Communication Networks 2

SS 2021

Assignment 4

Group 06

Name	Mat.Nummer
Paul Kloker	12034928
Juan Aramis Oposich	11701238

June 21, 2021

1 Task description and setup

The task of the last assignment is to model a network in NS-3, which is an open source network simulator, according to the following network diagram (figure 1). Furthermore, a 10 seconds long simulation with a ping measurement from node 1 to node 3 shall be done and the generated measurement data discussed.

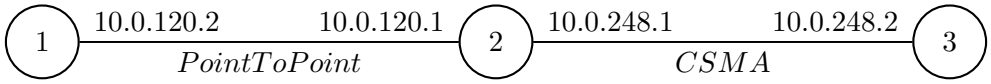

Figure 1: Network topology

For this the latest version of NS-3 was downloaded form `https://www.nsmam.org` and build with the command `$ ./bulid.py`.

2 NS3 Model

NS-3 models are C++ based and consist out of network nodes, network devices, communication channels and applications. Most of the code is based on the second tutorial in the example folder. Listing 1 shows the creation of the topology Because our system consists out of one PointToPoint connection and one Carrier Sense Multiple Access (CSMA) network, different kind of nodes were created and the given channel attributes were set by using the helper objects. To finish the topology creation helper objects were used to install the network devices and a IP stack for every node.

```
//P2P
NodeContainer p2pNodes;
p2pNodes.Create (2);

PointToPointHelper pointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate",
                                 StringValue ("11Mbps"));
pointToPoint.SetChannelAttribute ("Delay",
                                  StringValue ("80ms"));

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

//CSMA
NodeContainer csmaNodes;
```

```
csmaNodes.Add (p2pNodes.Get (0));
csmaNodes.Create (1);

CsmaHelper csma;
csma.SetChannelAttribute ("DataRate",
                          StringValue ("70Mbps"));
csma.SetChannelAttribute ("Delay",
                          TimeValue (MicroSeconds (1000)));

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

InternetStackHelper stack;the
stack.Install (p2pNodes.Get (1));
stack.Install (csmaNodes);
```

Listing 1: Topology creation

Node No.	**NS-3 Node Container Element**	**IP Address**
1	p2pNodes.Get(1)	10.0.120.2
2	p2pNodes.Get(0)	10.0.120.1
2	csmaNodes.Get(0)	10.0.248.1
3	csmaNodes.Get(1)	10.0.248.2

Table 1: Nodes and the according NS-3 node container element

To create a routing table with the `Ipv4GlobalRoutingHelper`, both subnets first got the right address space assigned to them (see 2).

```
Ipv4AddressHelper address;
address.SetBase ("10.0.120.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("10.0.248.0", "255.255.255.0");
Ipv4InterfaceContainer csmaInterfaces;
csmaInterfaces = address.Assign (csmaDevices);

Ipv4GlobalRoutingHelper::PopulateRoutingTables ();
```

Listing 2: IP address assignment and routing table creation

Listing 2 shows how the ping to the third node was created and installed on the first node. The attribute `Verbose` was set to true to create a ping-style output. The payload was set to 1024 bytes and the interval between the ping messages to 0.2 Seconds. The last two lines of section 2 define how long the ping application will run during the simulation

```
V4PingHelper ping = V4PingHelper (
csmaInterfaces.GetAddress (1));
ping.SetAttribute ("Verbose", BooleanValue (true));
ping.SetAttribute ("Size", UintegerValue(1024));
ping.SetAttribute ("Interval", TimeValue(Seconds(0.2)));
ApplicationContainer apps = ping.Install (p2pNodes.Get (1));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (10.0));
}
```

Before the simulation was invoked the stop time was specified and packet capturing was enabled for all P2P and CSMA nodes (see listing 3).

```
csma.EnablePcapAll("assignment4", true);
pointToPoint.EnablePcapAll ("assignment4", true);

Simulator::Stop (Seconds (10.0));
Simulator::Run ();
Simulator::Destroy ();
}
```

Listing 3: pcap and simulation start

The code was then compiled and execute with `waf`. This created the ping measurement and pcap data which will be discussed in the next section.

3 Data analysis

This chapter supports the understanding of the simulation model. Network model are been used to further estimate a real world scenario without a concrete build. Those models contains nodes and links just like a real internet system. For this problem we used NS-3 deploys a discrete-event simulator.

The assignment describes the needed network parameter like the network topology and the associated metrics. One task is to setup one point to point connection and also a CSMA link. those connections are implemented with **PointToPointHelper** and **CsmaHelper**. This classes are used to set up the IP addresses, MAC address and also channel delay. The channel delays, also propagation delays, is the time that it takes for a bit to reach from one end of a link to the other. The delay depends on the distance between the sender and the receiver, and the propagation speed of the wave signal.

4 Conclusion