Nephel3: A Cloud-Based ns-3 Code Generation Service

Adrian E. Conway
4-Layer Simulations LLC
Weston, MA 02493, USA
aec@ieee.org

EXTENDED ABSTRACT

A cloud-based web service Nephel3 is demonstrated for users to generate ns-3 simulation code for network models. The service takes as input a user-provided network model file and delivers as output an ns-3 'main' program. An accompanying generic user-based tool 4Layer Modeler provides a graphical interface for constructing and editing four layer Internet system models. The Nephel3 service and 4Layer Modeler tool facilitate the production of ns-3 code for complex models or large topologies. It can also help new users or non-experts to generate ns-3 code.

1. INTRODUCTION

The ns-3 network simulator is a widely used open-source tool in the simulation of Internet systems. The software comes with supporting documentation including a reference manual, model library information, Doxygen API information, and example ns-3 programs. For some new users or non-experts, however, it can require much effort and be time consuming to become proficient with using ns-3. The construction and editing of more complex or larger network simulations can also present practical challenges in itself. A user can also face additional difficulties when multiple different ns-3 modules or capabilities are to be used simultaneously. To address such challenges, graphical user interfaces (GUI) and automation tools for ns-3 are desirable.

Several works include hitherto developed automated means for creating ns-3 programs. Hallagan *et al*. [2] describe the Simulation Experiment Automation Framework which provides an XML-based network model file to a translator that generates ns-3 script. Addie [1] describes NetML which includes a tool to convert a NetML document into a ns-3 program. Saino *et al*. [3] describe the Fast Network Simulation Setup toolchain library that presents an XML file to adapters to form ns-2 or ns-3 simulation programs. The ns-3 topology generator project [4] and NetSH [5] provide GUI for generating ns-3 programs.

In this demo/poster, we present *Nephel3*, a cloud-based web service for machine-generating ns-3 simulation programs for user-provided network models. An accompanying graphical tool *4Layer Modeler* is also presented to construct generic Internet system models and generate corresponding network model files.

2. NEPHEL3 SERVICE

The Nephel3 service takes as input a user-provided network model file and delivers as output to a user a ns-3 'main' program, as shown in Fig. 1. The 4Layer Modeler tool provides a user with a graphical interface for building and editing generic network models, choosing options, specifying parameters, and creating corresponding network model (*.flm) files.

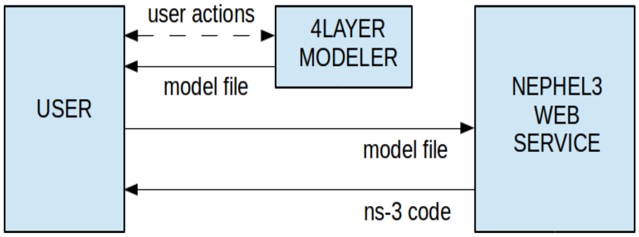
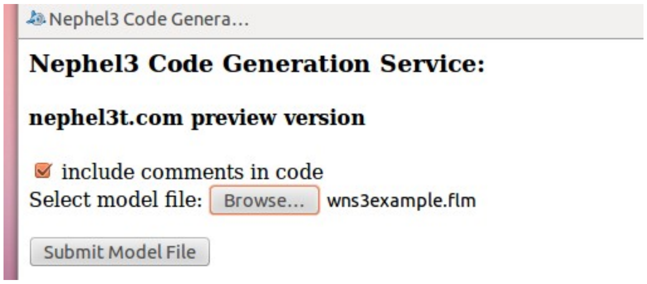


Figure 1: Code generation process.

At the time of this demo, the Nephel3 web service is provided at nephel3t.com, where 't' denotes a temporary 'preview' service. A 'release' service is to be provided at nephel3.com. The service is implemented in terms of Java servlets that run on the Google App Engine^{TM} platform. The servlet code written for generating ns-3 code relies upon coding and API information garnered from ns-3 documentation. Fig. 2 shows the nephel3t service in a web-browser, where a user can select a local *.flm model file and then submit it to the code generation service. The generated ns-3 code is returned to the user as text on a web page. Machine-generated comment lines can be included if desired.



Nephel3 Code Generation Service:

nephel3t.com preview version

☑ include comments in code
Select model file: wns3example.flm

Figure 2: Nephel3 web-browser service interface.

3. 4LAYER MODELER TOOL

The 4Layer Modeler tool is a stand-alone graphical interpreted Java application intended for desktop/laptop use. At the time of this demo, it is available for download at 4-layer.com. The main interface, shown in Fig. 3, includes tabs for providing a Network View, an Application View, a Mobility View, Tracing options, and Simulation Parameters. A drop-down Network Model menu enables one to open, close, save, save as, delete, and export/import *.flm model files.

In the Network View, various network elements may be added and interconnected graphically to form a network topology. These include nodes (denoted as routers, clients, servers), 'wired' links, LTE network elements (SGW/PGW, eNodeB, UE), CSMA-type networks and their devices (*i.e.*, interfaces), Wifi Infrastructure networks and their STA and AP devices, Wifi Adhoc networks and their STA devices, and Wimax networks and their BS and SS

Workshop on ns-3 (*WNS3'16*), June 15-16, 2016, Seattle, WA, USA.

devices. Multiple 'parallel' wired links can be added between pairs of nodes.

Network element options and parameters are editable via various dialogues. As one example thereof, Fig. 4 shows the dialogue for editing an LTE network instantiation, including LTE options, path loss, frequency reuse, handover, fading, scheduling, and eNodeB transmission parameters.

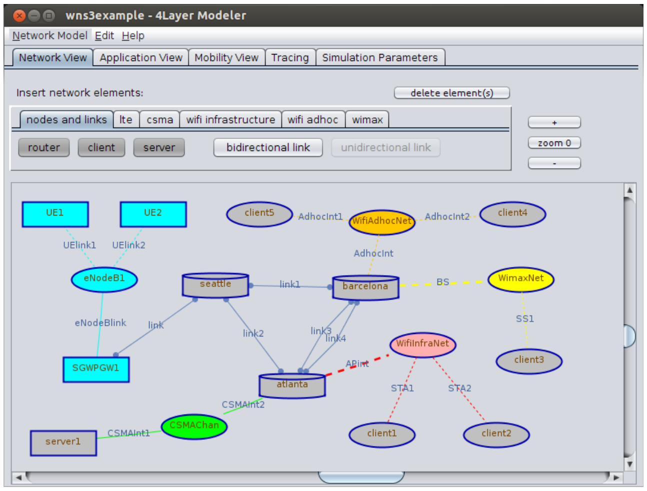
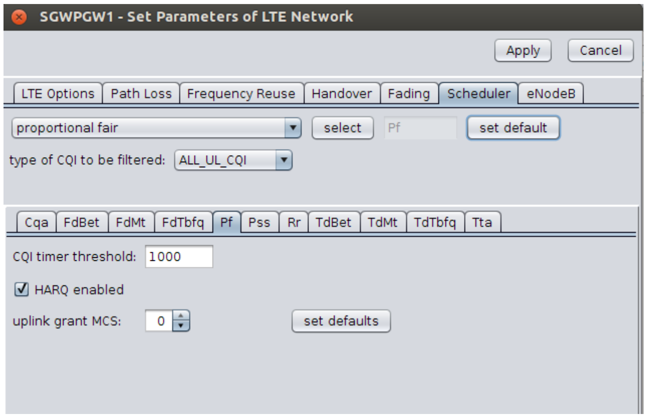


Figure 3: Main interface with Network View showing.



proportional fair

type of CQI to be filtered: ALL_UL_CQI

CQI timer threshold: 1000

☑ HARQ enabled

uplink grant MCS: 0

Figure 4: Dialogue for editing LTE options and parameters.

In the Application View of Fig. 5, end-to-end application-layer traffic can be added between nodes and UE. Application types include on/off source, bulk send, UDP client server, ping v4, and UDP echo. Multiple 'parallel' applications can be added between pairs of end nodes and UE. The parameters of instantiated applications are editable via dialogues.

In the Mobility View of Fig. 6, nodes are connected by solid black undirected lines if they are connected by 'wired' (*i.e.*, non-wireless) links in the Network View. Mobility models can be added to nodes and UE, including constant position, constant velocity, constant acceleration, 2D random direction, 2D random walk, and 2D Gauss Markov. Mobility references can also be pushed to nodes, as shown in Fig. 6 by directed grey lines. The parameters of instantiated mobility models are editable via dialogues.

In the Tracing tab, pcap and ascii tracing can be selected for interfaces (*i.e.*, devices). In the Simulation Parameters tab, parameters such as total simulation time can be specified.

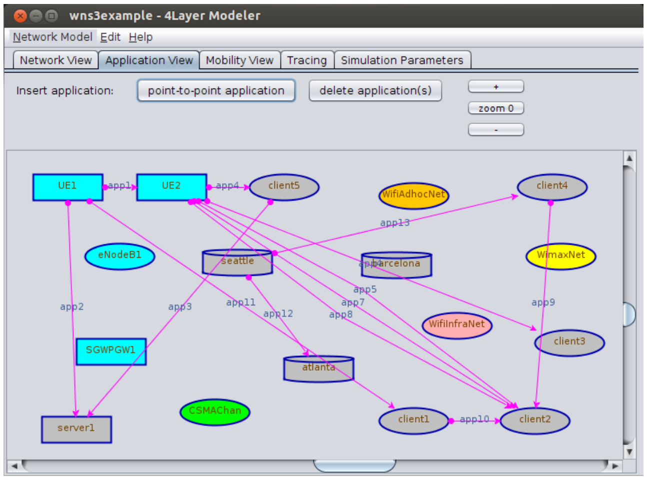


Figure 5: Application View showing end-to-end applications.

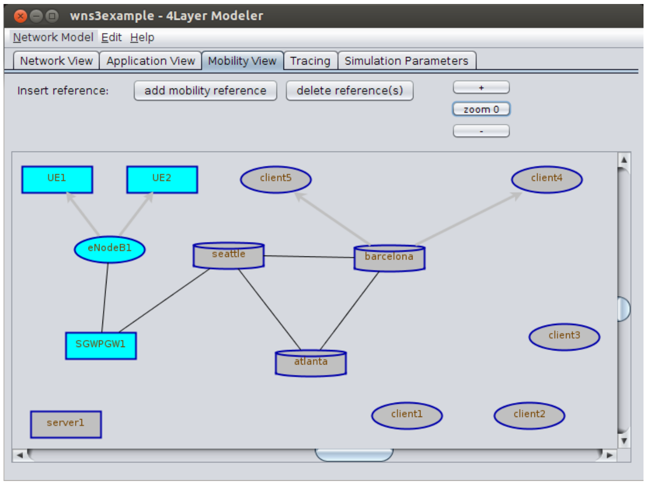


Figure 6: Mobility View showing mobility references.

4. REFERENCES

[1] Addie, R.G., Netml System for Analysis and Design of Networks (tutorial summary and slides), in *26th International Teletraffic Congress*, Karlskrona, Sweden, Sept 9-11, 2014.

[2] Hallagan, A., Ward, B., and Felipe Perrone, L., An Experiment Automation Framework for ns-3, in *3rd International Conference on Simulation Tools and Techniques* (SIMUTools), Torremolinos, Spain, March 15-19, 2010.

[3] Saino, L., Cocora, C., and Pavlou, G., A Toolchain for Simplifying Network Simulation Setup, in *6th International Conference on Simulation Tools and Techniques* (SIMUTools), Cannes, France, March 5-7, 2013.

[4] nsnam.org/wiki/Topology_Generator

[5] http://mailman.isi.edu/pipermail/ns-developers/2014-February/011742.html, [Ns-developers] NetSH for ns-3