ns-3 Network Simulator

April 2018

Outline

1. Introduction
2. Architecture
3. Build
4. Examples

Overview

- ns-3 is a discrete-event network simulator, targeted primarily for research and educational use developed by UC Berkeley licensed under the GNU GPLv2 license.
- ns-3 software infrastructure encourages the development of simulation models which are sufficiently realistic to allow ns-3 to be used as a real-time network emulator, interconnected with the real world and which allows many existing real-world protocol implementations to be reused within ns-3.
- ns-3 simulation core supports research on both IP and non-IP based networks. Wireless/IP simulations which involve models for Wi-Fi, WiMAX, or LTE for layers 1 and 2 and a variety of static or dynamic routing protocols such as OLSR and AODV for IP-based applications.
- ns-3 is not compatible with ns-2. ns-3 is actively maintained; ns-2 is only lightly.

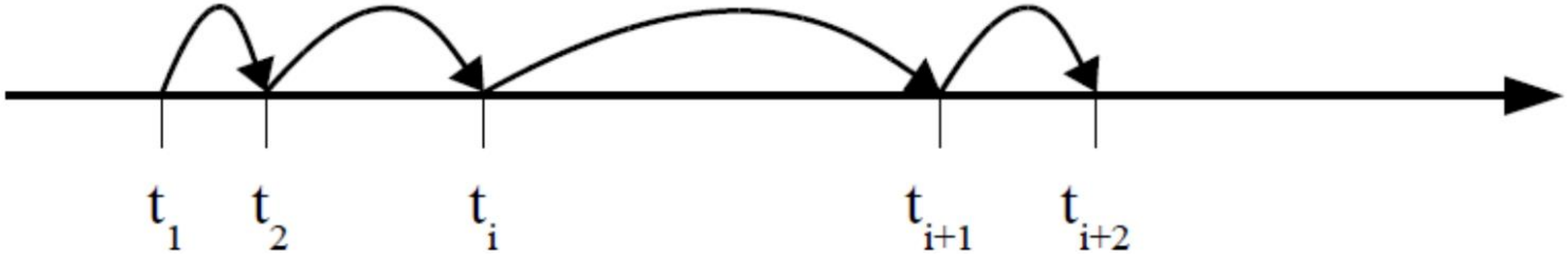
https://www.nsnam.org/overview/what-is-ns-3/

Documentation

- Official ns-3 documentation;
  - https://www.nsnam.org/documentation/
- Official ns-3 API documentation;
  - https://www.nsnam.org/docs/doxygen/index.html
- Official ns-3 tutorial;
  - https://www.nsnam.org/docs/tutorial/singlehtml/index.html
- Official ns-3 installation guide;
  - https://www.nsnam.org/wiki/Installation
- C++
  - The C++ Programming Language by Bjarne Stroustrup

Discrete Event Simulation

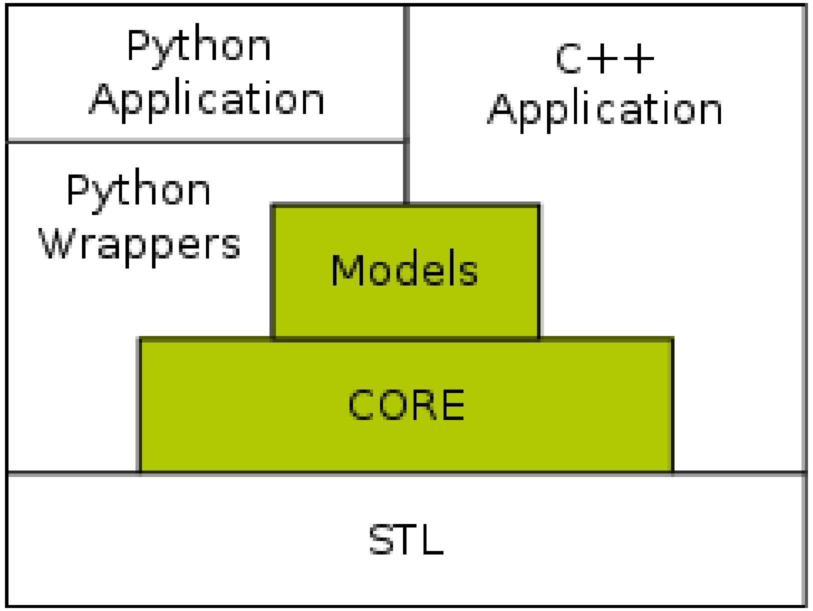
- The idea is to jump from one event to another.
- Events are recoded in a future event list (FEL).
- Each event notice is composed at least to data: time, type.
- Event routine or handler to process each event.
- Each event may change the system state and generate new event notices



http://users.jyu.fi/~arjuvi/opetus/tiea322/2015/Lecture2_Petrov.pdf

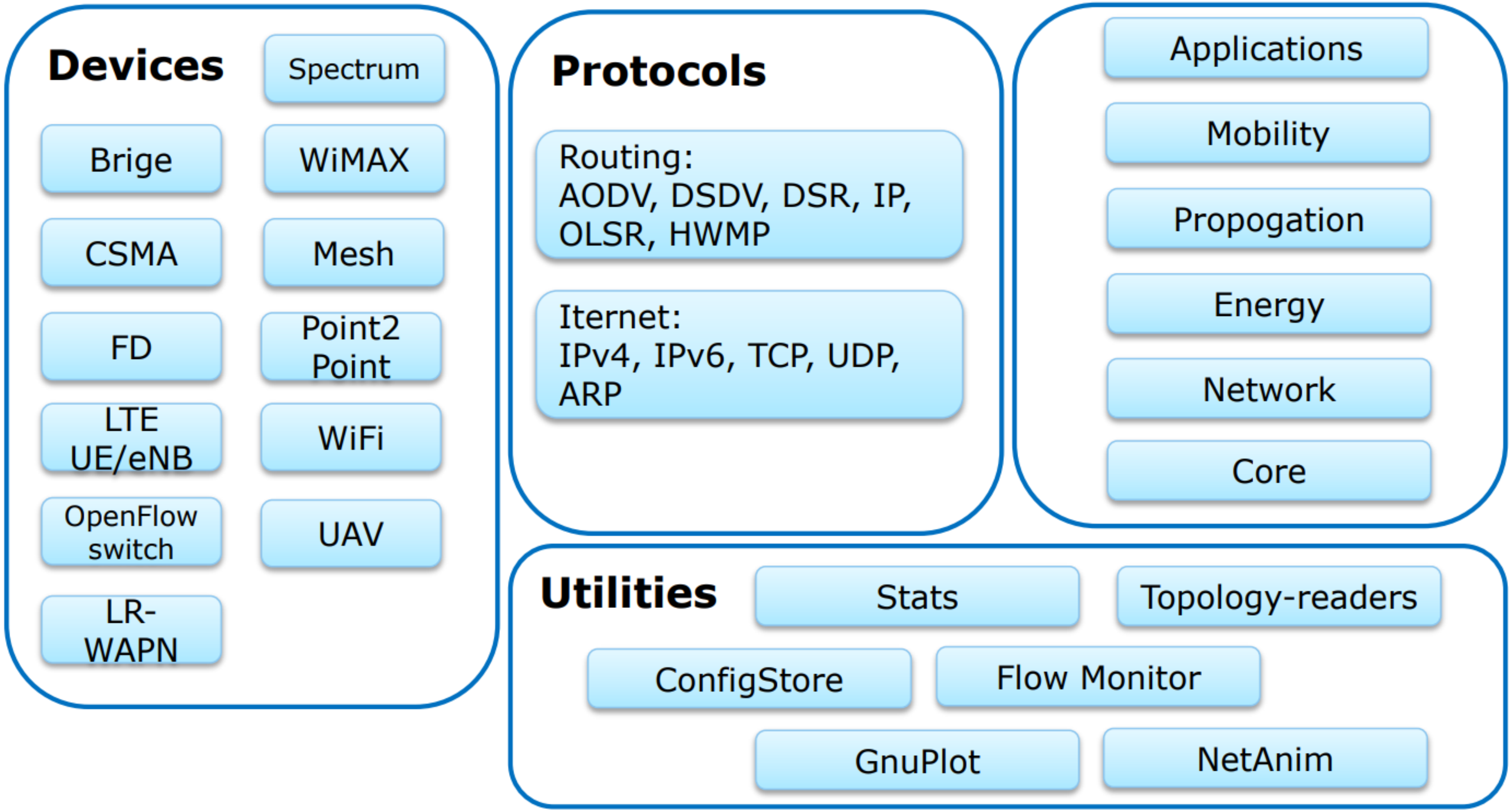
Architecture

- ns-3 is a C++ library which provides a set of network simulation models implemented as C++ objects and wrapped through Python.
- Users normally interact with this library by writing a C++ or a Python application which instantiates a set of simulation models to set up the simulation scenario of interest, enters the simulation main loop, and exits when the simulation is completed.
- ns-3 is primarily developed on GNU/Linux platforms.



https://www.nsnam.org/overview/key-technologies/

Modules



Devices

- Spectrum
- Brige
- WiMAX
- CSMA
- Mesh
- FD
- Point2 Point
- LTE UE/eNB
- WiFi
- OpenFlow switch
- UAV
- LR-WAPN

Protocols

- Routing: AODV, DSDV, DSR, IP, OLSR, HWMP
- Iternet: IPv4, IPv6, TCP, UDP, ARP

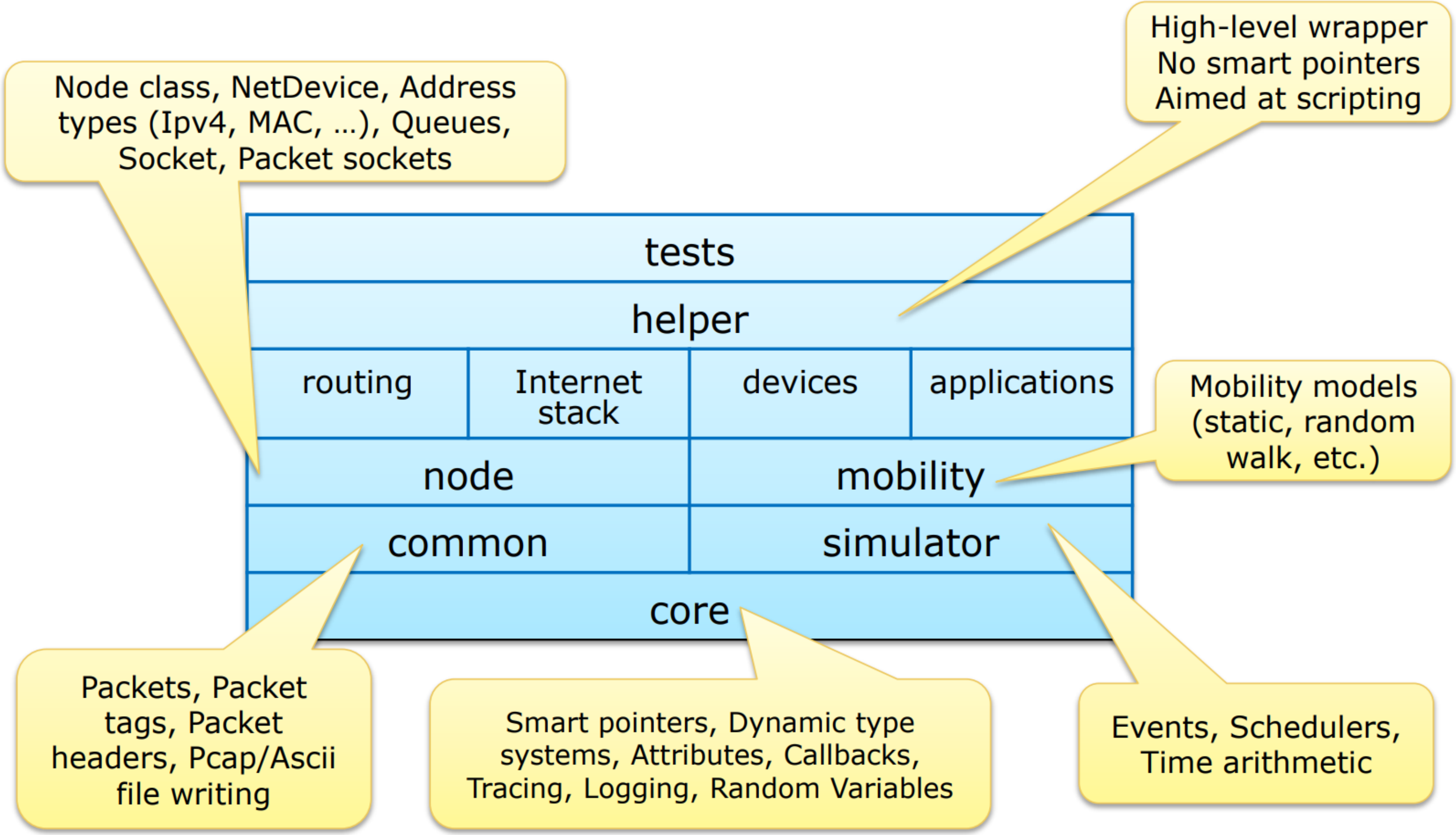
- Applications
- Mobility
- Propogation
- Energy
- Network
- Core

Utilities

- Stats
- Topology-readers
- ConfigStore
- Flow Monitor
- GnuPlot
- NetAnim

http://users.jyu.fi/~arjuvi/opetus/tiea322/2015/Lecture2_Petrov.pdf

Structure



High-level wrapper
No smart pointers
Aimed at scripting

Node class, NetDevice, Address types (Ipv4, MAC, ...), Queues, Socket, Packet sockets

Mobility models (static, random walk, etc.)

Packets, Packet tags, Packet headers, Pcap/Ascii file writing

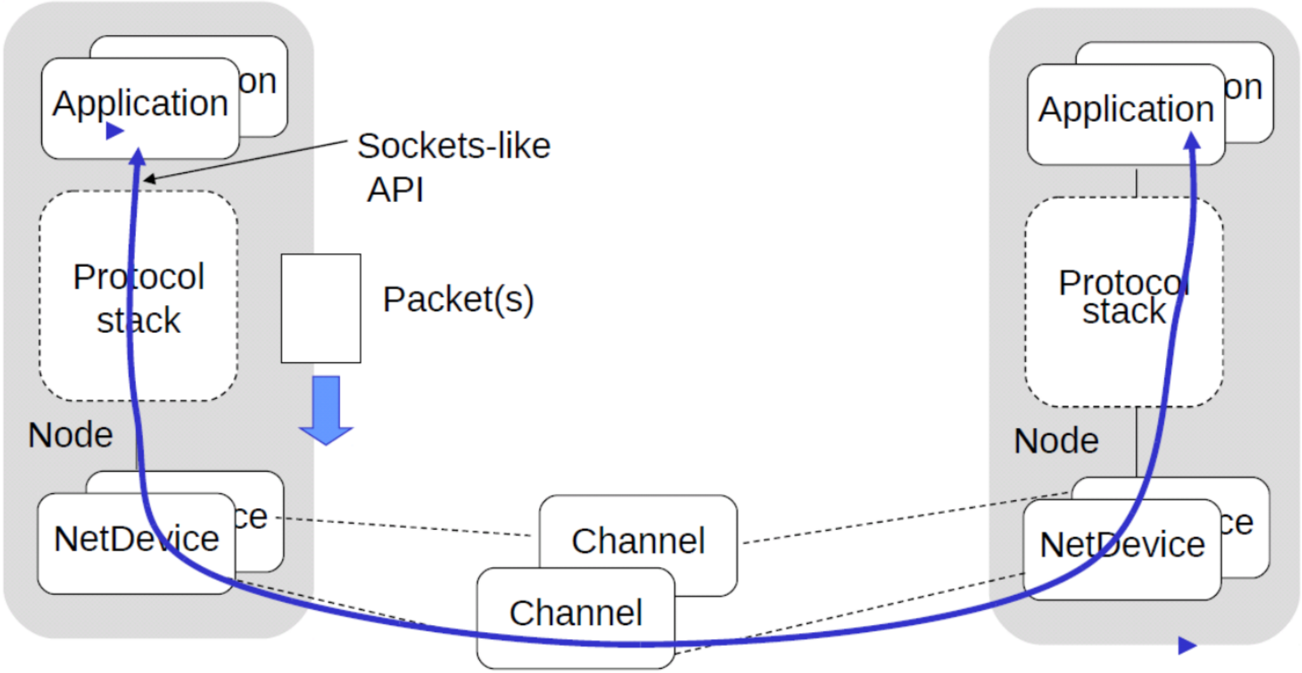
Smart pointers, Dynamic type systems, Attributes, Callbacks, Tracing, Logging, Random Variables

Events, Schedulers, Time arithmetic

http://users.jyu.fi/~arjuvi/opetus/tiea322/2015/Lecture2_Petrov.pdf

Architecture Elements

- Node;
  - Computer, access point, NIC.
- Net device;
  - Ethernet NIC, WifiNetDevice.
- Channel;
  - CSMA Channel, WifiChannel.
- Link;
  - Point-to-point, Bridge, Wifi, Mesh.
- Application (traffic generators);
  - On-Off: Generates streams, alternating on-and-off periods, either UDP or TCP.
  - Packet sink: receives packets or TCP connections.
  - Ping6, v4ping: send ICMP ECHO request.
  - Udp-client/server: sends UDP packet with sequence number.
  - Udp-echo: sends UDP packet, no sequence number.



https://www.cpe.ku.ac.th/~anan/myhomepage/wp-content/uploads/2013/06/2013-ns3-part1-introduction.pdf

Development Environment

- Ubuntu 16.04 is suggested;
  - https://www.ubuntu.com/download/desktop
- Virtualbox is suggested if Linux installation is not possible;
  - https://www.virtualbox.org/wiki/Downloads
  - At least 2 GB memory.
  - At least 20 GB hard drive.
  - At least 2 CPUs.
- After successfully obtaining environment, install following libraries and tools on Ubuntu machine;

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get install gcc g++ python python-dev qt5-default python-pygraphviz python-kiwi python-
pygoocanvas libgoocanvas-dev ipython openmpi-bin openmpi-common openmpi-doc libopenmpi-
dev autoconf cvs bzr unrar gdb valgrind uncrustify python-sphinx dia flex bison libfl-dev tcpdump sqlite sqlite3
libsqlite3-dev libxml2 libxml2-dev doxygen graphviz imagemagick texlive texlive-extra-utils texlive-latex-extra
texlive-font-utils texlive-lang-portuguese dvipng
```

https://www.nsnam.org/wiki/Installation

Build

- Download and extract latest ns-allinone release tarball.
  - https://www.nsnam.org/release/
- Build ns-3;

```
cd ns-allinone-3.*
./build.py --enable-examples --enable-tests
```

- Build ns-3 application;

```
cd ns-*
./waf
```

- Test ns-3; (optional)

```
./test.py -c core
```

https://www.nsnam.org/docs/tutorial/html/getting-started.html

Run Example

- There are several examples under examples folder. To run an example, copy it to scratch folder.

```
cp examples/tutorial/first.cc scratch/
./waf --run scratch/first

cp examples/tutorial/second.cc scratch/
./waf --run scratch/second

cp examples/traffic-control/traffic-control.cc scratch/
./waf --run scratch/traffic-control
```

- To enable log messages:
  - NS_LOG="<log-component>=<option>|<option>...:<log-component>..."

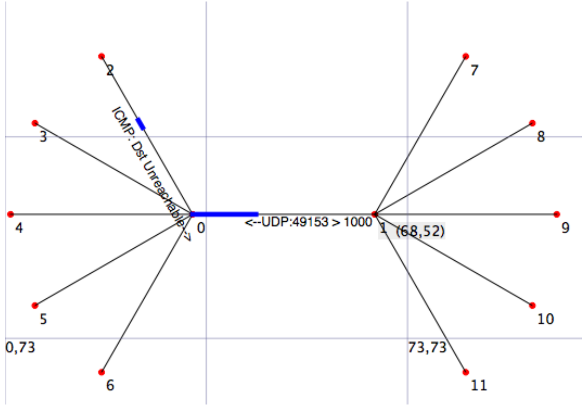
```
cp examples/tutorial/first.cc scratch/
NS_LOG=="star=all" ./waf --run scratch/star
```

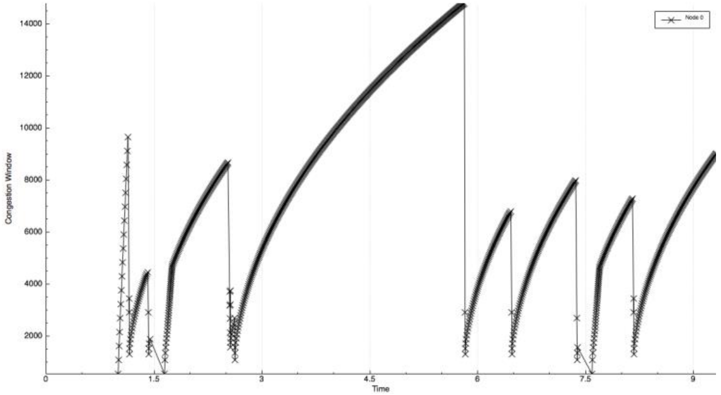
https://www.nsnam.org/docs/manual/html/logging.html

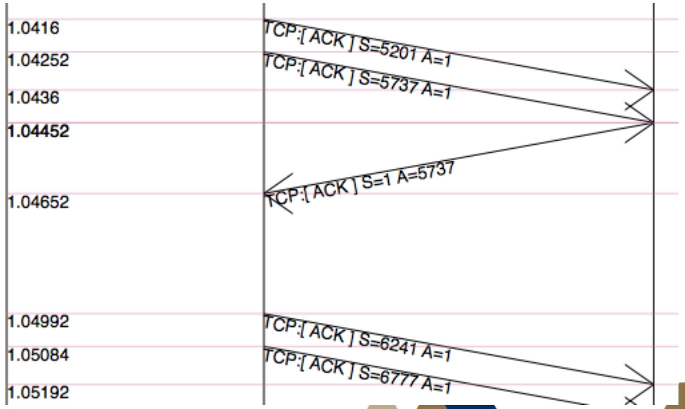
NetAnim

- NetAnim is an offline network animator tool which ships along with the ns-allinone-3.xx package.
  - https://www.nsnam.org/wiki/NetAnim_3.108
  - https://www.youtube.com/watch?v=-9a6dVbOds4
- It can animate the ns-3 network simulation using an XML trace file that is generated as an output during simulation.
- Build NetAnim;

```
cd netanim
make clean
qmake NetAnim.pro
make
```







https://www.nsnam.org/wiki/NetAnim_3.108

NetAnim

- Run matrix topology example to create ns-3 XML trace.

```
cp examples/matrix-topology/matrix-topology.cc scratch/
./waf --run scratch/matrix-topology

cd netanim
./NetAnim
```

- Netanim;
  - Open "*.anim.xml" in NetAnim.
  - Run simulation.
  - Observe stats and packets.
- Pcap traces;
  - Enable "p2p.EnablePcapAll" on matrix-topology.cc.
  - Open pcap files with Wireshark to investigate received and transmitted packages per node.

General Structure of a Script

- Handle command line arguments.
- Set default attribute values.
- Create nodes.
- Configure physical and MAC layers.
- Enable tracing.
- Set up network stack, routing and addresses.
- Configure and install applications.
- Set up initial positions and mobility.
- Connect trace sources and sinks.
- Schedule user-defined events and start simulation.

http://www.uio.no/studier/emner/matnat/ifi/INF5090/v11/undervisningsmateriale/INF5090-NS-3-Tutorial-2011-Oslo-slides.pdf

Common Script Elements

- NodeContainer
  - Provides a convenient way to create, manage and access any Node objects.
  - Stores pointer to those objects internally.
  - The nodes as they stand in the script do nothing.
    - To construct a topology is to connect our nodes together into a network.
- PointToPointHelper
  - To construct a point to point link.
  - Perform the low-level work required to put the link together.
  - Typically these two things NetDevice and Channel are intimately tied together (cannot expect to interchange).
    - Ethernet devices and CSMA channels.
    - Wifi devices and wireless channels
- FlowMonitor
  - To provide a flexible system to measure the performance of network protocols.
  - Detect all flows passing through network.
  - Stores metrics for analysis such as throughput, duration, delays, packet sizes, packet loss ratios

https://www.cpe.ku.ac.th/~anan/myhomepage/wp-content/uploads/2012/01/ns3-part3-wireline.pdf

Thank you for your time.

Any questions?

References

1) https://www.nsnam.org/documentation/
2) https://www.nsnam.org/overview/key-technologies/
3) https://www.nsnam.org/wiki/Installation
4) https://www.nsnam.org/docs/tutorial/html/getting-started.html
5) https://www.nsnam.org/docs/manual/html/logging.html
6) https://www.nsnam.org/wiki/NetAnim_3.108
7) http://users.jyu.fi/~arjuvi/opetus/tiea322/2015/Lecture2_Petrov.pdf
8) http://www.uio.no/studier/emner/matnat/ifi/INF5090/v11/undervisningsmateriale/INF5090-NS-3-Tutorial-2011-Oslo-slides.pdf
9) https://www.cpe.ku.ac.th/~anan/myhomepage/wp-content/uploads/2013/06/2013-ns3-part1-introduction.pdf
10) http://personal.ee.surrey.ac.uk/Personal/K.Katsaros/ns-3-workshop-part1.html
11) http://personal.ee.surrey.ac.uk/Personal/K.Katsaros/media/NS-3-Presentation-2013.pdf
12) https://www.cpe.ku.ac.th/~anan/myhomepage/wp-content/uploads/2012/01/ns3-part3-wireline.pdf
13) https://www.cpe.ku.ac.th/~anan/myhomepage/wp-content/uploads/2014/09/ns3-lecture10-flowmon.pdf