LAB2 - More about ns-3
CS169: Mobile Wireless Networks - Winter 2017

Kittipat Apicharttrisorn (Patrick)
Department of Computer Science and Engineering
University of California, Riverside

January 23-24, 2017
Table of Contents

1. Logging Modules
2. Command Line Arguments
3. Tracing Systems
NS-3 Logging

- Go to working directory
  
  $ cd
  /extra/CSUserName/cs169lab/ns-allinone-3.25/ns-3.25

- cp examples/tutorial/first.cc scratch/myfirst.cc

- ./waf

- ./waf --run scratch/myfirst

- Observe the logs... What cause them?

- `LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);`
Log Verbosity Levels

LOG_ALL
LOG_LOGIC
LOG_FUNCTION
LOG_INFO
LOG_DEBUG
LOG_WARN
LOG_ERROR
LOG_LEVEL_INFO

More verbose
$ export NS_LOG=UdpEchoClientApplication=level_all

$ export 'NS_LOG=UdpEchoClientApplication=level_all | prefix_func'

$ export 'NS_LOG=UdpEchoClientApplication=level_all | prefix_func : UdpEchoServerApplication=level_all | prefix_func'

$ export 'NS_LOG=UdpEchoClientApplication=level_all | prefix_func | prefix_time: UdpEchoServerApplication=level_all | prefix_func | prefix_time'

Log-all-application Masking

export 'NS_LOG=*=level_all | prefix_func | prefix_time'
Adding your own logs

- $ vim scratch/myfirst.cc
- `NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");`
- `NS_LOG_INFO("Creating Topology");
  NodeContainer nodes;
  nodes.Create (2);
- $ ./waf
- Clear NS_LOG variables
- $ export NS_LOG=
- Running...
- $ ./waf --run scratch/myfirst
- Do you see the log?
Seeing your own logs

- $ export NS_LOG=FirstScriptExample=info
- $ ./waf --run scratch/myfirst
- $ export NS_LOG=FirstScriptExample=level_info
- $ ./waf --run scratch/myfirst
When we wanted to alter attribute values, ...

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

Now, we want to pass these as command line arguments, then go to myfirst.cc and declare the command line parser

CommandLine cmd;
cmd.Parse(argc, argv);
Looking for help...

- $ ./waf --run "scratch/myfirst --PrintHelp"
- Print attributes
- $ ./waf --run "scratch/myfirst --PrintAttributes=ns3::PointToPointNetDevice"
- Remove these two lines from the script
  ```
  pointToPoint.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
  pointToPoint.SetChannelAttribute("Delay", StringValue("2ms"));
  ```
- Set attributes
- $ ./waf --run "scratch/myfirst --ns3::PointToPointNetDevice::DataRate=5Mbps"

Do not forget

Double quotes when passing arguments
Exercise 1

- Insert more LOG_INFO for FirstScriptExample
- Ex. Creating Topology, Assigning IP Addresses, Creating echoServer
- What is the default channel delay value?
- Set channel delay to 4 ms using CLA
- Set max packets to 4 using CLA
- Set data rate, channel delay, and max packets at the same run using CLA

Hints!

$ ./waf --run "scratch/myfirst --PrintGroup=PointToPoint"
$ ./waf --run "scratch/myfirst --ns3::UdpEchoClient::MaxPackets=4"
Using your own values

- Just before `cmd.Parse(argc, argv);` add:
  ```c
  uint32_t nPackets = 1;
  cmd.AddValue("nPackets", "Number of packets to echo", nPackets);
  ```
- Type:
  ```c
  echoClient.SetAttribute("MaxPackets", UintegerValue(nPackets));
  ```
- $ ./waf --run "scratch/myfirst --nPackets=4"

Good to know

Using your own values, you do not have to know ns3 Group and Argument names (ex. ns3::UdpEchoClient::MaxPackets) but knowing some of them allows you to get the default values used by ns3.
How can you look at IP headers by using LOG?

Tracing Systems are extended logging systems that provide users with more customizations.

Trace source → trace sink

ASCII Tracing: Add the following two lines before Simulator::Run

AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll (ascii.CreateFileStream ("myfirst.tr"));

Run the script again

Open myfirst.tr
Figure: ASCII Trace Event Locator

- +: An enqueue operation occurred on the device queue;
- -: A dequeue operation occurred on the device queue;
- d: A packet was dropped, typically because the queue was full;
- r: A packet was received by the net device.

Figure: ASCII Trace Event I

```
1  +
2  2
3 /NodeList/0/DeviceList/0/$ns3::PointToPointNetDevice/TxQueue/Enqueue
4 ns3::PppHeader (  
5     Point-to-Point Protocol: IP (0x0021))
6 ns3::Ipv4Header (  
7     tos 0x0 ttl 64 id 0 protocol 17 offset 0 flags [none]  
8     length: 1052 10.1.1.1 > 10.1.1.2)
9 ns3::UdpHeader (  
10     length: 1032 49153 > 9)  
11 Payload (size=1024)
```
Figure: ASCII Trace Event Locator

- +: An enqueue operation occurred on the device queue;
- -: A dequeue operation occurred on the device queue;
- d: A packet was dropped, typically because the queue was full;
- r: A packet was received by the net device.

Figure: ASCII Trace Event II

```
r
2.25732
/NodeList/1/DeviceList/0/$ns3::PointToPointNetDevice/MacRx
  ns3::Ipv4Header ( 
    tos 0x0 ttl 64 id 0 protocol 17 offset 0 flags [none] 
    length: 1052 10.1.1.1 > 10.1.1.2) 
  ns3::UdpHeader ( 
    length: 1032 49153 > 9) 
  Payload (size=1024)
```
Pcap Tracing

- Add this line of code before `Simulator::Run`
- `pointToPoint.EnablePcapAll ("myfirst");`
- `myfirst` will be the prefix of the real file names: `myfirst-0-0.pcap` and `myfirst-1-0.pcap`

Exercise

Find a computer with Wireshark installed (www.wireshark.org) and remote access to your lab machine (tango or delta) and secure copy (scp) `myfirst-0-0.pcap` and `myfirst-1-0.pcap` from tango or delta to your Wireshark-installed computer. Finally, open pcap files using Wireshark and look around the TCP/IP headers and packet sizes (total packet length vs. data length)
1. Logging Modules

2. Command Line Arguments

3. Tracing Systems
Questions?