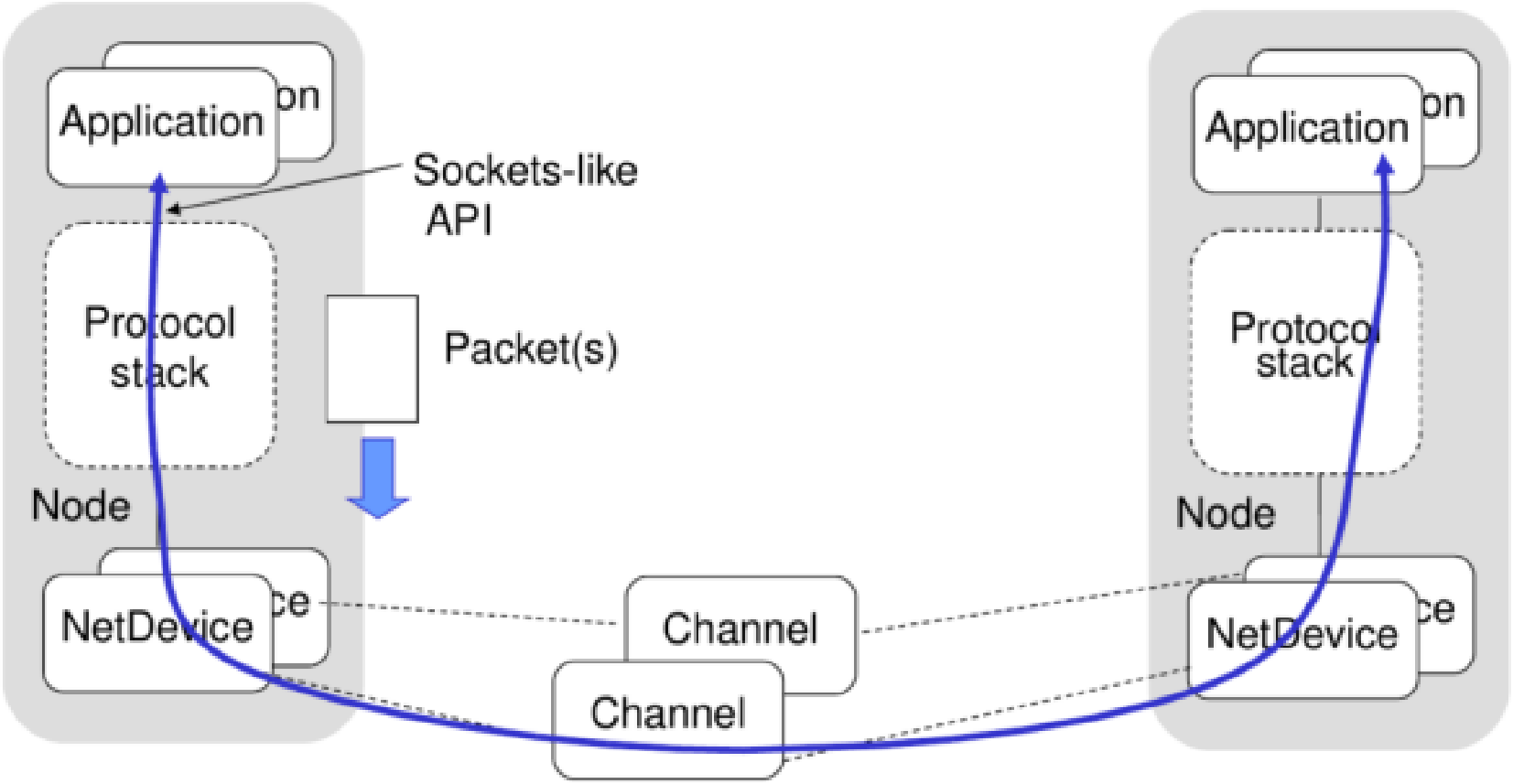
INTRODUCTION TO NS-3

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

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.)

Download ns-3

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

Addtional commands for remote access

- `$ ssh CSUserName@bolt.cs.ucr.edu`
- Lab 021: `$ ssh delta-xx`
- Lab 022: `$ ssh tango-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.

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);
```

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);
```

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;

- Some more example codes can be found here.
- NS-3 official tutorial can be found here.
- There are some Youtube lectures may help, like this.

Thank You