Experiences on Evaluating Network Simulators: A Methodological Approach

Michel Bakni, Yudith Cardinale, Luis Moreno

Experiences on Evaluating Network Simulators: A Methodological Approach

Michel Bakni[1], Yudith Cardinale[2], and Luis Manuel Moreno[2]
[1] Univ. Bordeaux, ESTIA, F-64210 Bidart, France
[2] Dpto. de Computación y T.I, Universidad Simón Bolívar, Caracas, Venezuela
Email: m.bakni@estia.fr; {ycardinale, 13-10934}@usb.ve

Abstract —There exists a variety of network simulators, used to imitate the protocols, nodes, and connections in data networks. They differ in their design, goals, and characteristics. Thus, comparing simulators requires a clear and standardized methodology. In this paper, based on a set of measurable and comparable criteria, we propose an approach to evaluate them. We validate the suggested approach with two network simulators, namely Packet Tracer and GNS3. In that regard, a test scenario is put forward on the two simulators, both in Linux and Windows environments, and their performance is monitored based on the suggested approach. This paper does not propose a method for selecting the best simulator, but it rather supplies the researchers with an evaluation tool, that can be used to describe, compare, and select the most suitable network simulators for a given scenario.

Index Terms—Network Simulators, Evaluation Criteria, Comparison Approach, Packet Tracer, GNS3

I. Introduction

Network simulation is one of the most powerful and predominant evaluation methodologies in the area of computer networks. It is widely used for the development of new communication architectures and network protocols, as well as for verifying, managing, and predicting their behavior. Network simulators have grown in maturity since they first appeared and they have become an essential tool of the research domain, for both wired [1] and wireless networks [2].

Simulators are easy to control, they save efforts in terms of time and cost, and allow easily repeating of the same experiment with input changes. However, they are only approximate models of the desired setting. Although the simulator is capable of simulating the whole network model, it is not possible to cover all of its aspects with the same level of details. Instead, the simulator focuses on one or two of the following aspects [3]: algorithms, application protocols, network protocols, and hardware. Then, the simulator fills the gaps in the other aspects using assumptions [4]. Hence, more studies are needed to establish guidelines that support researchers in the tasks of selecting and customizing a simulator to suit their preferences and needs.

One of the main motivation of this paper is to address this lack of guidelines. We propose a simple approach, based on a set of criteria to cover aspects related to the simulation process, as well as aspects related to the evaluation of the network simulator. Our criteria include ten items that can be applied to different network simulators in order to obtain a measurable and comparable assessment.

We do not pretend that our approach is a methodology that identifies the best network simulator, as there are wide varieties of possible network scenarios that demand different requirements and have a significant effect on the simulator performance. In consequence, the choice of the simulator is a scenario-dependent problem. Wherefore, this paper demonstrates how the suitability of simulators can be validated for particular needs, following a methodological approach comprised of simple steps and based on a set of criteria.

To illustrate the applicability of our proposed approach, we evaluate two network simulators, namely Packet Tracer and Graphical Network Simulator 3, widely known as GNS3.

Packet Tracer is a simulation tool for both wired and wireless networks. Moreover, it can be used to build complex topologies that simultaneously run different protocols, thus, it is a powerful tool to implement complex and inter-protocols scenarios that includes sophisticated topologies [5]. Packet Tracer allows the simulation of Cisco's IOS with a high degree of accuracy. It also allows simulating other information systems, such as servers and terminals, as well as some concepts of Internet of Things (IoT), but with a high level of abstractions. The simulator has an attractive customizable graphical user interface (GUI) and allows contribution for multi-users activities [6].

GNS3 is a network emulator that is used to run different network operating systems that were developed to run on a specific hardware. The emulator provides a hardware-independent interface for the operating systems to run as virtual machines on the same host. Thus, performance is a major topic [7]. The emulator has a built-in GUI and can easily inter-operate with other well-known network software such as Wireshark and virtual box, making benefits of their capabilities. GNS3 was suggested for both pedagogical [8] and research [9] purposes.

Manuscript received Month Day, 2013; revised Month Day, 2013.
doi:10.12720/jcm.v.n.p-p

Even though, both simulators can be considered for research activities, there is a lack of systematic and comprehensive studies that highlight their capabilities. Hence, we demonstrate that if they are properly evaluated, they can become available options for researchers to pursue in their studies.

In summary, the contribution of this work is twofold:

1) Propose a methodological approach and a set of criteria to evaluate network simulators; and
2) Evaluate Packet Tracer and GNS3 features, performance, advantages, and disadvantages based on the criteria previously proposed, to show their suitability for researchers in network domains.

The remainder of this paper is organized as follows. In Section II, we survey recent works focused on proposing criteria or methodologies to evaluate network simulators and studies that have evaluated the two simulators. Our proposed approach is described in Section III. How the methodology works, is illustrated in Section IV by evaluating Packet Tracer and GNS3. We draw some recommendations based on the results. Finally, Section V highlights conclusions and perspectives.

II. RELATED WORK

In this section, first, we survey studies focused on proposing methods and criteria to evaluate network simulators and then we describe works that have evaluated Packet Tracer and GNS3. We highlight their limitations and differences compared with our proposal.

A. Network Simulators Evaluation

The Virtual InterNetwork Testbed (VINT) Project [10] intended to develop methods and tools to address the scale and heterogeneity of the Internet protocols. One important result of the work was adding definitions related to the simulation issues, including the type and the nature of simulators, in addition to highlighting different interactions of the simulated protocols. In [11], there was another attempt to address the issues that concern the simulators developers concluded that there are four of them, namely the type of problem, the level of abstraction, the extensibility, and the diagnosis of existing codes. Later, a detailed and comprehensive study recognized modeling as a foundation stone in the choices of simulators [12].

In [13], nine evaluation criteria are proposed to evaluate wireless sensor networks. Some of them have been incorporated in our set of criteria. Some other works propose the evaluation of simulators in terms of computational run time, memory usage, and scalability [3], [14], [15], [16], [17].

Even though these works propose some aspects that should be taken into account to evaluate simulators, none of them propose a coherent and complete method to do the evaluation, neither evaluate Packet Tracer and GNS3, as we do.

B. Packet Tracer and GNS3 Evaluation.

A variety of studies has evaluated one or more different aspects of Packet Tracer and GNS3. Authors in [18] used Packet Tracer and GNS3 to study the traffic in networks that support both IPv4 and IPv6, either using the dual stack technique or the tunneling. As a result, the article concludes that Packet Tracer is "easy to use", but it does not simulate all services and functions like tunneling.

In [19], a comparison between Packet Tracer and GNS3 is presented in an academic context. Both simulators are evaluated as learning tools in computer network courses. After the experiments, the authors conclude that GNS3 is capable to run Cisco IOS and to create more realistic topologies when compared to Packet Trace. Another use for GNS3 as an educational tool and pedagogical comparison with Packet Tracer can be found in [20]. A comparison study mentioned 12 comparative items between GNS3 and Packet Tracer [21]. The items are: the GUI design, the memory requirement, the hardware models supported, the protocol supported, the commands supported, the computer systems supported, the ability to analyze traffic, the ability to exchange the topology, the types of connection supported, the certifications that use the simulators, the license, and the support for the instructor.

Other works focus on the evaluation of Packet Tracer in different contexts. In [22], the problem of support for tunneling in Packet Tracer was addressed. In their study, GRE tunnels were properly simulated in addition to many IPsec features. This is a good example of the problem of lack of comprehensive studies. In fact, the tunneling feature was supported since the version 5.3, which was released in 2010. In [23], a detailed study of the dynamic routing used Packet Tracer as a simulator. Four routing protocols were evaluated, they are Routing Information Protocol (RIP) (version 1 and 2), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). The article does not highlight on the simulator itself, thus, the simulation results were presented and discussed based on only the technical side of the network. A similar study that covers only RIPv2 and EIGRP can also be found in [24]. In [25], the effectiveness of Packet Tracer as a learning tool to teach routing protocols is demonstrated.

In [26], a performance study is presented based on a scenario implemented using Packet Tracer, The scenario covers both IPv4 and IPv6 networks. The study focuses on the delay, routing traffic and convergence when OSPF and EIGRP are used. In the end, the authors concluded that Packet Tracer is a useful tool for routing studies, especially to select a routing protocol and to design the optimal routing topology based on that.

A comprehensive study of the Link Layer technologies and protocols can be found in [27]. Trunk ports, static Virtual Local Area Networks (VLANs), Dynamic VLANs, Inter-Switch Link (ISL), and IEEE 802.1Q were tested and verified. In addition to that, the authors implement a scenario using both OSPF as a routing protocol, Dynamic Host Configuration Protocol (DHCP) as a client/service protocol, and access lists as a security application. Packet Tracer was able to simulate the network and trace the packets when different-layers protocols were simultaneously used.

In [28], the use of the Packet Tracer as an assessment tool is discussed. The application has an advantage that it allows the user to stop the simulation at a given moment and check all the messages exchanged among different network nodes. The author concludes that although the simulator was not primarily designed as an assessment and measurement tool, it can be used to aid certain educational purpose. The use of the Packet Tracer as an assessment tool is related to the nature of the study, while it does not appear to be used in performance studies, such a tool can add a benefit in the studies of the routing protocols.

GNS3 have been the focus of many other works. In [29], it was proposed as a simulation tool for pedagogical purposes. That article provides a brief summary of its advantages and disadvantages. Authors explain that the principal requirement of the simulator is the high resources needed by the external operating system to make devices to work. The reason behind that is the nature of the emulation process allowed by GNS3. In [30], a solution for the resource consumption problem is provided.

In [31], GNS3 is used to provide a simulation environment of the migration technologies from IPv4 to IPv6. The study includes three types of technologies: dual stack, tunneling, and translation. GNS3 was able to simulate the scenario suggested for each technology. The authors mentioned that GNS3 was used only for the simulation and they used another application for analyzing. In [32], GNS3 was preferred as the simulator to evaluate the performance of IPv4 and IPv6 in terms of three routing protocols (i.e., RIP, EIGRP, and OSPF), due to its capacity of modeling real word scenarios.

GNS3 was used in [33] as a simulation tool for MultiProtocol Label Switching (MPLS) technology. Authors made use of its ability of emulation to create different types of traffic managed with MPLS. Other uses for the simulators are in the security domain. In [34], the simulator works as a simulation testbed for several of IPv6 attack scenarios, GNS3 emulates different IOS for Cisco routers in the proposed scenarios. GNS3 is used to simulate an SQL-insertion cyber-attack in [35].

Although all these works cover many aspects of Packet Tracer and GNS3, they mainly address the aspects from the comparison point of view, without considering the simulators own capabilities or their maximum limits. In addition, most of these studies do not include items for the performance and memory requirements of the simulators. Finally, all these works insist on providing results rather than developing a coherent methodology, that, in turn, make them intended for students and teachers more than researchers.

In this paper, we provide a systematic approach to describe a generic network simulator, considering a set of criteria that include both a characterization of the simulator properties and a way to measure its performance.

III. Evaluation Approach and Criteria

In this section, we explain how we address the problem of evaluating network simulators. First, we describe the proposed evaluation approach, then, we provide, in detail, a list of ten criteria to be used as measurements for the evaluation.

A. Evaluation Approach

As far as we know, there is no fixed approach or methodology to evaluate network simulators. As long as the developing of simulators continues, any methodology will remain subject to modernization and modification [36]. Thus, we do not pretend to establish a methodology, instead, we propose a single approach based on few steps and a set of criteria to demonstrate how the suitability of simulators can be validated for particular needs. The primary objective of this approach is to evaluate qualitative aspects, as well as to obtain measurable and comparable values after applying the approach to a network simulator to describe its behavior, capacity, and performance.

Hence, to evaluate simulators, we propose the following steps:

1) Establish a set of criteria. The evaluation of the simulator requires clear and accurate criteria to assess the different aspects of the simulator. Qualitative criteria can be described by words or numbers, while quantitative criteria need to be measured. Moreover, there can exist composite parameters, that are composed of multiple sub-parameters. In the next section, we provide precise and specific definitions of ten parameters that describe and evaluate simulators from different qualitative and quantitative aspects.
2) Establish the experiment setup. It is worthy to install the selected simulator(s) on different systems (e.g., Windows, Linux, MacOS) under the same architecture. The way that operating systems manage system resources and the produced overhead have an important impact on the behavior of applications.
3) Evaluate the qualitative criteria of the simulator(s). Revise the available documentation of simulator(s) and elaborate a table highlighting their characteristics.
4) Design a test scenario to evaluate the measurable criteria. Decide the network elements that will be simulated according to the protocols that are intended to evaluate. Define the number and type of experiments, as well as the time of the simulation, taking into account the criteria to be evaluated.

5) Evaluate the measurable criteria of the simulator(s) by executing the designed experiments. Elaborate tables and graphics to show the results in order to facilitate the analysis and comparison (if there is a case).
6) Elaborate a discussion by analyzing the results.

These steps can be applied to evaluate a single simulator or to compare several of them.

B. Criteria

The following parameters can be used to evaluate the simulator, a detailed and precise definition is provided for each of them.

1) **Nature of the simulator:** The simulation consists of a number of models that are executed to interact with each other. The nature of the simulation is an assessment of how the simulation is performed. Precisely, the use of the word *simulation* means that the entire process is programmed. It means that only the software aspect is involved in the simulation. However, if the word *emulation* is used, hardware is also involved in the simulation process [37].
2) **Type of simulator:** It is a characterization of the philosophy underlying the simulator's behavior. Network simulators are based on two philosophies: it is either a discrete-event simulator or a trace-driven one. In the first, an initial set of events is generated, representing the initial conditions. Those conditions, in turn, generate another set of events and so on. The process continues like that, until the end of the simulation.
   In the trace-driven simulation, all events to be simulated are added to the simulator in the form of inputs. Thus, it can simulate it and trace the outputs [38].
3) **License:** This criterion represents an evaluation of the capability to use the simulator from a legal aspect. Simulators can be private property or they can be developed under a free or public agreement.
4) **User interface:** It is an evaluation of how can a user interact with the simulator, This criterion includes two aspects:
   - **Graphical User Interface (GUI):** an evaluation of the support for the graphic interface. Is it an integral part of the simulator? What is the level of details it can show?
   - **Supported programing languages:** Can users interact with the simulator by programming scripts? Can users develop a piece of software to interact with the simulator?
5) **Supported platforms:** It is the characterization of the usability of the simulator source code on different platforms and operating systems [39].
6) **Heterogeneity:** It is an evaluation of the ability to simulate heterogeneous systems where different types of nodes can exist in the same scenario [40].
7) **Modeling:** It represents an evaluation of the ability to modify existing models in the simulator or to implement and test new ones.
8) **Level of details:** It consists on evaluating the level of aspects that are being simulated. Those aspects, sorted in descending order, are abstract algorithms, high-level protocols, low-level protocols, and hardware. The lower the level, the less the assumptions and the more the constraints [13].
9) **Supported technology and protocols:** In order to evaluate the support provided for the protocols, TCP/IP model is used [41]. It is a 4-layer stack model that classifies the network protocols, features and services according to the function. Starting from the top, these layers are application, transport, Internet, and link layers.
   We have excluded the routing protocols from this stack and combined them into a single item. The reason behind this is the distribution of the routing protocols in the layers of the model, this does not serve the primary purpose of this item, namely the assessment of support to the protocols.
10) **Performance:** The main purpose of the study of performance is to provide a general idea of the effectiveness of the simulator in terms of implementation time and the consumption of available resources. However, the proposed approach includes three factors for the performance study:
    - **CPU Utilization:** it is a measure of the application performance [42], which consists in the percentage of time spent performing the simulation process of the total processing time [43], i.e., the percentage of the processor cycles that are consumed by the simulation.
    - **Execution time:** it is the time needed to complete a simulated scenario; measured in seconds.
    - **Memory usage:** it is the amount of memory used by the application, measured in bytes.

In the next section, we apply the approach to evaluate Packet Tracer and GNS3.

IV. APPLYING THE APPROACH

This section is dedicated to the practical aspect, in which we apply the proposed approach to evaluate Packet Tracer and GNS3. In the following, we describe how the proposed steps and set of evaluation criteria are considered to evaluate both simulators. At the end, we discuss about the suitability of our proposed approach.

A. Step 1: Establish a set of criteria

Following the proposed approach leads to a 10-items description for the simulator. The considered set of evaluation criteria is the one presented in Section III-B.

B. Step 2: Establish the experiment setup

In order to apply the proposed criteria, we installed both simulators on two different systems, namely Linux Ubuntu 16.04 LTS and Microsoft Windows 10 version 10.0.14393. They were installed on the same computer with the following characteristics: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz with 16 GB for the RAM, 915 GB of the hard disk is allocated for Linux while 909 GB is allocated for Windows.

C. Step 3: Design the test scenario

After the installation, nine of the evaluation criteria can be pointed out, according to the documentation and general knowledge about Packet Tracer and GNS3. Only the performance criterion requires special scenario preparation. Table I shows the result of this step.

Some of the information presented in Table I was directly obtained from the official website of both simulators, such as supported platforms. Others, like the supported technologies and protocols, required running the simulators to test and verify whether the support exists.

D. Step 4: Evaluate the qualitative criteria

We designed a scenario involving several experiments, in which we used the Spanning Tree Protocol (STP) to measure performance determinants. Originally, the STP is used in a layer 2 switched environment to create a loop-free path to data traffic. By default, the protocol convergence time is between 20 to 55 seconds. Several factors can affect the exact value, including the network complexity and the timers values. To consider that, we established the duration of each experiment in 60 seconds, while the convergence time is the time needed for the protocol to converge. All the following was implemented on both early-mentioned Linux and Windows systems.

The scenario is built in a way that reflects the CPU utilization and memory usage. To achieve that, we adopted a meshed topology, whose size is increasing exponentially every time we are repeating the test. The basic component of the topology consists of four 2960 Cisco Catalyst switches arranged in a ring topology. Fig. 1 (a) shows the ring topology of the basic component, which is the scenario of the first test. Then, the second test is done with two basic components, i.e., eight switches, as shown in Fig. 1 (b). The third one is composed by four basic components, with 16 switches (see Fig. 1 (c)), and so on increasing the number of basic components exponentially with base 2, until 64 basic components, with 256 switches. In total, we conducted seven tests with 1, 2, 4, 8, 16, 32, and 64 basic components, on each system (Linux and Windows), for both simulators.

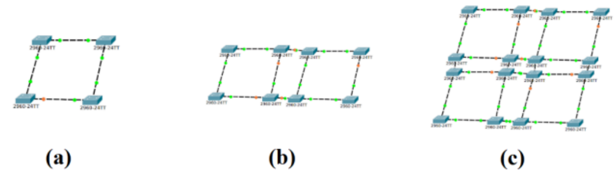

Fig. 1. The different topologies used in the suggested scenario, (a) a Basic component topology, (b) Two basic components topology, (c) Four basic components topology.

E. Step 5: Evaluate the measurable criteria

Information related to nine of the ten evaluation criteria are shown in Table I, representing the qualitative criteria. The scenario depicted in the previous subsection, was designed to evaluate the performance in terms of CPU utilization, memory usage, and converge time (i.e., the time in which STP converges), which are measurable criteria.

For both simulators, to obtain the performance values in Linux, we used Monit[1], an open source tool for monitoring processes on UNIX systems. For the tests in Windows, values were obtained from Task Manager, a built-in monitor of the CPU utilization and memory usage per process.

Fig. 2 shows the results of the CPU utilization of all the tests for Packet Tracer, when the suggested scenario is implemented on Linux. Fig. 3 shows the results for the same tests, when running the scenario on Windows. In both cases, we registered the percentage of CPU utilization every second during the simulation. Comparing both results tells that Windows is more suitable for this simulator in terms of CPU utilization.

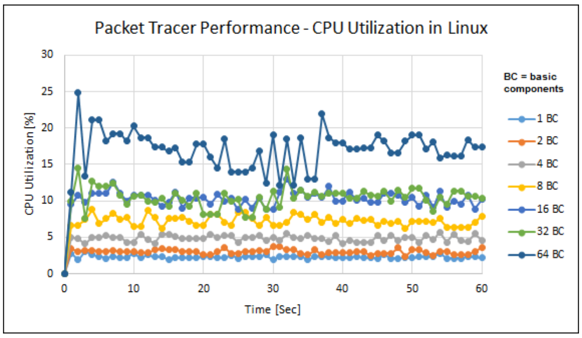

Fig.2. Packet Tracer CPU utilization - Linux.

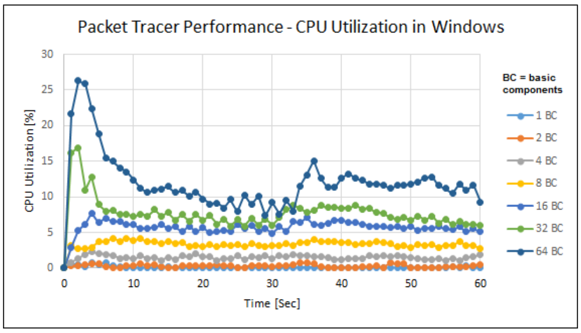

Fig.3. Packet Tracer CPU utilization - Windows.

[1] https://mmonit.com/monit/

TABLE I: NINE QUALITATIVE CRITERIA OF PACKET TRACER AND GNS3.

Criteria	Packet Tracer characteristics	GNS3 characteristics
Nature of the simulator	Simulator	Emulator
Type of simulator	Discrete-event	Discrete-event
License	Proprietary, but an End User License Agreement (EULA) exists	GPLv3
User Interface	**GUI:** Yes, a built-in GUI interface is supported, with a possibility to trace and store all events. Different languages are supported for the GUI including: English, Russian, German, Portuguese, Spanish and French. **Supported programing language:** None, it is private property, but scripting is allowed using the Cisco IOS Syntax.	**GUI:** Yes, a built-in GUI interface is available. **Supported programing language:** The simulator was built using Python, the distributed version does not allow direct changes, but the code repository is available on github (https://github.com/GNS3/gns3-server).
Platform	Linux, Android 4.1+, iOS 8+, and Microsoft Windows	Windows 7 (64 bit) and later, Mavericks (10.9) and later, any Linux Debian/Ubuntu distribution is supported.
Heterogeneity	Packet Tracer supports different types of real **routers** such as: Cisco 1941, Cisco 2901, Cisco 2911, and others, as well as different types of real **switch** like: Cisco Catalyst 2950, Cisco Catalyst 2960, Cisco Catalyst 3560-24PS. In addition to that, Linksys WRT300N **wireless router**, Cisco 2504 **wireless controller**, and Cisco Aironet 3700 **access point** are supported. Cisco ASA 5505 **firewall** is supported as well. Variety of IoT devices are supported.	GNS3 supports heterogeneity by providing an interface to run virtual machines, which has significant impact on the performance. Supported IOS: **Cisco** (IOU/IOL, vIOS/vIOS-L2, NX-OSv, ASAv, and others), **Juniper** (Olive, vSRX, and vMX), **MikroTik** (RouterOS and CHR), **Hosts** (Linux, windows, and Mac OS).
Modeling	It is not supported.	It is not supported.
Level of details	Packet level.	Bit level, using the Wireshark plug-in (https://www.wireshark.org/\#download)
Supported technologies and protocols	**Application Layer:** Protocols: DHCP, DHCPv6, FTP, HTTP, HTTPS, RADIUS, POP3, SMTP, SNMP, SSH, Telnet, TACACS. Technology: Access Lists, DNS, IoT, IoT TCP, SYSLOG. **Transport Layer:** Protocols: SCCP, TCP, UDP. **Network Layer**: Protocols: ARP, CAPWAP, HSRP, HSRPv6, ICMP, ICMPv6, IP, IPv6, NDP. Technology: IPSec, Cisco NetFlow. **Link Layer**: Protocols: Bluetooth, CDP, CTP, H.323, LACP, LLDP, PAgP, STP, USB, VTP. **Routing Protocols** BGP, EIGRP, EIGRPv6, OSPF, OSPFv6, RIP, RIPng,	It depends on the emulated operating system. Supports protocols at all levels and a wide variety of technologies, including Cisco technology

Fig. 4 displays a comparison of the memory usage for the same previous tests, for both operating system. We measured the percentage of memory usage of Packet Tracer at the beginning of each simulation test, i.e. the memory consumption is constant during the execution, there is no change.

Since Packet Tracer is a discrete-event simulator, it generates a subsequence of events that are gathered in a buffer list, this buffer is overflowed when the number of the basic components is more than eight, we encountered the same problem both in Windows and in Linux. Thus, it was not possible to obtain the convergence time of STP from tests whose topologies have more than eight basic components.

However, Fig. 5 shows the obtained results for the convergence time. As we note in Fig. 5, results both in Windows and in Linux are close in value, when there are eight basic components or less.

GNS3 needs high-performance requirements because it emulates the operating system at the hardware level, which imposes limits in term of scalability (i.e., there will be a point when all the available resources of the operating system are being used or allocated by the simulator, then there is no more expansion).

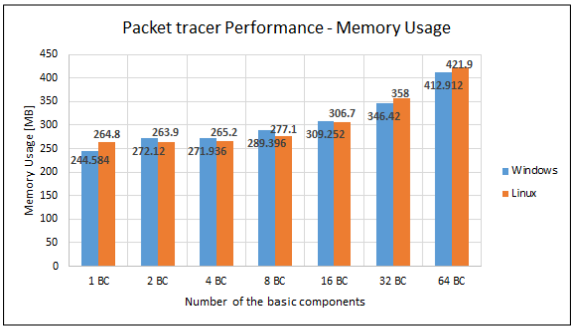

Fig.4. Packet Tracer memory usage.

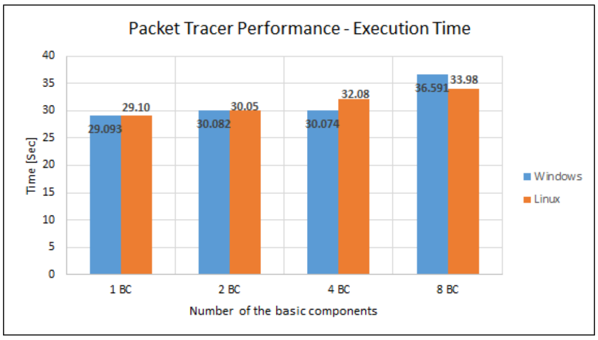

Fig.5. Packet Tracer CPU utilization execution time.

Consequently, we could not increase the number of the basic components beyond eight. Fig. 6 shows the CPU utilization when implementing the scenario in Linux as the number of the basic components change respectively as follow: 1, 2, 4, and 8. Fig.7 shows the same parameters but when the scenario is executed in Windows.

The two operating systems show two different ways of managing high CPU utilization. In Windows, regardless the number of the basic components, the utilization rate grows excessively to 100% for a limited period of time, but later decreases by 25-35% and then it grows up again in a significantly swinging pattern. In Linux, the utilization pattern tends to swing slightly 1-2% around a fixed value, it is almost 82-83% when there is only one basic component, and it is raised up into 91% when the basic components are four.

Fig. 8 shows the memory usage when the scenario is implemented using GNS3 in both Linux and Windows. In Windows. GNS3 presents a restrictive behavior to the increasing demand for memory, keeping a threshold around 3.6 GB, in which the operating system does not allocate more memory to the simulator. In Linux, the assignment of memory grows in an exponential way.

Fig. 9 displays the execution time of the scenario both in Linux and in Windows for GNS3. It is clear that the memory assignment strategy followed by the operating systems has an impact in the execution time of the simulator.

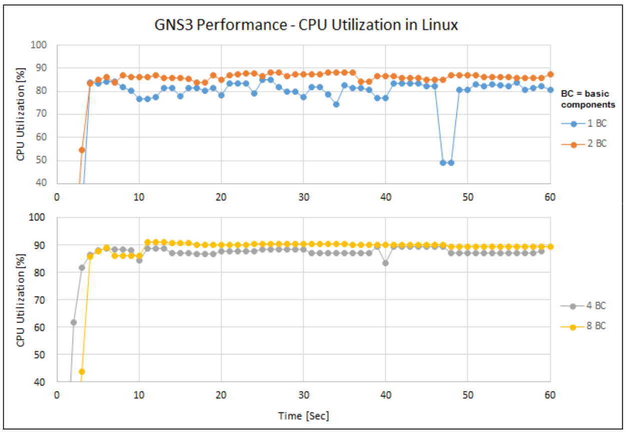

Fig.6. GNS3 CPU utilization - Linux.

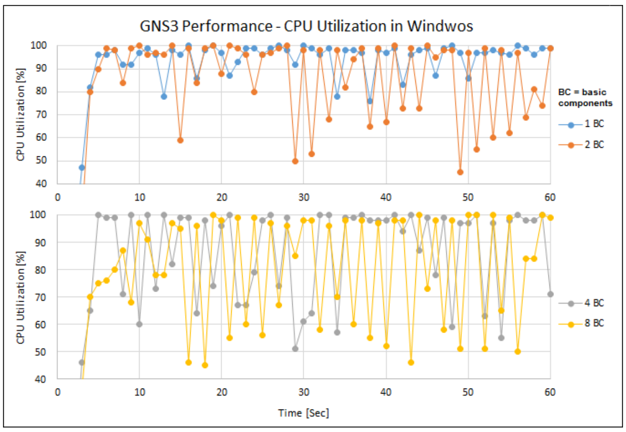

Fig.7. GNS3 CPU utilization - Windows.

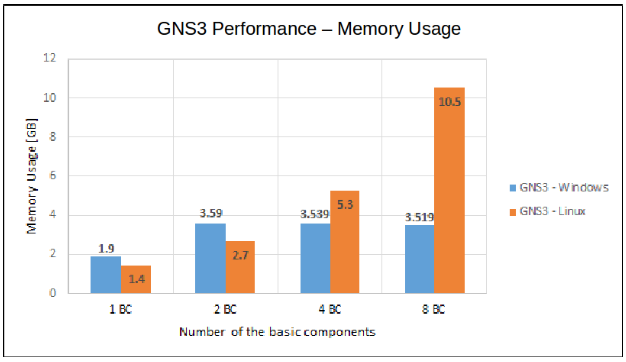

Fig.8. GNS3 memory usage.

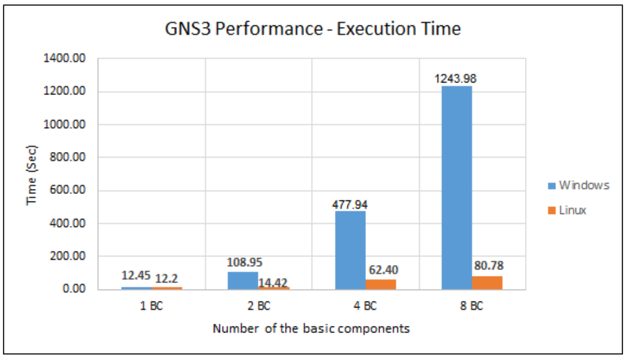

Fig.9. GNS3 execution time.

Because Windows restricts the memory allocation of GNS3, the execution time of the tests increases exponentially. In fact, when the simulation has 8 basic units, GNS3 needs more than 1200 seconds (i.e., more than 20 minutes) to perform the simulation on Windows. Unlike the simulator when it is running in Linux, which can do the same calculations in 80 seconds, thanks to the additional memory allocated by the operating system.

F. Step 6: Evaluate the measurable criteria

In this section, we present the analysis and the discussion of the evaluation of the two simulators, which are derived from the obtained results.

Analyzing the qualitative criteria in Table I, we can say that Packet Tracer supports a wide variety of protocols in each layer, this gives the researchers multiple choices to create different scenarios. In addition to that, it provides the same GUI and functions on both Windows and Linux platforms. On the other hand, GNS3 provides a similar diversity, not on the level of protocols and applications, but rather on the level of the operating system. Packet Tracer and GNS3 are different types of software (while Packet Tracer is a simulator, GNS3 is an emulator), which establishes differences at performance level. GNS3 trends to utilize more the CPU and demands much more memory to use. Even though, they present similar qualitative characteristics.

In terms of performance, the scenario was designed to drive the simulators to their maximum limits. In the case of Packet Tracer, the limit was reached with 8 basic components, when the buffer of events could not expand further. Otherwise, the simulator shows an ability to scale well for big topologies. On the contrary, when GNS3 was used, and due to its nature as an emulator, it rapidly adds constraints to on the scalability.

The proposed approach has successfully demonstrated the relationship between the performance parameters, namely the memory usage and the execution time. Moreover, the scenario provides an example that shows how the performance deteriorates when the emulator fails to manage the trade-off between the performance parameters. As a result, the execution time excessively increases when there are no additional memory to be allocated. This failure is due to restrictions in Windows, such an effect cannot be noted when the same scenario is running in Linux.

Finally, even though Packet Tracer is a private simulator, its available version is good enough for simulating complex topologies from both Wide Area Network (WAN) and Local Area Network (LAN) aspects. It does not allow researchers to test new protocols or algorithms, but it provides a massive set of protocols that can be used to create a large number of combinations of layered-protocols stacks. On the other hand, GNS3 provides emulation for a high variety of operating systems and hardware devices, which allows the emulation of scenarios near to real cases. Researchers can benefit this good characteristic. However, it has a strict limit of the scalability because of its software nature.

G. *Reflexions about the approach*

The application of our proposed approach to evaluating Packet Tracer and GNS3, allows us to point out some reflexions:

- A layered-protocols stack model is a powerful tool for categorizing the work done in the network by function, but there are some protocols that do not fit into a particular layer, because they perform functions belonging to more than one layer at the same time. Examples of those protocols are the Neighbor Discovery Protocol (NDP) and Address Resolution Protocol (ARP), they both work on the Internet and Link layers, and in this case, we categorize them in the upper layer, which is the network. Merging the technologies and protocols inside one criterion can become a complex issue if the simulator supports technologies that use more than one protocol, in different layers, this, in turn, will lead to a non-comparative item. In this case, it is better to separate technologies from protocols and by creating a new criterion. Then, the technologies item can have its own independent stack-layered model.
- The heterogeneity criterion needs to be described in more details, sub-criteria can be added based on further studies; the main goal is to enable the item to describe the simulator's ability to emulate different specific models or hardware.
- The study of performance characteristics, suggested by our approach, is scenario-oriented, i.e. a change in the scenario parameters can push the simulators to other limits that were not shown in this paper. For example, how much accurate the simulator can simulate or emulate a specific function or feature, such as the energy consumption of one group of nodes.
- We thought about adding a special criterion for the simulator version because it is an important piece of information, but it is related to each simulator itself, thus, it is not comparable among other simulators, that is why we did not consider the version as an item within the suggested approach.
- We are thinking of expanding the approach to include Wireless Sensor Network (WSN) simulators, but this requires further studies to modify the current approach or even developing an independent one. WSN requirements are different from those of wired ones. For example, mobility, energy consumption, energy harvesting, battery models, and others are specific-purpose concepts that are directly related to the nature of the WSN.

V. CONCLUSION

In this paper, we have addressed the difficulty of selecting a computer network simulator to fit a given scenario. To achieve that, we propose an approach of ten criteria that can be applied to the simulator to describe it in a measurable and comparable manner.

In order to test how efficient the suggested approach is, we apply it on Cisco Packet Trace and GNS3, which are general-purpose network simulators. The application of the approach proves that it does not only highlight general aspects of the simulators behaviors but it shows their disadvantages as well.

In a future study, we plan to apply the approach to compare other network simulators and include other measurable criteria, such as scalability. We also are working on extending the proposed approach to consider WSN simulators, by involving special items describing the determinants of these networks, such as power constraints, models for energy consumption, and power harvesting.

REFERENCES

[1] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai ,“A Survey of Peer-to-Peer Network Simulators,” *in Proc. of The Seventh Annual Postgraduate Symposium*, 2006, vol. 2.

[2] M. Imran, A. M. Said, and H. Hasbullah, “A survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks,” *in Proc. of The International Symposium in*

Information Technology (ITSim), 2010, vol. 2, pp. 897–902. doi: 10.1109/ITSIM.2010.5561571.

[3] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya. (2011). Wireless Sensor Network Simulators a Survey and Comparisons. *International Journal of Computer Networks*. 2(5). pp. 249–265

[4] D. Kotz, C. Newport, S. Robert, J. Liu, Y. Yuan, and C. Elliott, "Experimental Evaluation of Wireless Simulation Assumptions," *in Proc.of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004, pp.78-82. doi: 10.1145/1023663.1023679

[5] Jesin A, *Packet Tracer Network Simulator*, 1st ed. Packt Publishing Ltd, 2014, ch. 1, pp. 2-3.

[6] A. Smith, and C. Bluck, "Multiuser Collaborative Practical Learning Using Packet Tracer," *In 2010 Sixth International Conference on Networking and Services*, 2010, pp. 356-362. doi: 10.1109/ICNS.2010.56.

[7] C. Welsh, *GNS3 Network Simulation Guide*, 1st ed. Packt Publishing Ltd, 2013, ch. 1, pp. 5-6.

[8] W. Makasiranondh, S. P. Maj, and D. Veal. (2010). Pedagogical Evaluation of Simulation Tools Usage in Network Technology Education. *Engineering and Technology*. 8(3). pp. 321-326.

[9] L. Faxun. (2010). The Application of GNS3 in Network Experiments. *Engineering and Technology*. 10. pp. 32-40.

[10] S. Bajaj, L. Breslau,D. Estrin,K. Fall, S. Floyd, P. Haldar, and S. Kumar (1998). Virtual Internetwork Testbed: Status and Research Agenda. *Technical Report 98-678*, University of Southern California.

[11] H. Y. Tyan, "Design, Realization and Evaluation of a Component-based Compositional Software Architecture for Network Simulation," Ph.D. dissertation, The Ohio State University, 2002.

[12] K. Wehrle, M. Günes and G. Gross, *Modeling and Tools for Network Simulation*, 1st ed. Springer Science & Business, 2010.

[13] M. Jevtić, N. Zogović, and G. Dimić, "Evaluation of Wireless Sensor Network Simulators," *In Proc. of the 17th telecommunications forum (TELFOR 2009)*, 2009, pp.1303-1306.

[14] A. R. Khan, S. M. Bilal, and M. Othman, "A Performance Comparison of Open Source Network Simulators for Wireless Networks," *2012 IEEE international conference on Control System, Computing and Engineering (ICCSCE)*, 2012, pp. 34–38. doi: 10.1109/ICCSCE.2012.6487111.

[15] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus, "Comparative Study of Wireless Network Simulators," *In Seventh International Conference on Networking (icn 2008)*, 2008, pp.517-523. doi: 10.1109/ICN.2008.97.

[16] G. F. Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M. J. Reed (2003). Opnet Modeler and NS-2: Comparing the Accuracy of Network Simulators for Packet-level Analysis Using a Network Testbed. *WSEAS Transactions on Computers*. 2(3). pp. 700-707.

[17] E. Weingartner, H. Vom Lehn, and K. Vom Lehn, "A Performance Comparison of Recent Network Simulators," *In 2009 IEEE International Conference on Communications*, 2009, pp. 1-5. doi: 10.1109/ICC.2009.5198657.

[18] L. Valle-Rosado, L. Narváez-Díaz, C. González-Segura, and V.Chi-Pech. (2012). Design and Simulation of an IPv6 Network Using Two Transition Mechanisms. *International Journal of Computer Science Issues*. 9(6). pp. 60.

[19] L. M. I. Sari, P. Hatta, E. S. Wihidayat, and F. E. N. G. Xiao. (2018). A Comparison between the Use of Cisco Packet Tracer and Graphical Network Simulator 3 as Learning Media on Students' Achievement. *Journal Pendidikan Teknologi dan Kejuruan*. 24(1). pp. 132-136. doi: 10.21831/jptk.v24i1.16042.

[20] L. sun, J. Wu, and H. Yin, "Comparison between Physical Devices and Simulator Software for Cisco Network Technology Teaching," *In 2013 8th International Conference on Computer Science & Education*, 2013, pp. 1357-1360. doi: 10.1109/ICCSE.2013.6554134.

[21] T. S. Chou, S. Baker, and M. Vega-Herrera, "A Comparison of Network Simulation and Emulation Virtualization Tools," *In Proc. ASEE Annu. Conf. Expo*., 2016, pp. 1-9.

[22] C. Wang, and J. Y. Chen, "Implementation of GRE over IPsec VPN Enterprise Network Based on Cisco Packet Tracer," *In 2nd International Conference on Soft Computing in Information Communication Technology*, 2014. doi: 10.2991/scict-14.2014.34.

[23] C. Archana. (2015). Analysis of RIPv2, OSPF, EIGRP Configuration on Router Using CISCO Packet Tracer. International journal of Engineering science and innovative technology. 4. pp. 215-222.

[24] K. Dangwal and V. Kumar (2014). Comparative Study of EIGRP and RIP using CISCO Packet Tracer. *International Journal of Engineering Sciences & Emerging Technologies*. 6(6). pp. 475-480.

[25] M. M. N. Noor, N. Yayao, and S. Sulaiman (2018). Effectiveness of Using Cisco Packet Tracer as a Learning Tool: A Case Study of Routing Protocol. International *Journal of Information and Education Technology*. 8(1). pp. 11-16. doi: 10.18178/ijiet.2018.8.1.1004.

[26] C. Wijaya, "Performance Analysis of Dynamic Routing Protocol EIGRP and OSPF in IPv4 and IPv6 Networks," *in 2011 First International Conference on Informatics and Computational Intelligence*, 2011, pp. 355-360. doi: 10.1109/ICI.2011.64.

[27] N. H. Prasad, B. K. Reddy, B. Amarnath, and M. Puthanial. (2016). Intervlan Routing and Various Configurations on Vlan in a Network Using Cisco Packet Tracer 6.2. *International Journal for Innovative Research in Science and Technology*. 2(11). pp.749-758.

[28] D. C. Frezzo, J. T. Behrens, R. J. Mislevy, P. West and K. E. DiCerbo, "Psychometric and Evidentiary Approaches to Simulation Assessment in Packet Tracer Software," *in 2009 Fifth International Conference on Networking and Services*, 2009, pp.555-560. doi:10.1109/ICNS.2009.89.

[29] W. Makasiranondh, S. P. Maj, and D. Veal (2010). Pedagogical Evaluation of Simulation Tools Usage in Network Technology Education. World. *Transactions on Engineering and Technology Education*. 8(3). pp. 321-326.

[30] P. Gil, G. J. Garcia, A. Delgado, R. M. Medina, A. Calderon, and P. Marti, "Computer Networks Virtualization with GNS3: Evaluating a Solution to Optimize Resources and Achieve a Distance Learning," *in 2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 2014, pp. 1-4. doi:10.1109/FIE.2014.7044343.

[31] G. Y. A Y. Al-Gadi, A. Amin Babiker, N. Mustafa, and M. Hamied (2014). Evaluation and Comparisons of Migration Techniques from IPv4 to IPv6 Using GNS3 Simulator. *Journal of Engineering*. 4(8). pp. 51-57.

[32] D. R. Al-Ani, A. R. Al-Ani, (2018). The Performance of IPv4 and IPv6 in Terms of Routing Protocols Using GNS 3 Simulator. 4(8). pp. 1051-1056. doi:10.1016/j.procs.2018.04.147

[33] T. Ravi, H. Kumar, A. Dumka, and A. Anand, "Traffic Management in MPLS Network Using GNS simulator

Using Class for Different Services," in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 1066-1068
[34] W. N. A. W. Ali, A. H. M. Taib, N. M. Hussin, and J. Othman, "IPv6 Attack Scenarios Testbed," *in 2012 IEEE Symposium on Humanities, Science and Engineering Research*, , 2012, pp. 927-932. doi:10.1109/SHUSER.2012.6269008
[35] A. Al-Mahrouqi, P. Tobin, S. Abdalla, and T. Kechadi, "Simulating SQL-Injection Cyber-attacks Using GNS3," *in the 6th International Conference on Computer Modeling and Simulation (ICCMS 2015)*, 2015, doi:10.7763/IJCTE.2016.V8.1046.
[36] J. Nikoukaran, V. Hlupic, and R. J. Paul. (1999). A Hierarchical Framework for Evaluating Simulation Software. *Simulation Practice and Theory*. 7(3). pp. 219-231. doi:10.1016/S0928-4869(98)00028-7
[37] S. Robinson. (2008). Conceptual Modelling for Simulation Part I: Definition and Requirements. *Journal of the Operational Research Society*. 59(3). pp. 278-290. doi:10.1057/palgrave.jors.2602368.
[38] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling," John Wiley & Sons, 1990, ch. 24, pp. 418.
[39] M. Mekni and B. Moulin, "A Survey on Sensor Webs Simulation Tools," *in 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, 2008, pp. 574-579. doi: 10.1109/SENSORCOMM.2008.13
[40] K. Romer and F. Mattern, "The Design Space of Wireless Sensor Networks," *in IEEE Wireless Communications*, vol. 11, no. 6, pp. 54-61, Dec. 2004. doi: 10.1109/MWC.2004.1368897.
[41] Socolofsky, T. and C. Kale, "TCP/IP Tutorial", RFC 1180, January 1991. doi 10.17487/RFC1180.
[42] L. A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *in Computer*, vol. 40, no. 12, pp. 33-37, Dec. 2007. doi: 10.1109/MC.2007.443
[43] N. Nurseitov, M. PAULSON, and R. REYNOLDS.(2009) "Comparison of JSON and XML Data Interchange Formats: a Case Study". *Caine*. 9. p. 157-162.


Michel Bakni was born in Lattakia, Syria, in 1989. He received the B.S. degree in telecommunication and electronics from Tishreen University, Lattakia, in 2013 and the M.S. degree from the University of Technology of Belfort-Montbéliard (UTBM), France, in 2017, in mobile and distributed networks. He is currently pursuing the Ph.D. degree with the Doctoral School of the University of Bordeaux (UBx) and at ESTIA, a superior engineering School for Advanced Industrial Technologies. His research interests include Simulation, Wireless Sensor Networks, and Energy consumption optimization.


Yudith Cardinale is a Full Professor in Computer Science Department at Universidad Simón Bolívar (USB) since 1996. She graduated with honors in Computer Engineering in 1990 at Universidad Centro-Occidental Lisandro Alvarado, Venezuela. She received her M.Sc. Degree and Ph.D. in Computer Science from USB, Venezuela, in 1993 and 2004 respectively. Her research interests include parallel processing, distributed processing, operating systems, high performance on grid and cloud platforms, and web services composition, including semantic web. She is the Director of the Parallel and Distributed Systems Research Group (GRyDs) at USB and coordinates several national and international research projects. She has written a huge range of publications in areas such as parallel computing, grid computing, parallel check pointing, collaborative frameworks, and Semantic Web. Her home page is http://www.ldc.usb.ve/en/~yudith.


Luis Manuel Moreno graduated in Telecommunications Engineering at Universidad Simon Bolivar, Venezuela, in 2019. His main areas of research interest are operating systems, distributed systems and embedded systems.