

Providing a Graphical User Interface for *ns-2* with Visual Network Simulator

Poster Abstract

Eduardo M.D. Marques¹, Ricardo A.S.A. Plácido², Paulo N.M. Sampaio³

Laboratory for Usage-centered Software Engineering (LabUse)^{1,3}

Centro de Ciências Matemáticas (CCM)^{1,3}

Mathematics and Engineering Department (DME)

University of Madeira (UMa)

Madeira, Portugal

Phone: (+351) 291705288, Fax: (+351) 291705199

emarques@uma.pt, ricky.placido@gmail.com, psampaio@uma.pt

ABSTRACT

The growth of the number of hosts connected to the Internet and of new applications increased the importance of research and development of Quality of Service (QoS)-based networks. In order to optimize the proposal and the validation of new QoS architectures and technologies, simulation seems to be the best solution when considering the deployment efforts, cost and scalability. Unfortunately, the existing platforms and tools for simulating QoS-based Networks either do not support all the architectures and technologies or do not fully support their characteristics. Basically, a simulation life-cycle can be divided into three phases: specification of the scenario, simulation of the events and analysis of the results of the simulation. This paper presents a platform for the specification of networks scenarios supporting QoS and multicast, which can be simulated using *ns-2*. This platform provides the author with the possibility of covering all the three phases of the simulation life-cycle on-the-fly.

Keywords

DiffServ, multicast, network simulator 2, Quality of Service, simulation.

1. INTRODUCTION

The Network Simulator version 2 (*ns-2*) [1] is a general purpose network simulation platform with a large community of users and is an open source project. *ns-2* uses a split programming model where two languages, C++ and Tcl scripting, are used. C++ is preferentially used to develop the simulation kernel, a “core set of high-performance simulation primitives”. The definition, configuration and control of the simulation are done with Tcl scripts [2].

The specification of network scenarios with *ns-2* can be done either by the utilization of an appropriate graphical user interface (GUI) or by the direct edition of the respective source code file. The tools NSG2 [3], nsBench [4] and NSBM [5] are examples of GUI's to *ns-2*. However, these tools are more appropriate for the specification of simple scenarios. Most of the existing tools reveal to be limited or not intuitive when we need to specify more complex network scenarios. Needless to say that the direct edition of Tcl scripts is obviously too complex for the specification of such scenarios. The *ns-2* GUI has been pointed out as a reason to limit a broader use of *ns-2*, being a difficult tool to use by less experienced

users. Shilling [6] refers the “lack of documentation and the usage of two programming languages” as some factors that lead users to long learning periods.

The main contribution of this work is to present the main aspects of an *ns-2* compliant platform, called Visual Network Simulator (VNS), developed to provide a user-friendly interface and a more intuitive environment for the specification (authoring) of network scenarios, supporting advanced aspects such as QoS and multicasting.

2. VISUAL NETWORK SIMULATOR

The Visual Network Simulator (VNS) tool, centered on the NSG2 tool [3] capabilities, was developed aiming to facilitate the creation of network simulation scenarios to be executed over *ns-2* [7]. Two important innovations of VNS, compared to other similar tools are: (1) the support to Differentiated Services (DiffServ) [8] scenarios, and, (2) it provide a simple and intuitive set of icons to represent the components of a network. Examples of these icons are the hosts, the routers and an Internet cloud, each with a different symbol. Some others important functionalities of VNS were also taken into account, such as:

- Adding and configuration of links, agents and traffic sources;
- Modeling network scenarios with support to multicast;
- Selection of a dynamic routing protocol;
- Definition of the simulation output as an animation and/or graphics;
- Edition of the Tcl script generated, and;
- Saving the defined simulation scenario.

Once all the network topology, traffic sources and events are all specified with VNS, this tool can generate the Tcl script for the simulation of the scenario. The user can also edit the Tcl script in order to modify or introduce values in the simulation. These modifications can fulfill some VNS limitations concerning some non-foreseen parameters. In VNS we can also specify if the Animation and traces files are generated during the execution of the simulation. This data can be visualized in the *Network Animator* NAM [9] and xGraph [10] tools.

The VNS tool was built in a way to provide a smooth integration with *ns-2*, NAM and xGraph. This set of tools, each one with a particular task, are the components of the VNS platform.

2.1 Scenario creation

The creation (or edition) of a network simulation scenario is carried out using a set of objects available in the tool bar. Some objects, such as the node, link, agent and the application, are common to *ns-2* users or some other similar network simulators. Other objects, such as the router and the Internet, were created to simplify the modeling of the scenario and to make it more intuitive, thus enhancing the simulation capabilities of the tool.

Figure 1 presents a VNS screen capture with an example scenario. We can observe the toolbar with all the objects that can be applied on a scenario on the upper side of this figure. To start creating a scenario, the user has to define if the project is DiffServ-based or without QoS. The main difference between these two options is the presence of the object router in the case of a scenario with QoS (DiffServ). The use of the remaining objects is similar in both cases.

The options for adding and editing the most used elements of the simulation scenario are explained as follows:

- Hand – this option allows, with the zoom slider in the edge of the screen, to adjust the visualization of the scenario;
- Scene – provides quick access to a set of predefined scenarios from a folder;
- Node – enabling this option, the nodes of the scenario can be added or edited;
- Router – is a special node that can only be defined in the case of QoS scenarios. It can represent an edge or core router;
- Link – it is defined in the same way as in *ns-2*, except in the case of a DiffServ scenario where they present different configuration parameters;
- Agent – defines the transport protocol (TCP or UDP) used by an application;
- Application – defines the traffic that will be generated or which application is going to be simulated;
- The *Internet* object is an interesting abstraction implemented on VNS. This object was proposed to simplify the simulation of an Internet application;
- The *Parameters* option provides the configuration of some global simulation possibilities, such as: simulation duration, enabling of NAM traces, and, to present, or not, some results in the end of the simulation, and;
- The option *Generate TCL* initiates the translation from the graphical scenario created to the script in *ns-2* simulation description language.

The use of the left and right buttons of the mouse allows to add, setup and delete elements of the scenario. For each different object the user will find a similar configuration process. To add and move an object the left button of the mouse should be used. In the case of the access to more detailed options of an object, the right button of the mouse should be pressed on.

3. CONCLUSIONS AND FUTURE WORKS

The main contribution of VNS is to provide a rich authoring tool integrated with a set of important tools, composing a complete and intuitive authoring and simulation environment.



Figure 1. VNS interface with a DiffServ enabled scenario

There are available some other tools in the literature to create simulation scenarios ([3], [4] and [5]), however, most of them present some limitations. VNS aggregated the major qualities of several existing tools, such as QoS support and saving created scenarios. Further on, some new characteristics were also implemented, such as the use of more rich graphic objects to identify the network components.

As for future works, we plan to improve the VNS tool with the implementation of other QoS architectures and mechanisms, such as Integrated Services, Multi Protocol Label Switch (MPLS) and traffic engineering functionalities. We also intend to improve the scheduling of events for each object and to execute a usability validation of the tool by its users.

REFERENCES

- [1] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heiderman, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. "Advances in Network Simulation". IEEE Computer, 33 (5), pp. 59-67. May 2000.
- [2] E. Bajaj, L. Breslau, D. Estrin, F. Kevin, F. Sally, H. Padma, H. Mark, H. Ahmed and Y. Haobo, "Improving Simulation for Network Research". University of Southern California, 1999.
- [3] P. Wu, "NS2 Scenarios Generator 2", available from <http://sites.google.com/site/pengjungwu/nsg>, 2008.
- [4] E. Al-Shaer, "nsBench – Graphical User Interface for Network Simulator", <http://www.mnlab.cs.depaul.edu/projects/nsbench/>, 2005.
- [5] M. Welzl, M. Ali, and S. Hessler: "Network Simulation By Mouse (NSBM): A GUI Approach for Teaching Computer Networks with the *ns-2* Simulator", Proceedings of the International Conference on Interactive Computer Aided Learning (ICL 2006), 26-28 September 2006, Villach, Austria.
- [6] B. Shilling, "Qualitative Comparison of Network Simulation Tools", Institute of Parallel and Distributed Systems (IPVS), University of Stuttgart, 2005.
- [7] R. Plácido, "Simulacao de Redes Multicasting e Garantias de Qualidade de Servico", M.Sc. dissertation, University of Madeira, July 2008.
- [8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [9] D. Estrin, M. Handley, J. Heidemann, S. McCanne, Y. Xu, and H. Yu, "Network Visualization with the VINT network animator nam". Tech. Rep. 99-703, University of Southern California, Mar. 1999.
- [10] Xgraph - <http://www.isi.edu/nsnam/xgraph/>