

TERMWORK - 6

Problem Definition

Simulate a Full duplex connection in a wired network using NS3.

Course Learning Objectives

- To design & simulate the network in latest simulation tools
- To perform the real time network traffic analysis using network monitoring tools.

Theory

A First ns-3 Script

If you downloaded the system as suggested, you will have a release of ns-3 in a directory called repos under your home directory. Change into release directory. Change into the examples/tutorial directory. You should see a file named first.cc located there. This is a script that will create a simple point-to-point link between two nodes & echo a single packet between the nodes.

Boilerplate

The first line in the file is an emacs mode line. This tells emacs about the formatting conventions we use in our source code.

The ns-3 simulator is licensed using the GNU General Public License.

Module Includes

The code proper starts with a number of include statements

```
#include "ns3/core-module.h"
```

```
#include "ns3/simulator-module.h"
```

```
#include "ns3/node-module.h"
```

```
#include "ns3/helper-module.h"
```

./waf -d debug configure is used to configure the project to perform debug builds.

./waf to build the project.

Ns 3 Namespace

The next line in first.cc script is a namespace declaration using namespace ns3;

Logging

The next line of the script is the following,
NS_LOG_COMPONENT_DEFINE("FirstScriptExample");

Topology Helpers

NodeContainer

The next two lines of code in our script will actually create the ns-3 Node objects that will represent the computers in the simulation.

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

PointToPointHelper

The next three lines in the script are,

```
PointToPointHelper pointToPoint;  
pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));  
pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
```

The first line, instantiates a PointToPointHelper object to use the on the stack.

The next line, tells the PointToPointHelper object to use the value "5Mbps" as the DataRate.

The next line, tells the PointToPointHelper object to use the value "2ms" as the value of the transmission delay.

NetDeviceContainer

NetDeviceContainer devices;

devices = pointToPoint.Install(nodes);

These codes will finish configuring the devices & channel.

InternetStackHelper

It is a topology helper that is to internet stacks what the PointToPointHelper is to point-to-point net devices.

IPv4AddressHelper

The next two lines of code in first.cc, declare an address helper object & tell it that it should begin allocating IP addresses from the network 10.1.1.0 using the mask 255.255.255.0 to define the allocatable bits.

The next line performs the actual address assignment.

UdpEchoServerHelper

These lines are used to set up a UDP echo server application on one of the nodes we previously created.

UdpEchoClientHelper

The echo client application is set up in a method substantially similar to that for the server. There is an underlying UdpEchoClientApplication that is managed by an UdpEchoClientHelper.

Simulator

This is done using the global function
Simulator::Run();

Source Code

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"
```

```
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
int main (int argc, char *argv[])
{
```

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

```
    Time::SetResolution (Time::NS);
    LogComponentEnable ("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable ("UdpEchoServerApplication", LOG_LEVEL_INFO);
```

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

```
    PointToPointHelper pointToPoint;
    pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
    pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

```
    NetDeviceContainer devices;
    devices = pointToPoint.Install (nodes);
```



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

```
IPv4AddressHelper address;  
address.SetBase("10.1.1.0", "255.255.255.0");  
IPv4InterfaceContainer interfaces = address.Assign(devices);  
UdpEchoServerHelper echoServer(9);
```

```
ApplicationContainer serverApps = echoServer.Install(nodes.Get(1));  
serverApps.Start(Seconds(1.0));  
serverApps.Stop(Seconds(10.0));
```

```
UdpEchoClientHelper echoClient(interfaces, GetAddress(1), 9);  
echoClient.SetAttribute("MaxPackets", UintegerValue(1));  
echoClient.SetAttribute("Interval", TimeValue(Seconds(1.0)));  
echoClient.SetAttribute("PacketSize", UintegerValue(1024));
```

```
ApplicationContainer clientApps = echoClient.Install(nodes.Get(0));  
clientApps.Start(Seconds(2.0));  
clientApps.Stop(Seconds(10.0));
```

```
AnimationInterface anim("first.xml");  
AsciiTraceHelper ascii;  
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("first.tr"));  
pointToPoint.EnablePcapAll("first");
```

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

Conclusion

We ~~learn~~ successfully simulated a full duplex connection in a wired network using NS3.

Course Outcomes

We learnt the simulation of full duplex connection using ns3.

References

- <https://ns3simulator.com>
- <https://www.nsnam.org>