CS30096: Networks Lab

Sandip Chakraborty

NS-3 Basics

20 January, 2015

What is NS-3

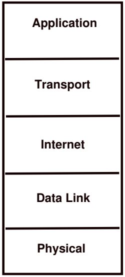
- Network Simulator (NS) - Why should we bother about?
- Discrete event simulator - what is a discrete event system? Can you give an example?

What is NS-3

- Network Simulator (NS) - Why should we bother about?
- Discrete event simulator - what is a discrete event system? Can you give an example?
- Computer network is a **discrete event system** - interrupt driven
  1. Receive a data packet at the network interface - interrupt 1
  2. Process the data packet - interrupt 2
  3. Generate the reply - interrupt 3
  4. Send the response packet - interrupt 4

Modeling Computer Network at Virtual Environment

- Protocol architecture - layering/modular
- Interrupt at every layer - every layer individually process the data packets/frames
- Physical layer - hardware/model driven (fading model, shadowing model)
- Data link layer to application layer - Software driven
- Network architecture/scenario - virtual embedding



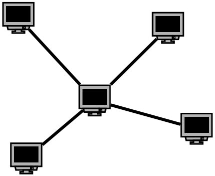
Application

Transport

Internet

Data Link

Physical



Basics of NS-3

- Download and Install - Go for the tutorial at NS-3 website `http://www.nsnam.org/` !!
- Modular architecture:
  - ns-allinone-3.21
    - bake
    - netanim-3.105
    - **ns-3.21**
    - pybindgen-0.17.0.876

NS-3 Hierarchy

- **ns-3.21**
  - bindings
  - build
  - doc
  - **examples**
  - ns3
  - **scratch**
  - **src**
  - utils
  - waf-tools

Execute a Script in NS-3

- Everything ns NS-3 is written in C++, Optional python binding is there. (Protocol stack - C++, Network scenario - C++/Python)
- Copy the script (.cc) file to the `scratch` folder.
- Execute the script using the `waf` tool:

  ```
   > ./waf --run scratch/<ns-3 program without extension>
  --command-template="%s <args>"
  ```

- Do not give the extension (.cc/.py) while executing the program using `waf`.

Understanding a NS-3 Program: Key abstractions

- `Node`
  - The abstraction of a basic computing device - think of a computer
  - The `Node` class provides methods for managing the representations of computing devices in simulations
- `Application`
  - Set of tasks - generates data traffic
  - This abstraction is represented in C++ by the class `Application`. The `Application` class provides methods for managing the representations of our version of user-level applications in simulations.

Understanding a NS-3 Program: Key abstractions

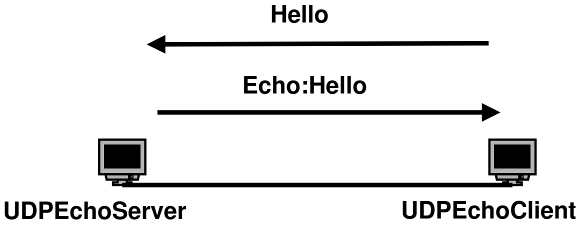
- Channel
  - Abstraction that connects the Nodes.
  - The Channel class provides methods for managing communication subnetwork objects and connecting nodes to them.
- Net Device
  - Think of a network interface card (NIC) - a peripheral to connect Nodes with the Channels
  - The net device abstraction is represented in C++ by the class NetDevice. The NetDevice class provides methods for managing connections to Node and Channel objects.

Understanding a NS-3 Program: Key abstractions

- `Topology Helpers`
  - In a real network, you will find host computers with added (or built-in) NICs. In NS-3 we would say that you will find Nodes with attached `NetDevices`.
  - In a large simulated network you will need to arrange many connections between Nodes, `NetDevices` and `Channels`.

NS-3 Program - First Look

- Open the file `/ns-allinone-3.21/ns-3.21/examples/tutorial/first.cc` and follow the tutorial !!



NS-3 Tracing: ASCII

- Generate the trace file in `.tr` file format.
- Enable Ascii tracing:

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

NS-3 Tracing: PCAP

- The acronym pcap (usually written in lower case) stands for packet capture, and is actually an API that includes the definition of a .pcap file format.
- You can visualize the packet details using Wireshark - We'll try this later!
- The pcap traces can be read out using a command line tool tcpdump.
- Enable pcap tracing:

```
pointToPoint.EnablePcapAll ("myfirst");
```

Happy Simulating!!