LAB1 - Introduction to ns-3

CS169: Mobile Wireless Networks - Winter 2018

Xukan and Thomas

Department of Computer Science and Engineering
University of California, Riverside

January 12, 2017

Table of Contents

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

TA Information

Names	Xukan Ran and Dang Tu Nguyen (Thomas)
Office	WCH367 (Networking Lab)
Office hours	Wed 3-4pm or by appointment
Emails	xran001@ucr.edu (Xukan) and tnguy208@ucr.edu (Thomas)

UNIVERSITY OF CALIFORNIA UCRIVERSIDE

Lab Logistics

- There will be six lab classes and may be exercises for you to practice
- After the sixth lab, you do not have to attend
- Homework will be posted with explicit due dates and times (the same for two lab sections)
- There will be one final project. I will post more information online and will let you know

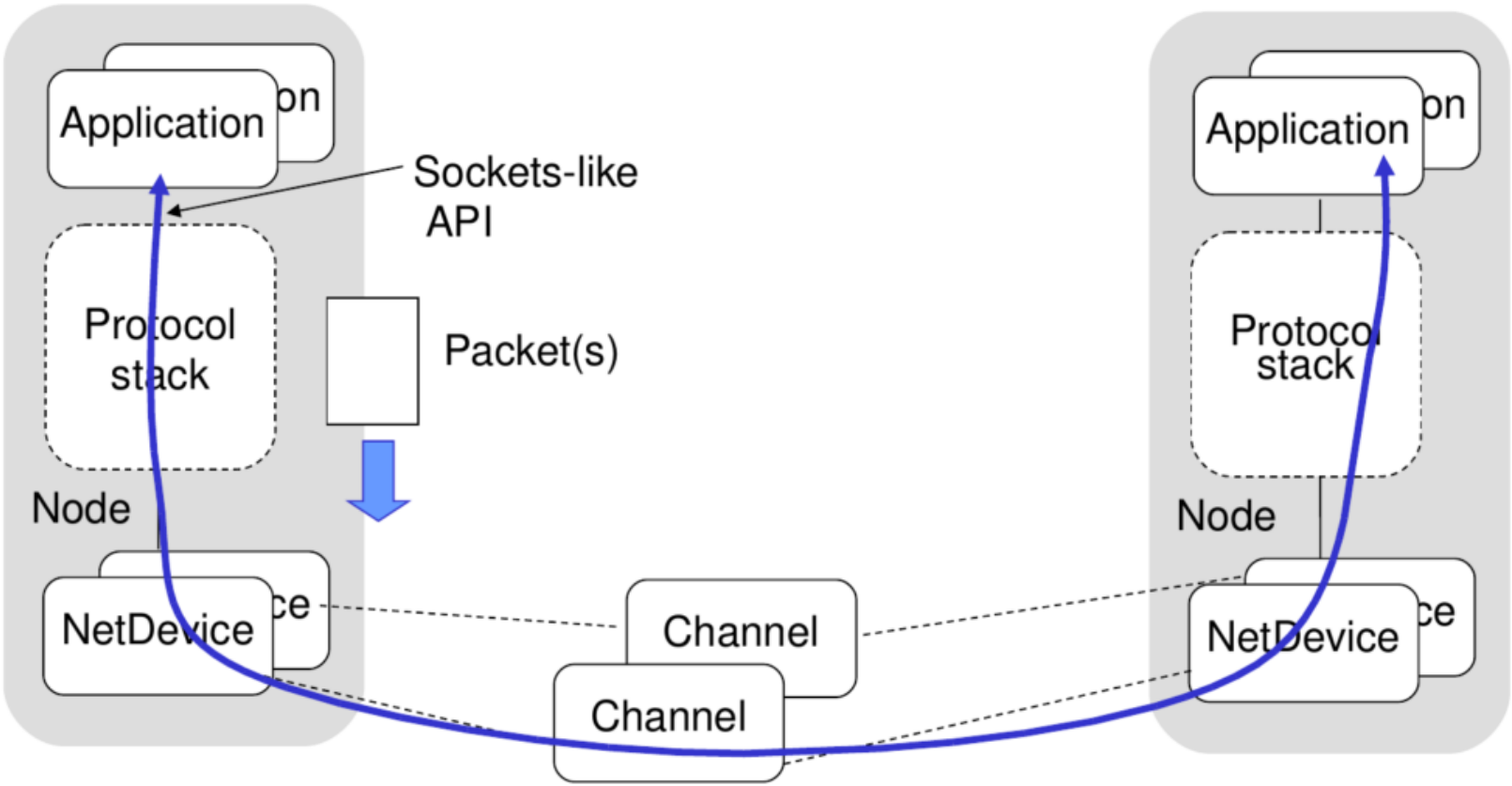
UNIVERSITY OF CALIFORNIA UCRIVERSIDE

What is ns-3?

- A discrete-event network simulator, targeted primarily for research and educational use
- ns-3 is free and publicly available for use
- ns-3 is written entirely with C++ while Python user code wrappers are available.
- We will focus on how to use ns-3 to simulate simple IP networks and WiFi channels

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

Architecture



Architecture

- Just like TCP/IP stacks
- Applications running in a node use Protocol Stack (TCP, UDP, IP, etc)
- Protocol Stack sends packets to NetDevices (or Network interfaces / adapters)
- Each NetDevice interfaces with each Channel (WiFi, LTE, etc.)

UNIVERSITY OF CALIFORNIA UCRIVERSIDE

Download ns-3

- `$ cd /extra/CSUserName`
- `$ mkdir cs169lab && cd cs169lab`
- `$ wget http://www.nsnam.org/release/ns-allinone-3.25.tar.bz2`
- `$ tar -xvf ns-allinone-3.25.tar.bz2`
- `$ cd ns-allinone-3.25`

Addtional commands for remote access

- `$ ssh CSUserName@bolt.cs.ucr.edu`
- `$ ssh wch133-xx`
- where xx is the machine number you are using

Build ns-3

- `$ ./build.py --enable-examples --enable-tests`
- `$ cd ns-3.25`
- `$ ./test.py`
- `$ ./waf --run examples/tutorial/hello-simulator`
- If you see Hello Simulator, congratulations! You have environments ready for running ns-3.

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

Running the first script

- `$ ./waf --run examples/tutorial/first`
- `$ vim examples/tutorial/first.cc`

Script headers

- Add module header files

```
#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"
```

- Namespace

```
using namespace ns3;
```

- Create log components

```
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");
```

More headers

- Set time resolution

```
Time::SetResolution (Time::NS);
```

- Enable log components and set log levels

```
LogComponentEnable ("UdpEchoClientApplication",
LOG_LEVEL_INFO);
LogComponentEnable ("UdpEchoServerApplication",
LOG_LEVEL_INFO);
```

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

Creating topology

- Create nodes

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

- Create Channel

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

- Create NetDevice and bind them to Channel

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

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

Protocol Stack

- Create InternetStack

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

- Set IP network address

```
Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
```

- Assign IP to NetDevice

```
Ipv4InterfaceContainer interfaces = address.Assign
(devices);
```

Building UDP Echo Server

- Create echo server
  ```
  UdpEchoServerHelper echoServer (9);
  ```
- Install echo server to node 1, mark it application, and set start and stop time
  ```
  ApplicationContainer serverApps = echoServer.Install
  (nodes.Get (1));
  serverApps.Start (Seconds (1.0));
  serverApps.Stop (Seconds (10.0));
  ```

Building UDP Echo Client

- Create echo client and set its attributes

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

- Install echo client to node 0, mark it application, and set start and stop time

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

Run simulator

- Run, destroy, return

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

Exercises

- Increase packet sizes 4K, 16K, 64K
- Have the client send the packets every one second for 4 packets
- Double data link rate
- Double data link delay
- Increase the number of echo clients to 2, 3, 4, 5 and have them send packets to the server

UNIVERSITY OF CALIFORNIA UCRIVERSIDE

...What we have learned?

1. TA Information
2. Lab Logistics
3. What is ns-3?
4. ns-3 Installation
5. Running the first ns-3 script
6. Exercises

UNIVERSITY OF CALIFORNIA UCRIVERSIDE

Next Lab...

- Logging modules
- Command line arguments
- Tracing systems
- Pcap tracing

UNIVERSITY OF CALIFORNIA UCRIVERSIDE