NS2IT: Simplification of computer network's simulation

Martin NAGY, Dávid OROS, Roman PANENKA, Martin PIRHÁČ, Hana SEVERÍNOVÁ, Martin SVETLÍK*

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
`timovy-projekt-pss14@googlegroups.com`

Abstract. Nowadays, Internet applications generate more and more communication traffic that needs to be wisely handled by computer networks. Therefore there is a big demand on communication protocols to have less payload and bigger efficiency. In this paper, we briefly explain our solution of enhanced simulation and visualization of network protocols based on ns-2 simulator, called NS2IT. Our goal is to simplify the simulation's process, data visualization and measurements' comparison. Proposed prototype teaches students about characteristics of given protocols and inform developers about protocols' performance in specific network situations. All these information are presented by rich web interface.

1 Introduction

Nowadays, the simulation of computer networks is becoming a rather difficult and complex task to accomplish. Because of the large amount of accessible technologies, billions of possible communicating couples, the large scale and complexity of the networks, the computer network simulations tend to produce large amount of data, which is almost impossible to interpret correctly in a reasonable amount of time. Similarly, a programmer has to be rather experienced when he wants to create a fully operable input file for the most used simulation tools. In this paper, we introduce the reader to the area of computer networks simulation, to the means by which various types of simulation can be achieved and how can we correctly and efficiently interpret the large amount of simulation output data with proposed solution.

Our goal is to develop a simplified process of network simulation in its whole lifecycle, from designing a network, simulating it and finally, correctly interpreting the output of the simulation with a focus on those aspects and values of the simulated network that are in the user's interest. Our project is aimed at one of the most known computer network simulators called ns-2. We briefly explain the characteristics and the comparison of ns-2 and ns-3, along with the types of computer networks that can be simulated with these simulators. We also introduce our solution that simplifies the usage of the ns-2 simulator by adding an easy-to-use input and output interface to this simulator. We also offer a concept of simulating networks over the Internet as a server side

* Master degree study programme in field: Computer Engineering
Supervisor: Peter Magula, Institute of Computer Systems and Networks, Faculty of Informatics and Information Technologies STU in Bratislava

application with a simple, yet powerful web interface that provides a high portability and accessibility to our solution with only minimal scripting or programming knowledge requirements on the user.

Our solution is a very agreeable way to bring the network simulation capabilities of the complex and powerful ns-2 simulator to inexperienced users, without demanding a profound knowledge of either programming skills, that are needed to create a fully operable input simulation file, or a good scripting knowledge, that is required to parse the large amounts of the output data, that the simulation creates. Our solution provides both of these capabilities in a neat and friendly way.

2 Related Work

The mail goal, as it was mentioned in previous chapter, was to simplify the network simulation as much as possible. This included everything from scenario creation to the analysis of the outputs. We wanted to focus on the wireless networks because they have become more and more popular and a proper understanding of wireless technologies would be essential to the future students of computer networks. We wanted to use existing tools to make our project even better.

We had to choose which network simulator fits best out needs. It is clear that a commercial simulator is not a way to go. Basically, there were two options: network simulator 2 (ns-2) and network simulator 3 (ns-3) which are both available under GNU GPL license. Since the simulator is meant to be the core of our solution, we had to make proper choice regarding both simulators' characteristics, input and output formats and supported protocols.

Ns-2 is old and stable simulator. Its development started at Berkeley University in 1996. It runs on Unix-based systems and the core is written in C++. Simulation scenarios are written in OTCL (Object Tool Command Language). The reason why two languages are used is that writing code in C++ requires a lot of time, debugging, etc. But creating scenarios requires an agile way of writing code, so OTCL (also referred simply as TCL) is used here [1] [6].

On the other hand, we have ns-3, a fresh new simulator. Development of this simulator has started because ns-2 was technically old and the architecture was not designed to offer easy extension of simulator core. Despite its similar name with ns-2, it is coded from scratch. Ns-3 core is written in C++ and simulation scenarios can be written in C++ or Python. Here, we can find a similar approach where a scripting language is used to create simulation scenarios [2] [5].

The difference between ns-2 and ns-3, considering the wireless networks, is that ns-2 did not support wireless network simulations in its first versions. This functionality was added later. Ns-3 has had this functionality from the beginning, moreover, it was one of the ns-3 authors' goals to focus on wireless networks and simplify simulations of wireless networks. This means that wired and wireless simulations are very similar when writing scenario in ns-3. However, this attribute is not very important for our project because users will not have to write a scenario, they will use our comfortable graphical interface. A very big advantage of ns-2 is its protocol support. It has many protocols implemented in simulation core, plus there is a huge community of users who have created a lot of extensions.

To sum it up, we identified the main advantages and disadvantages of each simulator [8]:

- Ns-2
  - Advantages
    - Stability
    - Many extensions and tools available from community
    - Many protocols supported
  - Disadvantages
    - Old core architecture

    - Different scenarios for wireless and wired networks
- Ns-3
  - Advantages
    - Easy extensible architecture
  - Disadvantages
    - New simulator, many bugs are not fixed
    - Number of supported protocols is low
    - Lack of properly tested protocols

Finally, we decided to use the ns-2 simulator. The main reason was its stability and the number of supported protocols. In the next years, ns-3 will become a powerful tool with easy extensible architecture but now it just suffers from lack of properly verified protocols.

3 NS2IT Overview

NS2IT is a solution to make your network simulations as easy as they can be. It consists of four main modules which are connected together as displayed in Figure 1.

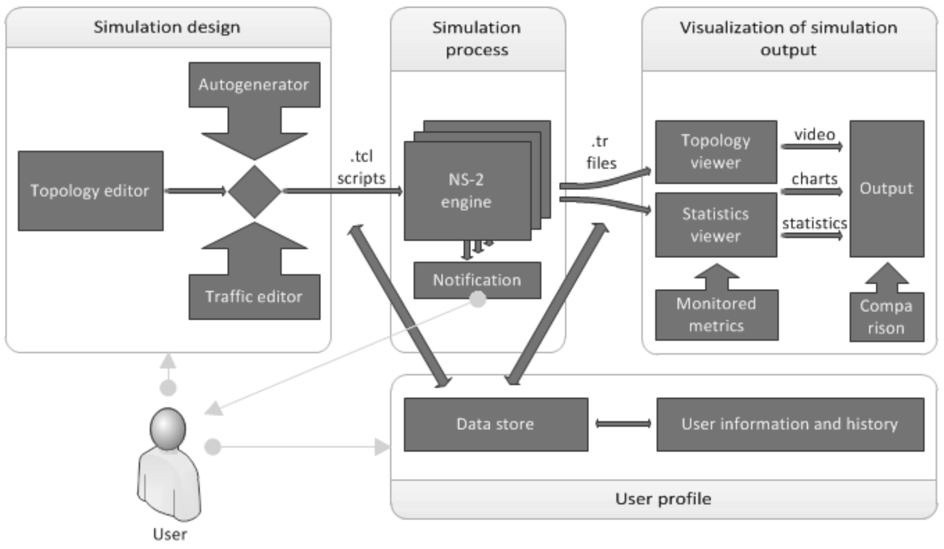

Figure 1. NS2IT architecture

The first of all, there is a module called simulation design. It is the very first module which interacts with user. In simulation design module, user can create his own network topology with *Topology editor* and *Traffic editor* tools. Topology editor is used to define which and how many elements will the network have. In this case, user has to create also traffic between these network elements, what can be achieved with *Traffic editor*. It is very important to know what network’s behaviour we want to simulate. Then, we can set up traffic properly. When everything is properly set up, module called *Autogenerator* will create TCL script, which is output of *Simulation design* module.

Simulation design module offers also another option. User can pick the network topology from our database of predefined and saved topologies. This is more simple way. It can be used when user does not know how to set up the topology and traffic. Also, it can be used to recreate previous simulation topology with new traffic and run simulation again.

The second module, called *Simulation Process*, is based on ns-2 network simulator. We use more ns-2 instances to run more simulations simultaneously. Input to second module is TCL script from *Simulation design*. Every instance of ns-2 will create output trace file right after simulation described in input TCL script ends. In simulation process we add feature *Notification*. Some of simulations can take really long time (couple hours), so NS2IT can notify user (via e-mail) when simulation is done.

A very important module, which makes our solution unique, is *Visualization of Simulation output*. Input for this module is trace file from ns-2 engine. This is the process where simulation output file is modified for visualization components. We prepare input for these components based on user's requests entered in *Monitored metrics* forms. There are two types of viewers. The first one is *Topology viewer* where user can watch his simulation and traffic. The second one is *Statistics viewer*. Based on monitored metrics, here are shown charts and statistics from simulation. Output of the third module is one summary of the whole simulation. Here, the user can see information which he requested in previous steps.

If there are more simulation summaries, user can pick them up and show next to the each other. This is useful in case when user would like to compare behaviour of network under different circumstances (e.g. the same topology with different traffic, etc).

The last one from main modules is *User Profile*. In this module there are no working components which cooperate to get summary done. Nevertheless, it is important for the whole solutions. This module is responsible for storing and handling all users' data. Every previous created TCL script and trace file can be recreated and modified to run a new simulation without the need to create everything from a scratch. All simulations' summaries are stored in a history, so they can be picked up anytime to compare with a fresh new simulation summary on comparison screen.

As we can see in Figure 1, there is an one-way flow relation among first three main modules. The output from one module is input for the next one. Only the fourth module *user profile* is related by a two-way flow with the input and output of the second module *Simulation Process*. This means that any TCL and trace file is immediately stored in *user profile*'s database and any of these files from database can be reused in a new simulation process.

4 Simulation preparation

As a standard ns-2 input is considered script written in TCL language. There are lot of things needed to be defined in this script such as network topology parameters, communication protocols and simulation timing [4]. Due to complexity of these scripts and diversity of simulation possibilities, it is very difficult to achieve full automation of script preparation process. Therefore our solution provides only few possible simulation inputs.

The first possibility for user is to use already existing script, written by other user and stored in our database. Our solution includes also random generator for simple simulations. The advantage of this possibility is that there is no need for users to have any knowledge about ns-2 simulator and TCL scripting. They can just NS2IT. Running simulation made by other user or generated by our system may help beginners to become familiar with our system functionality and capability. In addition, random generated simulation scenarios offers great possibility for network behaviour observation, considering randomness typical for Internet environment.

For advanced users there is a possibility to run their own home made simulations on our system. In this case may be our topology and traffic editors helpful, by partially automating TCL script creation. Nevertheless, good knowledge of ns-2 simulator and TCL scripting language is necessary. On the other hand, this method of input grants the user maximum control over simulation parameters. Before starting the simulation, user should choose network metrics he wants to evaluate and visualize after simulation to ensure their recording during the simulation.

After successful completion of the simulation, the user will be notified via e-mail and then will be able to work with the results through visualization screen interface.

5 Data collection

The main goal of proposed system is relevant data collection during the simulation and its transparent evaluation and comparison. However, not all required data necessary for network metrics computing are in ns-2 standard output known as trace file by default. Therefore, there is a need to set required data collection in TCL script just before the simulation for its later processing.

We use numerous shell scripts for computing of network metrics that the user intends to observe. In calculation process, there are used data just recorded in trace file or other output files generated during simulation. Data from this output are sorted and filtered according to their relevance to particular metrics observed by the user. Calculated metrics can be further graphically and statistically evaluated and simulations can be compared with each other.

6 Visualization

The user interface is common XHTML 1.0 Transitional web site generated by server-side scripting language PHP. This enabled us to create dynamic web application and assured cross-platform availability. To provide better human-computer interaction, we used AJAX (Asynchronous JavaScript and XML) programming features [3]. Many of our solution features as simulation, data collection and preparation, measurements recalculation etc. have to be processed by various server scripts. Therefore we enhance user AJAX interactions by server-web interface. This module enables authorized web applications execute server shell scripts and access their outputs. All this is done with efficient and secure way.

The user has to be logged in, to interact with the web application and simulations. This ensures application about the identity of the user. After successful authentication, user can create new simulation with various monitored measurements or access his previous simulations. Our interactive approach enables the user to watch the simulation progress and compare selected measurements.

Our main goal is to simplify reading of simulation’s outputs. To achieve this goal with user-friendly approach, our solution provides following types of visualization:

- simple topology design visualization,
- graphs and charts of simulation output,
- comparison of several measurements.

Simple topology design visualization is generated by 3rd party solution *iNSpect* [7]. After simulation is done by ns-2 engine, *iNSpect* creates topology screenshot in time of simulation beginning. User is also able to record video sequences of chosen simulation parts. This feature is very useful for wireless dynamic networks.

Graphs and charts are very useful for studying various network quantities as throughput, delay, delay variation (jitter), loss rate etc. After simulation is processed, user can choose network measurement or ask for measurement comparison. This request is send to server in asynchronous manner where needed data are chosen and calculated. Data are passed to plotter solution *GnuPlot* [9] and sent back to user as an image in form of chart as shown in *Figure 2*.

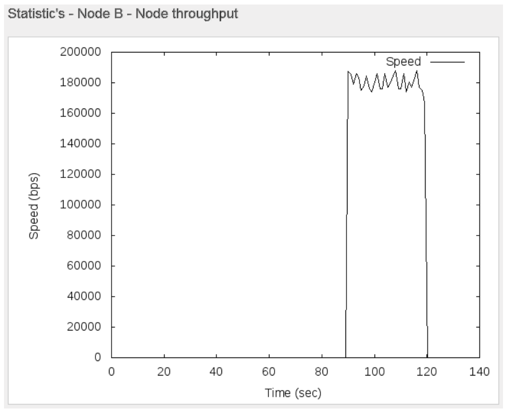

Figure 2 - Sample output chart generated from acquired data

7 Conclusions

In this paper we described our solution to simplification of the computer networks' simulation. We introduced the reader to the background of the networks' simulation and the difficulties it brings, the input file along, with the complexity of the simulation output. We gave the reasons why the simplification of simulations is needed and how we are going to achieve it.

This paper also explained the possible advantages of our solution, in the contrary to the classic way of using the ns-2 computer networks' simulator. We proposed a highly portable and cross-platform accessible solution that could be used in many kinds of simulation with a wide range of use. Our goal is to make the network simulation using ns-2 simulator accessible and easy-to-use to a wide range of users, from total beginners to network professionals.

8 References

[1] Ns-2 Roadmap, http://nsnam.isi.edu/nsnam/index.php/Roadmap, November 2010.

[2] Ns-3 overview,. http://www.nsnam.org/docs/ns-3-overview.pdf, August 2010

[3] Website Navigation Design, http://www.rocketface.com/organize_website/website_navigation.html, February 2009.

[4] Altman E., Jiménez T.: NS Simulator for beginners, Lecture notes, http://www-sop.inria.fr/members/Eitan.Altman/COURS-NS/n3.pdf, 2003.

[5] Henderson T., Lacage, M.: An end-to-end tour of a simulation.2, http://www.nsnam.org/workshops/wns3-2009/ns-3-tutorial-part-2.pdf, March 2009

[6] Issariakut T, Ekram H.: Introduction to Network Simulator NS2, 2009. ISBN: 978-387-71759-3.

[7] Toilers team: About iNSpect, http://toilers.mines.edu/Public/Code/Nsinspect.html, October 2010.

[8] Weingartner E., Lehn H., Wehrle K.: A performance comparison of recent network simulators. In: ICC'09 Proceedings of the 2009 IEEE international conference on Communications, 2009.

[9] Williams T., Kelly C.: GnuPlot documentation. http://www.gnuplot.info/docs_4.4/gnuplot.pdf, 2003.