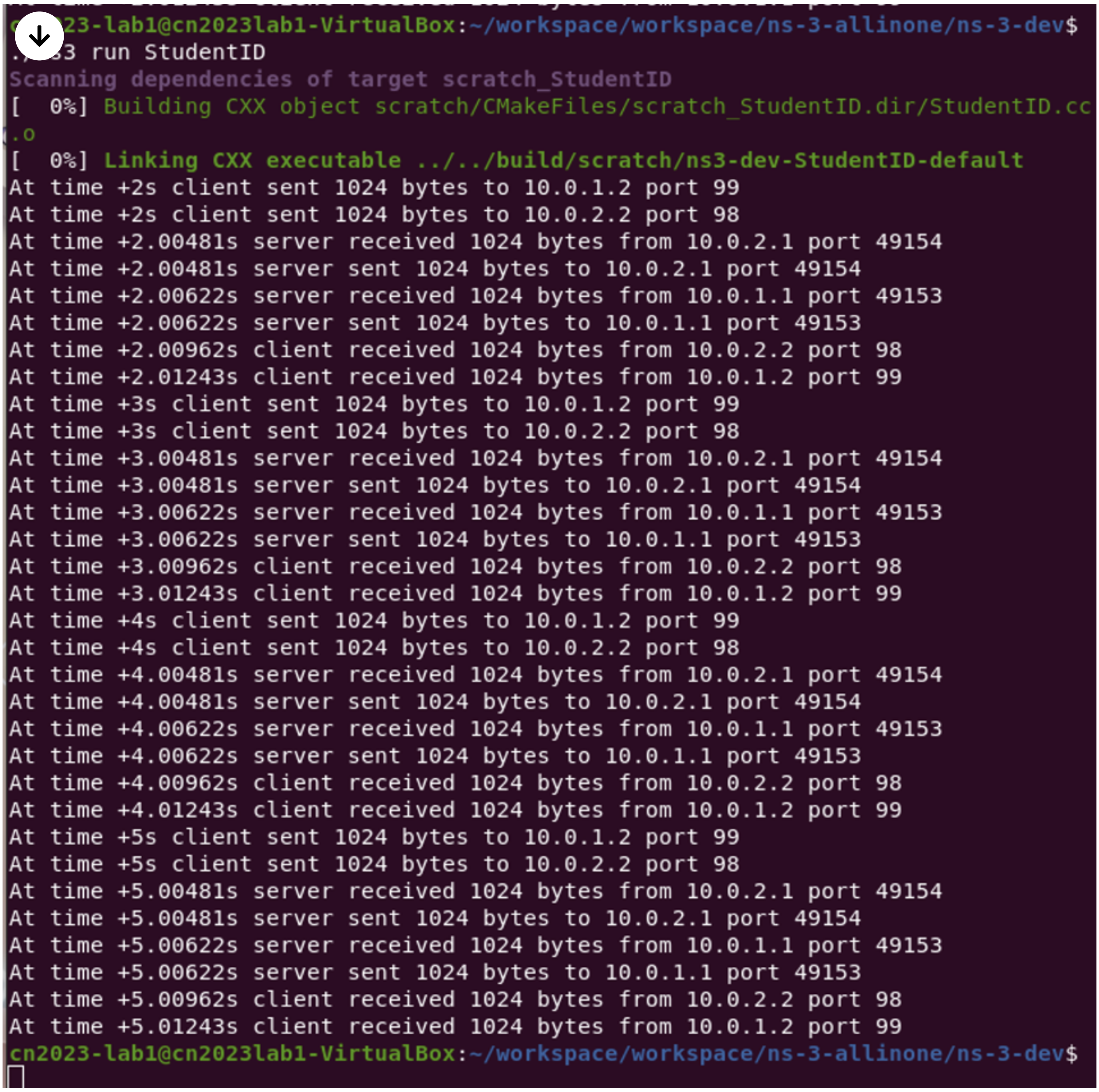
Similarily, we can infer the another flows as following.

```
    UdpEchoServerHelper echoServer2(98);
    ApplicationContainer serverApps2 = echoServer2.Install(nodes.Get(2));
    serverApps2.Start(Seconds(1.0));
    serverApps2.Stop(Seconds(10.0));
    UdpEchoClientHelper echoClient2(interfaces2.GetAddress(1), 98);
    echoClient2.SetAttribute("MaxPackets", UintegerValue(4));
    echoClient2.SetAttribute("Interval", TimeValue(Seconds(1.0)));
    echoClient2.SetAttribute("PacketSize", UintegerValue(1024));
    ApplicationContainer clientApps2 = echoClient2.Install(nodes.Get(0));
    clientApps2.Start(Seconds(2.0));
    clientApps2.Stop(Seconds(10.0));
```

Finally, we run simulator.

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

Final Result



```
cn2023-lab1@cn2023lab1-VirtualBox:~/workspace/workspace/ns-3-allinone/ns-3-dev$
./ns3 run StudentID
Scanning dependencies of target scratch_StudentID
[  0%] Building CXX object scratch/CMakeFiles/scratch_StudentID.dir/StudentID.cc
.o
[  0%] Linking CXX executable ../../build/scratch/ns3-dev-StudentID-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
cn2023-lab1@cn2023lab1-VirtualBox:~/workspace/workspace/ns-3-allinone/ns-3-dev$
```

2. What is the different between network simulation and emulation ?

- Network simulators create models, while emulators mimic networks.

3. 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 ?

- Yes. The first results are the same bacause two consecutive runs have the same seed. So their results are the same.
- No. Because when I change the setting, the new seed will be picked randomly, so the result will be different with high probability.

4. Following the previous question, how to deal with this problem ?

- We can manually modify the seed when making adjustments.

5. Bonus

1. What have you learned from this lab?

- I learn how to use ns3 model.

2. What difficulty have you met in this lab?

- I think wrong the function of the following.

```
    pointToPoint1.Install(nodes.Get(0), nodes.Get(1));
```

- This will build a bidirection link, instead of one direction.