An Experiment Automation Framework for ns-3

(Poster Abstract)

Andrew Hallagan, Bryan Ward, and L. Felipe Perrone
Department of Computer Science
Bucknell University
Lewisburg, PA, U.S.A. 17837
{andrew.hallagan, bryan.ward, perrone}@bucknell.edu

ABSTRACT

Recent studies have indicated that tools that automate the execution of simulation experiments can serve to enhance both the usability of network simulators and the credibility of the studies developed with them. In this poster, we present the architecture for an experiment automation framework for the ns-3 network simulator. The architecture was designed with two specific goals in mind. First, it raises the level of abstraction of the ns-3 user interface to make the simulator easier to use in large scale experimental studies. Second, it provides functionalities to guide the user along known correct methodologies for modeling and simulation thereby increasing the confidence one can have in the correctness of the experimental results.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation support systems—*environments*; D.2.6 [**Software Engineering**]: Programming environments—*performance measures*

General Terms

Experimentation, Measurement, Performance

Keywords

Simulation tools, network simulation, best practices

1. INTRODUCTION

We present the architecture of a framework to manage and automate experiments with the ns-3 network simulator. The design of this framework is driven by two main goals: to provide enhanced usability for the simulator and to include mechanisms that will lead to credible experimental results.

This project is motivated by previous work on SWAN Tools[5], a web-based experiment automation framework, which allowed users to define experiments, launch their execution on a distributed network of machines, and explore results using a standard web browser. The lessons learned in this earlier experience prompted us to keep certain elements of the old architecture and to redesign other elements so that the new architecture would be more extensible and flexible. For example, SWAN Tools constrained users to work with a pre-defined model for the network protocol stack in the wireless node; we identified the need to allow users of the new framework to build their own protocol stacks. We also recognized the need for users to have fewer constraints in the description of experiments, which led us to construct two different user interfaces to the framework: a more rigid one for novices and a more flexible one for experienced users.

In order to meet these broad design objectives, our framework for ns-3 defines derivations of the eXtensible Markup Language (XML): one language for the description of the simulation *model* and another language for the description of the simulation *experiment* that will use that model. The type of interface to the system chosen by the user determines how these descriptions are generated. The interface for novices generates the XML-based language files based on user input provided via web browser interactions. The interface for advanced users requires the manual construction of description files using common text editors. Description files can be validated against formal specifications before they are used by the framework. We have constructed a proof-of-concept translator that takes in these XML-based descriptions and generates ns-3 scripts in C++ and Python code.

The process of running experiments using the generated ns-3 scripts is handled by an execution manager component within our framework. The design of this component includes a server built around the REpresentational State Tranfer (REST) model. Clients communicate with the server to fetch simulations to run, then they execute them and periodically report back to the server the results generated.

2. XML MODELING

The XML-based languages we define are used to describe the simulation model, the factors that can be varied in the model, and the lists of levels used in a simulation experiment. The general structure of the model is specified as a document written with our *model description* language. This document consists of a collection of sub-models, where each element has strict correspondence to an existing ns-3 model. Two additional documents express the user defined

SIMUTools 2010 March 15–19, Torremolinos, Malaga, Spain.

design of experiment. A first document, written in our *experiment description* language, contains lists of levels for every factor instantiated in the model description. It specifies how possible combinations of different levels for factors in the model might appear in an experiment. (The language allows factorial designs to be expressed.) A second document uses a *restriction description* language to indicate the combinatorial relations between subsets of model levels. This document effectively prunes down the experimental design space by requiring parameters to occur in tandem or in exclusion.

All the documents written in these languages are validated against one or more *RELAX NG* schemas. [3] After considering the use of XML Schema, the W3C standard for XML document validation, we opted for the more expressive RELAX NG due to its ability to describe and validate more complex properties and relationships between XML documents. We have found the RELAX NG compact syntax to be more natural and convenient in expressing the grammatical constructs we developed.

While our current goal is to build a framework for ns-3 specifically, we envision that we are taking a step toward the creation of technologies that will be portable to other simulation platforms. A long term goal is to incorporate ontologies for network simulation into this framework. By encoding our model and experiment description in XML, we are laying the groundwork to enable the future integration of our tools with ontologies for the network simulation domain. [2]

3. EXPERIMENT MANAGER

The XML files described above are used as inputs to an *experiment execution manager* which runs as a web application. This execution manager uses experiment description files to generate the points in the experiment design space that correspond to simulation runs. An HTTP based service gives the user the ability to have some control over the execution of the experiment from a web browser interface. The use of a standard protocol for this service allows the client processes which execute simulations to interact with the framework using well-established and well-known libraries. The decoupling that comes from the use of HTTP allows programmers to develop clients in any language or platform.

As the Akaroa 2 project did for the ns-2 simulator, this framework implements the paradigm of Multiple Replications in Parallel (MRIP) for ns-3. [4] Independent simulations are dispatched to execute concurrently across different client computers and report results back to the experiment manager via a REST interface. Clients periodically communicate with the experiment manager to evaluate if enough samples of estimated metrics have been collected to produce adequate confidence intervals. When clients report results to the experiment manager, they receive responses indicating whether they should continue to execute or terminate the simulation. These interactions allow clients to have automatic determination of the simulation run length (similarly to what is done in Akaroa 2) and allow the server to dispatch new simulations to be run.

4. USER INTERFACES

Less experienced users find a friendly interface to the framework via a standard web browser. This feature follows directly from the fact that the framework is designed as a web application. This interface allows the user to create custom experimental designs and use pre-built simulation models without the need to write code in our XML-based description languages. The web interface also allows one to view raw output results in tabular format in the browser and to use facilities to create custom plots similar to those in ANSWER [1] and SWAN Tools [5]. The processing of simulation output data is integrated with the plotting facilities so that best practices in statistics are always used.

More experienced users can access command-line interface tools to create custom models and experimental descriptions in our XML-based languages, to validate them against schemas, and to manage the execution of simulation experiments. These command-line tools are being built around an open REST API that allows access to the services provided by the web application framework.

5. RELATED WORK

Several of the features of this framework have been implemented as individual components of other projects, for different underlying simulators. This framework builds on the ideas and designs presented in SWAN Tools [5], ANSWER [1], and Akaroa 2 [4]. Our contributions with this project include the development of a design that unifies these components into one single framework, and the provision of interfaces differentiated according to the level of user expertise. We expect that the guidance that our framework provides in the construction of models and experiments, in their execution, and in output data processing will help users to produce more credible results.

6. REFERENCES

[1] Andreozzi, M. M., Stea, G., and Vallati, C. A framework for large-scale simulations and output result analysis with ns-2. In *Proc. of the 2nd Intl. Conf. on Simulation Tools and Techniques (SIMUTools '09)* (2009), pp. 1–7.

[2] Miller, J. A., Baramidze, G. T., Sheth, A. P., and Fishwick, P. A. Investigating ontologies for simulation modeling. In *Proc. of the 37th Annual Symposium on Simulation (ANSS 04)* (Washington, DC, USA, 2004), IEEE Computer Society, p. 55.

[3] Murata, M., Lee, D., Mani, M., and Kawaguchi, K. Taxonomy of XML schema languages using formal language theory. *ACM Trans. Internet Tech. 5*, 4 (2005), 660–704.

[4] Pawlikowski, K. Akaroa2: Exploiting network computing by distributing stochastic simulation. In *Proc. of the 1999 European Simulation Multiconference* (Warsaw, Poland, 1999), pp. 175–181.

[5] Perrone, L. F., Kenna, C. J., and Ward, B. C. Enhancing the credibility of wireless network simulations with experiment automation. In *Proc. of the 2008 IEEE Intl. Conf. on Wireless & Mobile Computing, Networking and Communications (WiMob '08)* (2008), pp. 631–637.