Technical Note

Statistical Framework in NS-3

Lesly Maygua-Marcillo[1], Luis Urquiza-Aguiar[1], Martha Paredes-Paredes[1]

[1] Departamento de Electrónica, Telecomunicaciones y Redes de Información, Facultad de Eléctrica y Electrónica, Escuela Politécnica Nacional (EPN), C. Ladrón de Guevara E11-253, Quito PO.Box 17-01-2759, Ecuador; lesly.maygua@epn.edu.ec, cecilia.paredes@epn.edu.ec.

* Correspondence: luis.urquiza@epn.edu.ec; Tel.: +34-93-401-5997

Abstract: NS-3 is one of the most used Network Simulators because it includes different modules, applications and protocols to simulate a wide range of network. Since new proposals usually involve a large number of simulations, a procedure to automatize the simulation process and extract statistics is very useful. "Statistical Framework" (SF) is an important module of NS-3 because it fulfills this need. SF allows us to collect simulation data and obtain statistical reports for NS-3. In this report, we present a detailed description of statistical framework module and two useful examples of its configuration.

Keywords: network simulator, NS-3, statistical framework

---

1. Introduction

When we use network simulators it is important to obtain the detailed information of the results of the simulations. [1] This information can be obtained in different ways, for instance generating pcap traces [2], printing files, graphics with gnuplot and flowmonitor [3].
NS-3 contains a module, named Statistical Framework (SF), that allows us to get data and statistics for analysis of simulation results in an automatic way. Therefore, we can optimize the simulation process by using statistical framework[4]. The rest of this report is organized as follows: Sec. 2 briefly describes SF. Then, Sec. 3 reviews the files needed to run a simulation with SF. After that, in the Sec. 4 we show the configuration of the files described in Sec. 3. Finally, we will present two examples using SF.

2. Overview

Statistical framework provides functions to calculate, record and present data and statistics to analyze network simulations [4].

This framework is located in the directory `examples/stats`. There are various files that allow data collection and generation of graphics in this directory. The core components of the framework are the following [4]:

- One experiment trial is conducted by one instance of a simulation program (parallel or serially).
- A control script executes instances of the simulation, varying parameters as necessary.
- Data is collected by connecting the stat framework to existing trace signals. The data is stored in a database for plotting and analysis.
- Trace signals or direct manipulation of the framework may be used to instrument custom simulation code.

Fig. 1 shows the basic components of the framework and their interactions.

We can see in the picture that the process is managed by a experimental control (lower rectangle), which executes the different simulation instances (squares). Within each square the construction of the simulation is carried out. Finally, a data base is generated for further analysis.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2018 doi:10.20944/preprints201808.0158.v1

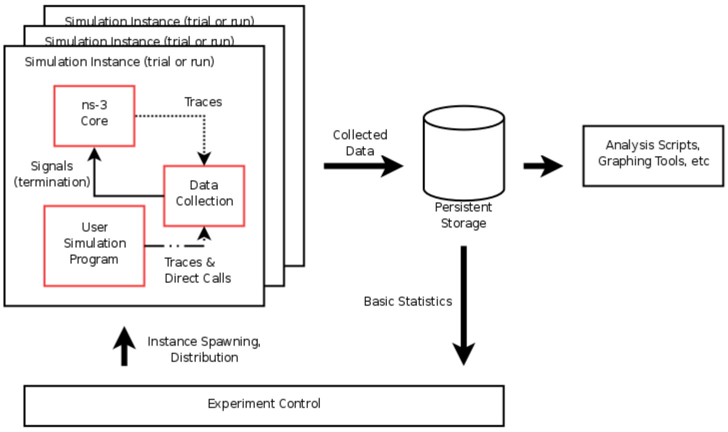


Figure 1. Component files of statistical framework [4]

3. Component files

The component files of statistical framework are shown in Fig. 2. They are the following:

- file.cc: File where the simulation program is created. It contains protocols, nodes, applications, data collector and generator.
- file-apps.(cc,h): They implement the methods that are used for the data collection in the simulation.
- file.gnuplot: It allows us to generate the graphics.
- file.sh: It is a control program developed in Bash (Bourne-Again SHell) that allows the execution of multiple simulations [7]. In addition, this file allows to vary simulation parameters. It collects the results and sets the selected output format (database, text file for OMNeT ++, image). This file will be executed within the `examples/stats` directory, using the following command:

```
bash example.sh
```

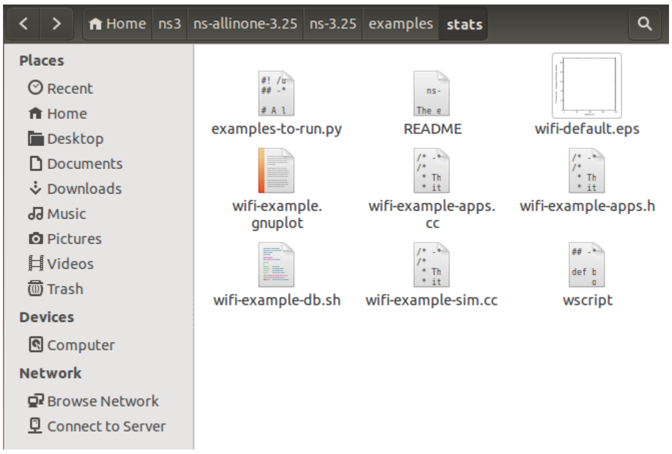


Figure 2. Component files of statistical framework

4. Configuration

Now, we explain step by step the configuration that must be done in the files aforementioned in the previous section. For this, we will focus on the default files that has the framework.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2018 doi:10.20944/preprints201808.0158.v1

4.1. wifi-example-sim.cc

This is the main script in the framework because contains the structure of simulation (protocols, nodes, applications, collector, among others).

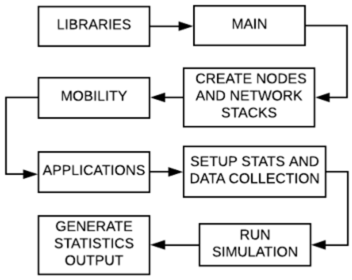
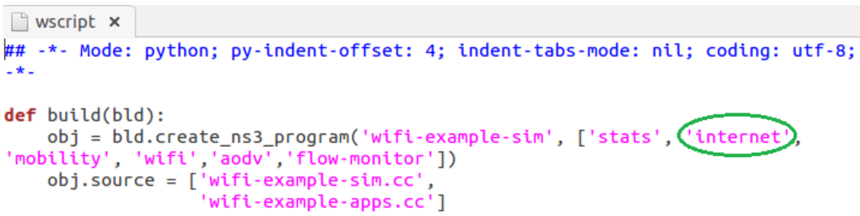


Figure 3. Structure of file

Each block of the Fig. 3 shows the structure of file.cc. To generate a simulation we must include these blocks according to the scenario that you want to simulate. For example, there are cases in which one scenario requires mobility in the nodes while in others it is not necessary. Everything regarding protocols, devices, generation and data collection must be configured in this file.

4.2. wscript

The configuration of this script is important because it includes all the modules that are used in the simulation. If a simulation requires additional modules it is necessary to add these, in the script. Otherwise, problems arise in the execution of the simulation.



```
wscript ×
## -*- Mode: python; py-indent-offset: 4; indent-tabs-mode: nil; coding: utf-8;
-*-

def build(bld):
    obj = bld.create_ns3_program('wifi-example-sim', ['stats', 'internet',
'mobility', 'wifi','aodv','flow-monitor'])
    obj.source = ['wifi-example-sim.cc',
                  'wifi-example-apps.cc']
```

Figure 4. Wscript

In the Fig.4 we show the context of **wscript** file. The green circle shows one of the modules included in the simulation.

4.3. wifi-example-db.sh

The configuration of this file will allow us:

– To execute several simultaneous simulations: To configure this tool we must add the following command lines in the file:

  Number of trials: `TRIALS="1 2 3 4 5 . . ."`
  Loop size: `for trial in $TRIALS`.

– To vary simulation parameters: At the beginning of the file the parameters must be configured specifying their variation. For example, if you want to vary the distances or the number of nodes in a scenario you must add:

```
DISTANCES="90 100 110 120 130"
SIZES="5 10 15"
```

This automatizes the simulation process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2018 doi:10.20944/preprints201808.0158.v1

5. Examples

Below, we show two examples using statistical framework in ad-hoc network simulations.

5.1. Example 1

In this example we simulate an ad-hoc network with five nodes using AODV routing protocol [5],[6]. The scenario is shown in Fig.5. We consider the following items:

– Variation of distances between nodes.
– The transmission is made between node 0 and node 4.
– We will test with 3 trials.

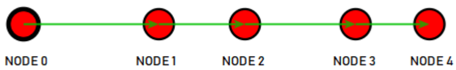


Figure 5. Scenario with five nodes

To run the simulation we must use the command line mentioned in the section 3.

In this case, the simulation has been configured in such way that the results are displayed on the screen and through a database. The Fig. 6 shows a bit of the results on the screen.

Tx	Rx	P.Tx	P.Rx	Thrput
10.0.0.1	10.0.0.5	100	97	16.1032 Mbps
10.0.0.4	10.0.0.3	1	1	253.91 Mbps
10.0.0.4	10.0.0.5	1	1	200.342 Mbps
10.0.0.3	10.0.0.4	1	1	37.3504 Mbps
10.0.0.3	10.0.0.2	1	1	55.6212 Mbps
10.0.0.2	10.0.0.1	1	1	207.232 Mbps

Figure 6. Results of the example 1

The Fig. 7 shows the database generated with statistical framework. In this case, we have used an external program to read the data.

In this way the results have been obtained successfully. At the end of the tutorial we will attach the scripts so that they can be analyzed.

5.2. Example 2

In this example we simulate an MANET [8] with different number of nodes using AODV [9] routing protocol and Random WayPoint as mobility model [10]. The scenario is shown in Fig.8.

We consider the following:

– Variation in the number of nodes (10-20-30).
– The transmission is made between two specific nodes.
– We will test with 3 trials.

In the Fig. 9, we can see the results of a trial with 10 nodes.

Similar to the example 1 we have generated the results on screen and in a database.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 August 2018 doi:10.20944/preprints201808.0158.v1

run	name	variable	value
run--90-1	node[0]	wifi-tx-frames	203
run--90-1	node[4]	wifi-rx-frames	204
run--90-1	node[0]	sender-tx-packets	100
run--90-1	node[4]	receiver-rx-packets	100
run--90-1	node[0]	tx-pkt-size-count	100
run--90-1	node[0]	tx-pkt-size-total	102400
run--90-1	node[0]	tx-pkt-size-max	1024
run--90-1	node[0]	tx-pkt-size-min	1024
run--90-1	node[0]	tx-pkt-size-sqrsum	104858000
run--90-1	node[0]	tx-pkt-size-stddev	0
run--90-1	.	delay-count	100
run--90-1	.	delay-total	710099096
run--90-1	.	delay-average	7100990
run--90-1	.	delay-max	41765295
run--90-1	.	delay-min	6701099
run--95-1	node[0]	wifi-tx-frames	203

Figure 7. Scenario with five nodes

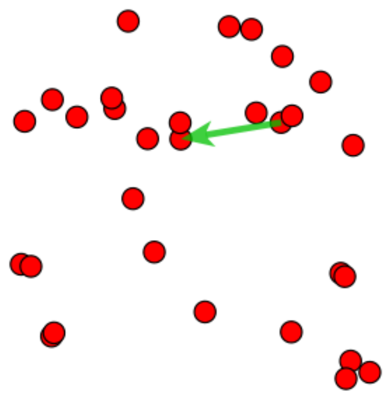


Figure 8. Manet with 30 nodes

NODOS: 10

Tx	Rx	P.Tx	P.Rx	Thrput
10.0.0.1	10.0.0.10	100	98	16.2704 Kbps
10.0.0.9	10.0.0.4	1	1	70.4079 Kbps
10.0.0.9	10.0.0.10	3	3	2.2436 Kbps
10.0.0.4	10.0.0.9	2	2	1.22938 Kbps
10.0.0.4	10.0.0.1	3	3	0.107184 Kbps
10.0.0.9	10.0.0.1	1	1	28.2513 Kbps
10.0.0.8	10.0.0.1	2	2	0.024996 Kbps
10.0.0.8	10.0.0.10	1	1	29.9818 Kbps
10.0.0.10	10.0.0.1	1	0	-0 Kbps
10.0.0.1	10.0.0.9	1	1	26.5173 Kbps
10.0.0.1	10.0.0.8	1	1	48.2538 Kbps

Figure 9. Results of transmission in a Manet with 10 nodes

6. Conclusions

In this note, we have reviewed how to automatize the simulation process in NS-3. To achieve this objective we resort in the Statistical Framework of NS-3, which can manage a batch of simulations. We have provided detailed information on this process by explaining two examples step by step. Future work includes simple modifications to the main script to take advantage of parallel processing.

Supplementary Materials: The following are available at https://perpapr2.lfurquiza.com/results , the examples of SF used in this tutorial

Acknowledgments: The authors gratefully acknowledge the financial support provided by the Escuela Politécnica Nacional, for the development of the project PIJ-16-01 - "Modelamiento de la Packet Error Rate (PER) incluyendo condiciones de Peak-to-Average Power Ratio (PAPR) para transmisiones Ad-Hoc".

Appendix Running examples in NS3 virtual machine

In order that to enhance the usefulness of this framework, we invite interested reader to review our previous tutorial "Deployment of NS-3 with Eclipse IDE" available in [11]. In this link, you can find a virtual machine with NS-3 ready to work. To simulate the examples that we presented with the statistical framework in this tutorial, you must follow the next steps:

1. Get the virtual machine.
2. In the virtual machine, go to the directory: `ns3-27/examples/stats`.
3. Replace the file `"wifi-example-sim.cc"` by the file `"example1.cc"`. Change the name of the latter to the default name `"wifi-example-sim.cc"`.
4. Replace the file `"wifi-example-db.sh"` by the file `"example1.sh"`
5. Run the simulation with the command: `bash example1.sh` inside the directory mentioned in the second step.

1. M. Solera and F. Ruiz, "Simulación de Protocolos de Enrutamiento para Redes Móviles Ad-Hoc Mediante la Herramienta de Simulación NS-3," 2014. [Online]. Available:http://repositorio.cedia.org.ec/handle/123456789/960.
2. NS-3 Tracing (2017), [Online]. Available:https://www.nsnam.org/docs/manual/html/tracing.html
3. Results in ns-3 (2014), [Online]. Available:http://repositorio.cedia.org.ec/bitstream/123456789/960/14/T_14_Procesadoresultados_vf.pdf
4. Statistical Framework (2017), [Online]. Available:https://www.nsnam.org/docs/manual/html/statistics.html
5. L. Felipe, U. Aguiar, A. I. Co-directora, C. Tripp, and B. Barcelona, "Proyecto final de máster: Design and implementation of routing protocols with anonymity for vehicular ad-hoc networks in urban environments," 2012.
6. Y. Mai, Y. Bai, and N. Wang, "Performance Comparison and Evaluation of the Routing Protocols for MANETs Using NS3," vol. 5, pp. 187–195, 2017.
7. Tesis (2018), [Online]. Available:https://www.nsnam.org/docs/manual/html/statistics.html
8. B. Panda, J. Swain, D. Mishra, and B. Sahu, "Analysis of effect of mobility and transmission power on AODV and DSR in mobile Adhoc network," IFIP Int. Conf.Wirel. Opt. Commun. Networks, WOCN, 2014.
9. K. Sharma, N. Mittal, and P. Rathi, "Comparative Analysis of Routing Protocols in Ad-hoc Networks," Int. J. Adv. Sci. Technol., vol. 69, pp. 1–12, 2014.
10. Random Waypoint Model (2008), [Online]. Available:https://www.netlab.tkk.fi/~esa/java/rwp/rwp-model.shtml
11. Vega-Sánchez, J.; Maygua-Marcillo, L.; Urquiza-Aguiar, L.; Barbecho-Bautista, P. Deployment of NS-3 with Eclipse IDE. Preprints 2018, 2018050313 (doi: 10.20944/preprints201805.0313.v1)