

METHODOLOGY TO EVALUATE WSN SIMULATORS: Focusing on Energy Consumption Awareness

Michel Bakni¹, Luis Manuel Moreno Chacón², Yudith Cardinale², Guillaume Terrasson¹, and Octavian Curea¹

¹Univ. Bordeaux, ESTIA Institute of Technology, F-64210 Bidart, France

²Universidad Simón Bolívar, Caracas, 1080-A, Venezuela

Abstract. *Nowadays, there exists a large number of available network simulators, that differ in their design, goals, and characteristics. Users who have to decide which simulator is the most appropriate for their particular requirements, are today lost, faced with a panoply of disparate and diverse simulators. Hence, it is obvious the need for establishing guidelines that support users in the tasks of selecting and customizing a simulator to suit their preferences and needs. In previous works, we proposed a generic and novel methodological approach to evaluate network simulators, considering a set of qualitative and quantitative criteria. However, it lacks criteria related to Wireless Sensor Networks (WSN). Thus, the aim of this work is three fold: (i) extend the previous proposed methodology to include the evaluation of WSN simulators, such as energy consumption modelling and scalability; (ii) elaborate a study of the state of the art of WSN simulators, with the intention of identifying the most used and cited in scientific articles; and (iii) demonstrate the suitability of our novel methodology by evaluating and comparing three of the most cited simulators. Our novel methodology provides researchers with an evaluation tool that can be used to describe and compare WSN simulators in order to select the most appropriate one for a given scenario.*

Keywords: *Methodology, Simulators, Wireless Sensors Networks, Energy Consumption.*

1 INTRODUCTION

In computer networks, network simulation is one of the most used and powerful evaluation methodologies. It has been used for the design and development of communication architectures and network protocols, as well as for verifying, managing, and predicting their behaviors. Since Wireless Sensor Networks (WSNs) provide a mean to capture and understand the reality and to interact on response of the gathered data [39], they have gained attraction in the research domains. Thus, simulators are also useful tools to evaluate WSNs.

In the last decades, several simulators have been either extended to include WSNs or built as WSNs simulators from the beginning. Different research groups develop different simulators according to their needs. For example, some simulators are designed to simulate the entire system. These simulators focus on the scalability, thus, their performance is a cornerstone in this regard. Others concern with the structure of the node and its energy consumption. For these simulators, wireless propagation and energy consumption modelling is what attracts attention.

Therefore, users who have to decide which simulator is the most appropriate for their particular requirements, are today lost, faced with this panoply of disparate

simulators. Thus, it is obvious the need of establishing guidelines and a systematic approach that support users in the tasks of selecting and customizing a simulator to suit their preferences and needs. To support this decision making, some works have tried to evaluate and compare several simulators and shyly propose some guidelines and steps to carry out such systematic evaluation [5][7][11][12][14][19][21][24][28][30][32][33][35][38]. However, as far as we know, there is not a generic methodological process to evaluate and compare network simulators that can be applied independently of their types and simulation scenarios.

In our previous works [2][3], a generic and novel methodological approach to evaluate network simulators is proposed. This approach is based on a set of qualitative and quantitative criteria. Even though, this methodological approach is generic and efficient to evaluate network simulators, it still lacks criteria related to WSN. Thus, the aim of our work, presented in this paper is threefold: (i) **extend the innovative and generic methodological approach** to include the evaluation of characteristics of WSN simulators, such as scalability and energy consumption awareness; (ii) **elaborate a study of the state of the art of WSN simulators**, with the intention of identifying the most used and cited in scientific articles; and (iii) **demonstrate the suitability of our innovative methodology** by evaluating and comparing several of the most cited WSN simulators.

The application of our methodological approach to evaluate three of the most cited simulators leads to results that are measurable and comparable. It allows a comprehensive overview of simulators features, advantages, and disadvantages. Therefore, this generic methodological approach provides researchers with an evaluation tool that can be used to describe and compare WSN simulators in order to select the most appropriate one for a given scenario.

2 RELATED WORK

The flexibility and validation in model construction offered by network simulation has fostered the research and development of multiple and different simulators. Thus, in order to select an appropriate network simulator, it is important to have good knowledge about their strengths and weaknesses, as well as to know how reliable are the models used by the simulators. In particular, for WSNs it is important to evaluate the scalability and energy consumption awareness of simulators.

To support this selection process, some works have proposed comparative criteria to carry out the evaluation of network simulators. For WSNs, the most recent and cited comparative studies are [5][7][11][12][14][19][21][24][28][30][32][33][35][38]. Most of them propose generic comparative qualitative criteria, such as type of simulator, API, languages supported, platforms supported, licenses, network support type, user interface [7][11][24][28][30][38]. Only the works proposed in [30][32] consider quantitative criteria, such as CPU utilization, memory usage, execution time, and scalability. Other studies also consider energy consumption modelling (e.g., wireless propagation, power consumption, battery, topology, antenna, radio

propagation, noise, and application modelling) and the challenges that face their implementations [5][12][14][21]. Few of such works are dedicated to evaluate WSN simulators in function of the energy consumption of each component of the WSN nodes and how they model the energy consumption of each component [19][35].

All these works, mainly evaluate WSN simulators based on a set of qualitative criteria, related and not related to scalability and energy consumption, but they do not establish any methodological process to perform the evaluation. We consider most of the same generic qualitative criteria, plus quantitative criteria, to evaluate any type of network simulators, as well as specific criteria to evaluate WSN simulators, such as scalability and energy consumption awareness. Besides, we propose a methodological approach to make such evaluation in a systematic and formal way.

The work proposed in [33], timidly proposes a methodology. However, the proposed guidelines and steps are focused on performing the network simulation, by following these steps: (i) evaluate the simulator based on a set of generic criteria (e.g., general features, visual support, flexibility, user support); (ii) select benchmarks to evaluate the simulated scenarios (e.g., network design, network protocols); (iii) conduct the simulation process; (iv) evaluate and analyze results. This methodology is focused on how to perform the simulation process; while our methodology, besides of considering such aspect, is intended to be generic, flexible, and suitable to support the selection of the most appropriate network simulator for a target simulation scenario, according to the user preferences and requirements.

3 WSN SIMULATORS: STATE-OF-THE-ART

We present a study of WSN simulators that are used in current researches.

3.1 Systematic Review

In order to find, select, and analyze the most popular and recent WSN simulators, we have followed a systematic review consisting of three main steps: (i) search of works dealing with WSN simulators; (ii) selection of relevant articles; and (iii) statistically find simulators cited in the set of the selected papers.

For the first step, the search was done on the Google Scholar search engine, which provides links to scientific repositories such as IEEE Xplore, ACM, and Springer. The search was based on tags that included the keys *WSN* and *simulator*, combined with tags related to the focus of the papers, such as *Survey*, *Review*, *Comparison*, *Evaluation*. We obtained more than 50 scientific articles.

For the second step, we select the most relevant articles related to WSN simulators evaluation, proposal, and comparison. From the more than 50 scientific papers obtained in the first step, some of them do not focus on simulators, but on designing and evaluating WSNs. We select works from 2010 and some older ones that have been widely cited. The final result was 37 relevant papers, categorized according to their main focus: (i) comparison papers, that evaluate and compare different

simulators; (ii) survey papers, in which authors present a general review of WSN simulators; (iii) simulator specific papers, which introduce the design or features of a particular WSN simulator; and (iv) trend papers, which contain explanation of the definitions and trends of how researchers evaluate WSN simulators.

In the third step, we analyze the selected papers and present statics of referenced WSN simulators on each category.

3.2 Categories of scientific articles

The selected papers were classified in four groups:

Comparison papers, which include comparative studies of WSN simulators, based on self-defined criteria that evaluate on each simulator, in order to analyze the differences among them [5][7][11][12][14][19] [21][24][28][30][32][33][35][38].

In [11], authors make a review of some of the open source network simulators (i.e., NS2, NS3, OMNeT++, and JSIM), comparing them according to languages supported, platforms supported, licenses, network support type, user interface, and API. In [30], authors compare NS2, NS3, OMNeT++, and GloMoSim. A unified scenario is applied by simulating a MANET routing protocol, in order to measure memory usage, computational time, and scalability, from which NS3 demonstrates the best performance. Similarly, in [7][21][28][32][33][38], some of popular WSN simulators (NS2, NS3, TOSSIM, OMNeT++, JSIM, Castalia, QualNet, EmStar, ATEMU, Avrora, SENS, COOJA, etc.) are described and compared based on the their general characteristics, their merits, and their limitations. The studies presented in [5][24], evaluate more than 20 simulators.

In [12], authors make a survey of available tools to evaluate WSN applications. They identify a set of models that are necessary to have in a WSN simulator: wireless propagation model, fine-grained energy expenditure model, non-linear battery model, and application model. In [14], authors compare Castalia, TOSSIM, and NS3 based on the sustainability to test dynamic network reconfiguration protocols. One of the topics that they evaluate is the energy consumption model of the simulators. They identify that the ability to model the RF states of the sensors is important to model the energy of sensors. In [35], authors compare Castalia, MiXiM, TOSSIM, and WSNnet, based on topology, antenna, radio propagation, noise, RF, medium access control, and energy consumption modelling. They execute a series of real experiments and calibrate the radio propagation model and the energy consumption model. In [19], authors compare NS2, NS3, TOSSIM, and OMNeT++, focusing on the modelling of the energy consumption. They describe the energy consumption of each component of the WSN nodes and show how the studied simulators model the energy consumption of each component.

Survey papers, that describe WSN simulators in a general way, but there is no comparison among them [1][9][10][15][16][22][36][40]. In [29], a review of network modelling and simulation tools is presented, including WSN simulators, such as

NS2, OPNET, and GloMoSim. Authors in [17] present a review of several WSN simulation tools. They mostly focus on their suitability for large-scale WSNs.

Simulator-specific papers, which focus on describing new WSN simulators[13][18][20][23]. In [23], the support for heterogeneous networks in IDEA1, is presented. In [13], WebShawn, an online WSN simulator is presented. A4WSN, an architecture-driven modelling platform for analysing and developing WSNs is presented in [18]. NS2 is the simulator used in [18] to analyse S-MAC and leach in WSNs.

Trends papers focused on studying proposed approaches to evaluate WSN simulators and research trends. In [26], authors compile a large set of papers of wireless communication-related conferences and review the statistics about the tools (i.e., testbeds and simulators) the researchers use to evaluate their experiments. Additionally, they address the issues and challenges facing the proper use of WSN simulators. They assert that simulators do not reproduce actual environmental conditions of deployed systems, thus experimental testbeds can be developed to replace simulators. In [6], authors discuss topics to consider when addressing IoT issues. They present the research trends on IoT simulators in the last years. To achieve that, authors describe existing tools that are used by researchers to prove and evaluate their findings on IoT research. They claim that more work is needed to conduct large-scale, robust and effective IoT simulation, and prototype evaluations.

3.3 Statistical Analysis of Selected Papers

In total, in the selected papers there are 369 citations, distributed among more than 100 WSN simulators. According to the number of citations, simulators are categorized into three groups: (i) Group 1, composed by simulators with more than 12 citations; this group includes 7 simulators; (ii) Group 2, involves all simulators with 6 to 12 citations; it contains 12 simulators; and (iii) Group 3, covers all simulators that are cited less than 6 times; it contains 87 simulators. Figure 1 presents the number of citations and the number of simulators of each group.

The total sum of citations that have the simulators of the Group 1 is 125, which represent 33.88% of the citations distributed in 7 simulators. Group 2 has in total 117 citations, which means the 31.71% of the citations. Group 3 has 127 citations, which represents a 34.42% of the total of citations. Figure 1 shows that Groups 1 and 3 contain more citations than Group 2. In the case of the Group 3, those citations are distributed in a larger number of simulators.

Figure 2 shows the number of citations of the simulators of Group 1, in which the most cited simulators are NS2, TOSSIM, and OMNeT++. NS2 and TOSSIM are presented in 24 papers, and OMNeT++ is presented in 20 papers.

This study can help to identify the most used WSN simulators, but the number of citation is not enough to provide comparison-based view. Therefore, a more robust approach to compare and evaluate WSN simulators is needed. In next sections, our methodology is presented. Then, in order to show its applicability and suitability, the most cited simulators are evaluated and compared.

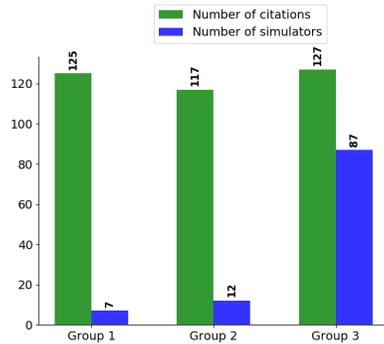


Fig. 1. Citations of WSN simulators

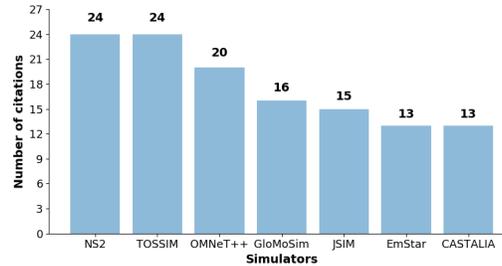


Fig. 2. Most cited simulators.

4 METHODOLOGY TO EVALUATE WSN SIMULATORS: OUR PROPOSAL

The methodology proposed in our previous works [2][3] does not include qualitative criteria to analyze the simulator scalability and the support of simulators on evaluating traces of energy consumption, sensor nodes mobility, and wireless medium modelling. These characteristics are present in WSN and are less important for general networks. Therefore, considering features that characterize WSN simulators, we propose an extension of the previously proposed methodology.

4.1 Methodological process

The methodological process consists on the following steps:

Step 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 a word or number, while quantitative criteria need to be measured.

Step 2. Establish the experimental setup. The platform in which simulators are installed to be evaluated should not be neglected. Using a specific simulator, the way the operating systems manage system resources and the produced overhead have an important impact on the performance and behavior of such simulator. Hence, it is worthy to install the simulators on different systems (e.g., Windows, Linux, MacOS) under the same architecture.

Step 3. Evaluate the qualitative criteria of the simulator(s). To comply this step, it is recommended to revise the available documentation of simulator(s) and elaborate a table highlighting their characteristics.

Step 4. Design a test scenario to evaluate the measurable criteria. In a data network, a scenario is defined by a set of parameters that characterize a specific use case. According to the protocols that are intended to evaluate, it is important to decide the network elements to be simulated, the number and the type of experiments, as well as the time of the simulation, taking into account the

criteria to be evaluated.

Step 5. Evaluate the measurable criteria of the simulator(s) by executing the designed experiments. In order to obtain the results, each designed scenario has to be implemented on the simulator(s). In order to facilitate the analysis and comparison (if it is the case), results must be presented in tables and graphics.

Step 6. Elaborate a discussion by analyzing the results.

With this six-steps methodology, users can evaluate network simulators to select the most appropriated according to their needs and scenarios. For the comparative analysis, we also propose a set of criteria, which complement the Step 1.

4.2 Criteria used in the methodology

The set of criteria used in the methodology are as follows:

1. Nature of the simulator: The nature of the simulator is an assessment of how the simulation is performed. Precisely, the term *simulation* means that the simulated process is programmed and only the software aspect is involved in the simulation. But if the word *emulation* is used, the hardware is also involved in the simulation process [31].

2. Type of simulator: Network simulators are based on two philosophies, discrete-event simulator or trace-driven one. In the first case, 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 until the end of the simulation. This philosophy is highly used in WSNs, since it allows to easily simulate hundreds of jobs running on different WSN nodes [40]. The trace-driven simulation mostly plays a crucial role in real time applications. It provides information which lets users to have more detailed knowledge of the simulation model [24].

3. License: From a legal aspect, simulators can be private property or they can be used under a free or public agreement.

4. User interface: It is an evaluation of how a user can interact with the simulator. This criterion includes two aspects: (i) **Graphical User Interface (GUI):** Is it an integral part of the simulator? What is the level of details it can show?; and (ii) **Supported programming languages:** Can users interact with the simulator by programming scripts? Can users develop a piece of software to communicate with the simulator?

5. Supported platforms: It is the characterization of the usability of the simulator source code on different platforms and operating systems.

6. Heterogeneity: It is an evaluation of the ability to simulate heterogeneous systems where different types of nodes can exist in the same scenario.

7. Level of details: It consists on evaluating the level of aspects that are being simulated. Sorted in descending order, they are: abstract algorithms, high level protocols, low-level protocols, and hardware. The lower the level is, the less the

assumptions is and the more the constraints are.

8. Modelling: It represents the ability to modify existing models in the simulator or to implement and test new ones.

9. Mobility modelling: It indicates the support of the WSN simulators for modelling mobility of sensor nodes.

10. Wireless medium model: It means the ability to model the wireless medium used to estimate the radio signal strength, quality, and delay between transmitter and receiver units.

11. Energy consumption modelling: It refers to the current draw at the sensor node level. A node sensor is typically composed by four major components: sensing unit, the power supply unit, the processing unit, and the communication unit. Units that consume energy are the sensing, the processing, and the communication units; while the power supply unit, which can embed energy harvesting solutions, provides energy. The aspects considered to evaluate this criterion are: (i) **Battery modelling:** which consists on evaluating if non-linear or linear battery models are implemented in the simulator; (ii) **RF states modelling:** that represents the capacity to consider and simulate all RF states, such as Idle, Sleep, Receiving, and Transmitting; and (iii) **Limitations:** to address the constraints of simulators regarding power consumption modelling. With these parameters, a scenario can be designed to evaluate energy consumption, such as total power consumption or the energy consumption on each RF state.

12. Supported technology and protocols: To evaluate the support provided for the protocols, TCP/IP model is used [34].

13. Measurable criteria: The main purpose of the measurable criteria is to provide a general idea of the effectiveness of the simulator in terms of the consumption of available resources, scalability, and energy consumption modelling capacities. Our methodological approach includes four factors for the simulator performance study, for both network and WSN simulators: (i) **CPU Utilization:** it is a measure of the simulator performance [4], which consists in the percentage of time spent performing the simulation process of the total processing time [25]; (ii) **Execution time:** it is the time needed to complete a simulated scenario; measured in seconds; (iii) **Memory usage:** it is the amount of memory used by the simulator, measured in bytes; and (iv) **Scalability:** it is the measure of the capacity of the simulator of simulating huge scenarios; how many network or WSN components can be simultaneously simulated without degrading the simulator performance?

All these measurable aspects evaluate the simulator(s) performance. However, for WSN simulators there are other measurable aspects that evaluate their capacity of modelling energy consumption of WSN nodes. Hence, this measurable criterion is considered only for WSN simulators: **Trace of nodes energy consumption:** it represents the obtained measures of energy consumption of WSN components, ac-

ording to the modelling allowed by the evaluated simulator(s). This criterion does not measure the performance of the simulator itself, but its capacity of measuring the energy consumption of WSN nodes.

In [3], we apply the methodology to evaluate Packet Tracer simulator. In [2], the goal was to compare data network simulators on a standard base, thus, the selection of one simulator over the others can be justified. The approach was applied on GNS3, a data network emulator, as well as on Packet Tracer. In this work, in order to show how to apply this extended methodological approach for WSN simulators, we evaluate and compare three of the most cited ones.

5 APPLICATION OF THE PROPOSED METHODOLOGY

To show the suitability and flexibility of the extended methodology, we apply it to evaluate and compare the most cited WSN simulators, identified in Section 3: NS2, TOSSIM, and OMNeT++. NS2 is a discrete event network simulator, initially designed to simulate wireless LAN protocols, though later was expanded to mobile ad-hoc networks support. TOSSIM, stands for Tiny OS simulator, is a discrete-event simulator for TinyOS applications. TOSSIM is a TinyOS emulator, but not a general WSN simulator [1]. OMNeT++ (for Objective Modular Network Testbed) is a discrete event simulator based on C++. In this work, OMNeT++ is used with INET, which is an OMNeT++ framework that has implemented models for wired, wireless, and mobile networks. In the following, we explain how the methodological process was applied to evaluate these three WSN simulators.

Step 1: Establish a set of criteria. As it is illustrated in Table 1, the set of criteria considered are the ones described in Section 4.

Step 2: Establish the experiment setup. To evaluate the considered simulators in different systems, they are tested on 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 of RAM, 915 GB of disk allocated for Linux, while 909 GB is allocated for Windows.

Step 3: Evaluate the qualitative criteria. The qualitative criteria of the three simulators are summarized in Table 1. We particularly comment about energy consumption modelling of each simulator. TOSSIM does not have a model of the energy consumption. However, PowerTOSSIM z [27], an extension of TOSSIM, adds that functionality. The energy consumption model of PowerTOSSIM z considers four hardware components that consume energy: microprocessor, LED, RF module, and memory. Additionally, PowerTOSSIM z models modern batteries, by simulating their nonlinear behavior, expressed by the effects of rate capacity and recovery capacity [14]. PowerTOSSIM z has disadvantages as well: (i) it is a plug-in to TOSSIM, i.e., PowerTOSSIM z cannot dynamically change the fixed power-consumption parameters during the simulation [14]; (ii) the energy consumption caused by power-state transitions and the time needed for that, are not taken into

account; (iii) PowerTOSSIM z cannot simulate energy harvester units; and (iv) this plugin has a poor level of support. The project is not longer maintained.

The energy consumption model in NS2 maintains a total value for the energy stocked at each node in the WSN. When the energy of a node is completely consumed, the model declares it as *dead*. The model considers only the consumption of the RF module, it is based on a state machine that has the following states for the RF model: Idle, Sleep, Receiving, and Transmitting. NS2 cannot simulate non-linear batteries. The batteries of the sensor nodes are ideal batteries. Moreover, NS2 energy model does not support energy harvesting or battery recharging. It monitors the changes in the power level in one way only, i.e., the consumption.

The energy modelling in INET is divided in 3 parts: energy consumption models, energy generation models, and energy storage models. These models can represent the energy in two different ways: charge and current (CC) or energy and power (EP). Energy consumer models describe the energy consumption of units and node components over time. There are three models of energy consumption. Two of them are based on the RF states and they differ in the way the energy is represented (EP or CC). The third model is a basic model that represents the energy consumption of a node with two states only (Idle and Active). It is used to have a general overview of the energy consumption without focusing on the details.

Regarding the mobility modelling, NS2 and OMNeT++/INET support simulations for mobile nodes. TOSSIM does not have this feature. However, an extended plug-in for TOSSIM can be added to provide the support for the mobility, it is called MOB-TOSSIM [8].

Step 4: Design the test scenarios. Basic scenarios are designed to evaluate the performance of the selected simulators and their energy consumption modelling capacity. The performance is measured in terms of CPU utilization, memory usage, execution time, and scalability. A meshed topology is adopted for the WSN, whose size is increasing exponentially for different tests. The basic component (BC) of the topology consists of four sensor nodes, each one placed in the vertex of a 10 meters x 10 meters square. The first test includes only one BC with four sensor nodes. The second test is done with two BC, i.e., eight nodes. The third one is composed by four BC, with 16 nodes, and so on. In total, eight simulations take place on each system (Linux and Windows), with the number of BCs changing as: 1, 2, 4, 8, 16, 32, 64, and 128 for each simulator.

Each node in the WSN is configured to use IPv4 and ICMPv4. The goal is to create a data message with an echo request to all other nodes in the topology. A node that receives the echo request, replies back the same message. Each simulation lasts 100 seconds. The frequency is 1 Hz, which means that one echo message is sent every second. As a result, there are 100 echo request messages sent per simulation.

To evaluate the energy consumption models, another test scenario is proposed. This scenario consists of two nodes, which are 10 meters apart from each other.

Table 1. Comparison of WSN simulators using the proposed criteria

Criterion	TOSSIM	NS2	OMNeT++/INET
Nature of the simulator	Emulator	Simulator	Simulator
Type of the simulator	discrete-event	discrete-event	discrete-event
License	BSD-license	GNU GPLv2 license	Academic Public License. INET models are licensed under LGPL or GPL.
User interface	GUI: Yes, through TinyViz. Supported programming language: Python, C++ and NesC	GUI: Yes, through Nam. Supported programming language: C++ and OTcl	GUI: Yes, a built-in GUI interface is available Supported programming language: C++ and NED
Supported platforms	Linux and Windows	Linux, MacOS and FreeBSD	Windows, Linux and Mac OSX
Heterogeneity	No	No	Yes
Level of details	Code level	Packet Level	Packet level
Modelling	Available	Available	Available
Mobility model	Yes, through MOB-TOSSIM	Yes	Yes
Wireless medium model	Path loss models: lognormal shadowing Other models: noise modelling	Path loss models: shadowing, ray ground, free space	Path loss models: free-space, log-normal shadowing, rayleigh fading, 2-ray ground, rician fading, nakagami fading Other models: Background noise, obstacle loss and propagation models
Energy model	Battery model: No RF states: Yes Limitations: Cannot model energy harvester units	Battery model: Only for Ideal Battery RF states: Yes Limitations: Cannot model sensing and processing units	Battery model: Yes RF states: Yes Limitations: Cannot model sensing and processing units
Supported technology and protocols	TOSSIM simulates entire TinyOS applications, including the network stack that supports TinyOS implementation.	Application Layer: DHCP, telnet, FTP, HTTP Transport Layer: TCP, UDP, SCTP, XCP, TFRC, RAP, RTPM Network Layer: IPv4, IPv6 Link Layer: HDLC, GAF, MPLS, LDP, Diffserv, DropTail, RED, RIO, IGMPv2, IGMPv3, WFQ, SRR, Semantic Packet Queue, REM, CSMA, 802.11b, 802.15.4, Satellite Aloha Routing Protocols: RIP, AODV, Click, DSDV, DSR, NixVectorRouting, OLSR	Application Layer: HTTP, DHCP, BitTorrent, P2P, Video Streaming, Voice Transport Layer: TCP, UDP, SCTP, RTP, RTCP. Network Layer: ARP, HIP, IPv4, IPv6, MIPv6 Link Layer: 802.11, 802.11p, 802.1e, WiMAX, Routing Protocols: AODV, BGP, GPSR, link-state routing, OSPF, OSPFv2, PIM, RIP

Table 2. Parameters of the energy consumption scenario

Parameter	802.11b	802.15.4
Bitrate	11 Mbps	250 Kbps
MAC	CSMA/CA with RTS/CTS CSMA/CA with CCA	
Transmitting power	750 [mW]	52 [mW]
Receiving power	220 [mW]	59 [mW]
Sleep power	0.2 [mW]	0.06 [mW]
Idle power	0.2 [mW]	0.06 [mW]

One of the nodes is periodically sending an ICMPv4 echo request to the other node. When the other node receives the request, it replies back the same message. The echo request and the echo reply are identical in length and format. Therefore, the energy consumption of both nodes will be the same. The communication of nodes is made using two different wireless link protocols: 802.11b and 802.15.4. For each protocol the payload length of the ping message starts at 10 bytes, then, it is gradually being increased by 10 bytes, until the payload size reaches 90 bytes. In total, there are 9 simulation per protocol. Each simulation is repeated 3 times for different values of frequency of the ping messages: 0.1, 1, and 2 Hz.

For the 802.11b scenarios, the energy consumption parameters were taken from the data sheet of HDG204 RF Module¹, while for the 802.15.4 scenario was used the data sheet of CC2420 RF Module². Each simulator was configured to use the models of the protocols with the values of the standards. The values of the energy consumption for each module is shown in Table 2.

Step 5: Evaluate the measurable criteria. NS2 is only evaluated in Linux, since it is the only platform that supports its installation. OMNeT++/INET is installed on both Windows and Linux. The NS2 version used is the 2.35³, for OMNeT++, it is 5.4.1⁴, and for INET, it is 4.1.0-810053f713⁵. TOSSIM (PowerTOSSIM z) is not installed in none of the systems, since it has a poor level of support for the recent OS versions and it is not possible to install the simulator on the systems used. Thus, PowerTOSSIM z is not evaluated in terms of measurable criteria.

Performance scenarios: In the performance scenario, the CPU utilization is evaluated for the simulators during 100 seconds of simulation. The results of the performance evaluation of the CPU utilization for different BCs are shown in Figure 3. NS2 tends to consume all available CPU cycles, whatever the number of the BCs is, while OMnet++ consumes the CPU differently in Linux than in Windows. Figure 3 shows that the CPU utilization in Windows is always less than Linux when the same scenario is implemented. In both OSs, as the number of BCs increases, the average value of CPU utilization increases as well.

¹ <https://media.digikey.com/pdf/DataSheets/H&DWireless0PDFs/HDG204DS.pdf>

² <http://www.ti.com/lit/ds/swrs041c/swrs041c.pdf>

³ <https://sourceforge.net/projects/nsnam/>

⁴ <https://github.com/omnetpp/omnetpp/tree/omnetpp-5.4.1>

⁵ <https://github.com/inet-framework/inet/tree/v4.1.0>

Figure 4 represents the results of memory usage for both simulators on a logarithmic scale as the number of BCs increases. NS2 shows proper memory usage when the BCs are 4 or less. After that, the usage tends to follow an exponential orientation. On both operating systems, OMNets++ shows a strictly controlled memory usage as the number of the BCs increases. The memory usage in Windows shows lower values compared to Linux when the same scenario is being implemented.

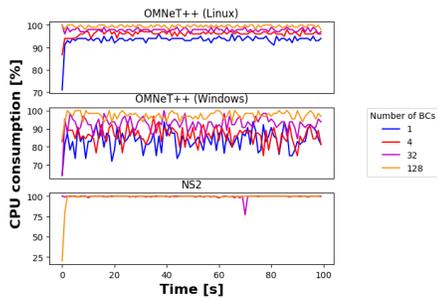


Fig. 3. CPU utilization of NS2 and OMNeT++

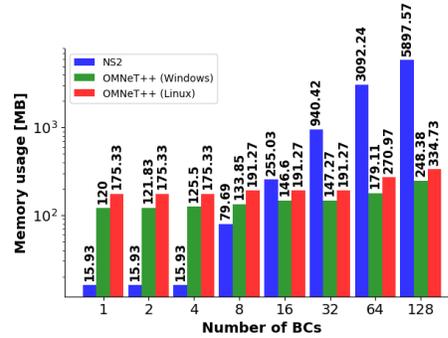


Fig. 4. Memory usage of NS2 and OMNeT++

In order to obtain the execution time in OMNeT++, the express-mode is used, since the normal mode was intentionally built to run slowly to allow the user to trace the events that are occurring during the simulation. Figure 5 represents the execution time for the simulators on a logarithmic scale. We note that NS2 has lower execution time for the scenarios with less than 16 BCs, while OMNeT++ has lower execution time for the scenarios that have 16 BCs or more. The execution times of OMNeT++ in Windows and Linux are similar.

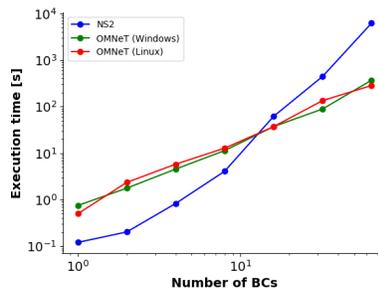


Fig. 5. Execution time of NS2 and OMNeT++

Scalability, as the capacity of supporting scenarios with a huge quantity of WSN components, can be deduced from the CPU utilization, memory usage, and total execution time in terms of number of BCs. Results shown on Figures 3, 4,

and 5 demonstrate that OMNeT++ scales better than NS2. Even though the CPU utilization of OMNeT++ increases as the number of BCs increases, it is comparable to the CPU utilization of NS2 for the largest scenario (Figure 3), its memory usage increases less than NS2 for larger scenarios (Figure 4), and its total execution time is linear in contrast to the super-linear execution time of NS2 (Figure 5).

Energy consumption scenarios: The main objective of the energy consumption scenario is to demonstrate the information that can be obtained from the two simulators. To do so, the same scenarios were implemented on them.

NS2 has only a command-line interface; thus the output is text displayed on the terminal. Information related to energy consumption is not included. Thus, we developed an animator that was integrated to NS2 as a plug-in, in order to control the simulation time, to capture the output, and to extract the energy consumption information. On the other hand, OMNeT++ stores information about the simulations in files, that can be exported in multiple formats for later data processing. OMNeT++ shows the same results both on Windows and Linux, regarding the energy consumption evaluation. Therefore, the results of the energy consumption scenario in OMNeT++ are presented only once and without mentioning the OS.

The energy model in both simulators trace only the energy consumption of RF module, i.e., the consumption of the node CPU and the sensors are not included.

By comparing the results for the same scenarios obtained from NS2 and OMNeT++, there are differences and similarities. In the 802.11b scenarios, both simulators have the capability to accurately simulate the CSMA/CA mechanism, including parameters of PHY and MAC layers of each frame sent during each phase of the mechanism, such as RTS and CTS frames. Additionally, the data and acknowledgement (ACK) frames are simulated as well. The implementation uses the standard guideline to define the length of each frame used in the protocol, as well as the preamble length and the PHY header. The time spent sending RTS, CTS, and ACK frames are similar for both simulators as shown in Table 3. But the time spent to send data frames is higher in the OMNeT++ simulator as Table 4 shows.

Table 3. Time spent in 802.11b for control frames

Frame type	Time in OMNeT++ [μs]	Time in NS2 [μs]
RTS	207	207
CTS	203	202
ACK	203	202

Table 4. Time spent in 802.11b for data frames

Payload [<i>Byte</i>]	Time in OMNeT++ [μs]	Time in NS2 [μs]
10	246	239
30	261	253
50	275	268
70	290	282
90	304	297

By examining one of the repeated interval of the simulation (i.e., the time that includes sending one ping messages), the results show that the energy consumption

of both simulators are not the same. Figure 6 shows the energy consumption in an interval when the frequency is 1 Hz, for the 802.11b scenario, for both simulators. Each pair of columns represents a payload size; the columns to the right is for results obtained from OMNeT++, while the column to the left is for results obtained from NS2. In general, when the same scenario is implemented, the reported energy consumption in OMNeT++ is slightly higher than NS2.

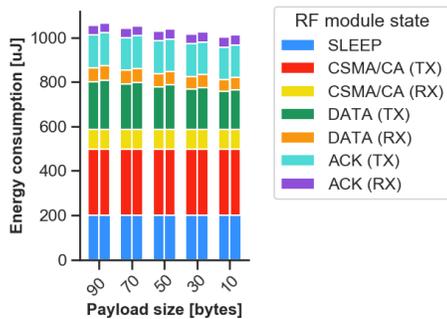


Fig. 6. Energy consumption results of simulations using 802.11b for NS2 and OMNeT++

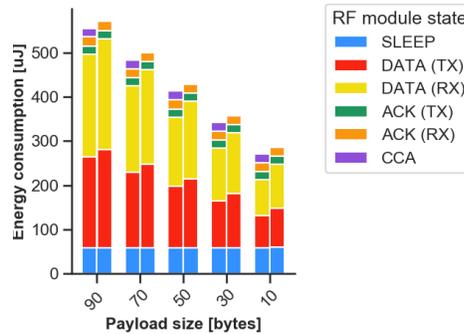


Fig. 7. Energy consumption results of simulations using 802.15.4 for NS2 and OMNeT++

In the 802.15.4 scenarios, both simulators present different implementations of the protocol, this, in turn, affects the energy consumption. Figure 7 shows the energy consumption for one interval of 802.15.4, when frequency is 1 Hz.

In both simulators, an access method is implemented for 802.15.4, it is called Clear Channel Assign (CCA). However, this mechanism is not linked to the energy model in OMNeT++, i.e., using this mechanism does not consume energy. On the other hand, in NS2, CCA is linked to the energy model. As a result, in term of energy consumption, in NS2 there is one more state for the RF model, that is shown in Figure 7 as a violet rectangle called CCA.

Step 6: Elaborate a discussion. From the methodological process, it is possible to detect advantages and disadvantages of the three analyzed WSN simulators. NS2 is a generic data network simulator that was later adapted to suit WSN, while OMNeT++, was built to support the WSN from the beginning. TOSSIM, is an emulator for TinyOS, which is an OS widely used for embedded systems.

The principal drawback of TOSSIM is that it is not compatible with the modern systems. It was not possible to install and run the evaluation scenarios in TOSSIM. Therefore, only the qualitative parameters are available for the comparison with the other simulators. Although, NS2 is only supported on Linux, and despite the fact that it is no longer maintained in favor of NS3, it is one of the most cited simulators in the research domain. This work shows that NS3 has not completely

replaced NS2. NS3 is still in development and many protocols supported in NS2 have not been yet implemented in NS3. OMNeT++ is supported both on Linux and Windows. The project is still maintained and the simulator is regularly updated.

From the performance point of view, the obtained results showed that NS2 is more suitable for projects with less than 128 nodes. In these cases, the execution time in NS2 takes place for a short period of time. When the number of nodes is equal to or greater than 128, the simulation scenario lasts in NS2 longer than in OMNeT++. Moreover, the memory used for scenarios in NS2 with more than 128 nodes are much larger than the memory used in the OMNeT++ simulations for the same scenarios. OMNeT++ is more stable in terms of scalability. As the number of nodes increase, the memory usage increase. However, the CPU utilization is similar for all simulation scenarios. Moreover, the execution time in OMNeT++ using the fast-mode is greater than the execution time in NS2 for the scenarios with less than 16 BCs and it is shorter in the scenarios with 16 or more BCs, which evidence its better ability to scale. By analyzing these results, we note that there is a clear pattern for CPU utilization. Whatever the OS is and whatever the complexity of the scenario. Even though OMNeT++ consumes CPU less than NS2 when the same scenario is implemented, the simulator tends to consume almost all available CPU cycles. On the other hand, as the complexity of the scenarios is increasing, the memory usage increases slightly. Although the results of simulation in OMNeT++ are identical when the same scenario is implemented in Windows and Linux, performance results in Windows show a better CPU and memory utilization.

Both simulators are evaluated using scenarios that are developed to verify the energy consumption models. Results show that they use a similar philosophy to model the energy consumption in a WSN node. The concept is totally based on the RF states of the transceiver, i.e., NS2 and OMNeT++ consider only the energy consumption of the RF module. They do not take into account the consumption of other hardware devices, such as the node CPU or the node on-board sensors.

NS2 and OMNeT++ implement 802.11b with a high level of details. Although the implementations are close, there are still differences in the time spent sending and receiving the data frames. The same problem occurs in the scenarios of 802.15.4. As a result, for the two protocols, when the same scenario is implemented, the time spent sending and receiving data frames in OMNeT++ is always greater than that of NS2. Therefore, when the same scenario is implemented, the energy consumed in OMNeT++ is greater than the energy consumed in NS2.

Results of the energy consumption scenarios indicate that OMNeT++ has consistent results across the two platforms used in the evaluation, i.e., Windows and Linux. This is an important characteristic for simulators that operate across multiple platforms. NS2 is only supported on Linux platforms.

OMNeT++ has a built-in GUI. It allows the user to graphically run the scenarios and to easily debug the source code. It also has a built-in Integrated Develop-

ment Environment (IDE) that helps the developer to identify errors and to check the syntax before compiling the code. These features consume a large amount of resources on the host machine in which the simulator is running, which can be a drawback when working on hosts with limited resource.

6 DISCUSSION ABOUT THE PROPOSED METHODOLOGY

Nowadays, with the huge variety of available simulators, it is important to identify which simulator suits the most for a given scenario. The problem of selection always arises, no matter if the simulator is going to be used for academic purposes or industrial development. From the previous proposals [3] [2], we add criteria to address the evaluation of WSN simulators, in terms of their scalability and capability of modelling mobility, wireless medium, and energy consumption.

In a simulation environment, scalability is a subject governed by the hardware of the simulator host (CPU and memory). Hence, it is not addressed as a separated criterion. Instead, it is handled as a scenario parameter to observe how the number of the nodes can impact the performance in term of CPU utilization and memory usage. Thus, an approximate threshold for the number of nodes which makes decline the simulator performance can be detected. On the other hand, energy issues can be addressed using modelling technologies. Our proposed methodology addresses these issues: it proposes guidelines and criteria to measure the scalability of simulators and to evaluate their energy consumption awareness modelling.

Most WSN simulators models the energy consumption of the RF module. Although the RF activities are responsible for the major part of the energy consumption in the node, the consumption of CPU and sensors cannot be neglected. In [37], authors calculate the power consumption average of the sensor unit, the RF module, and the microcontroller for a WSN application. In their specific application the average of power consumed for the RF activities were 62%, the average of power consumed for the sensor and the microcontroller were 14% and 24% respectively; which means that the RF activities can consume more than the sum of the other units. Therefore, it is important for a simulator to model the energy consumption of all units present in the node in order to get an accurate estimation of the energy consumed by the node. Thus, our methodology evaluate all these aspects.

The proposed methodological approach is flexible, allowing to integrate another items to cover new aspects needed by users. For instance, it is possible to add criteria to evaluate the simulators capacity of modelling the antenna or the battery behaviour. By following the methodology steps, the advantages and disadvantages of one or more simulators for a certain application can be identified. Thus, the selection of one of them can be well justified and probed, as well as its suitability for specific user needs and scenarios.

Although the methodology provides a comprehensive method to compare WSN simulators, there are still aspects to be covered. For example, the study of energy modelling can be extended to include the support for the battery model. When

considering the estimation of the node lifetime, the model that trace the remaining energy is different from the one that trace the consumed energy. The support of parallel processing is another item that can be extended as well. This feature exists in some simulators and has a huge effect on performance.

Besides, wireless link protocols have special role in WSN. Thus, it is recommended to separate it from the protocol items and consider additional aspects that concerns the users of the simulators, such as the support of different bit rates and fragmentation. Finally, WSNs are still in developing and new technologies will be adapted. Thus, new features will be added and WSN simulators have to answer to that. Our methodology faces all these challenges by being extensible, flexible, and generic, and still being a powerful tool to evaluate and compare network simulators.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have address the difficulty of selecting a WSN simulator to fit a given scenario. To achieve that, we extend our previous proposed methodology, by integrating new criteria to address WSN evaluation, such as scalability and the modelling of mobility, wireless medium, and energy consumption.

In order to demonstrate the efficiency and suitability of our methodology, we elaborate the state of the art of WSN simulators, following a systematic review of most cited and recent scientific papers. From this review, we select the three most cited WSN simulators (i.e., NS2, TOSSIM, and OMNeT++) to evaluate and compare them following our proposed methodological approach. The application of the methodology 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 include other evaluation criteria, such as the capacity of simulators for parallel processing and support of different bit rates and fragmentation. We are also working on proposing an energy consumption model to include the support for the battery behaviour modelling.

ACKNOWLEDGEMENTS

The work presented in this paper has been financially supported in part by the Regional Council of New Aquitaine (as part of the Call for Projects 2016 funds), in the frame of the OUDINI research project.

References

1. Abuarqoub, A., Hammoudeh, M., Alfayez, F., Aldabbas, O.: A Survey on Wireless Sensor Networks Simulation Tools and Testbeds, vol. 3, chap. 14, pp. 283–302. IFSA (01 2016)
2. Bakni, M., Cardinale, Y., Moreno, L.: Experiences on evaluating network simulators: A methodological approach. *Journal of Communications (JCM)* pp. 1–11 (2019)
3. Bakni, M., Cardinale, Y., Moreno, L.M.: An Approach to Evaluate Network Simulators: An Experience with Packet Tracer. *Revista Venezolana de Computación* **5**, 29 – 36 (Jun 2018)
4. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. *Computer* **40**(12) (2007)
5. Chéour, R., Jmal, M.W., Kanoun, O., Abid, M.: Evaluation of simulator tools and power-aware scheduling model for wireless sensor networks. *IET Computers & Digital Techniques* **11**(5), 173–182 (2017)

6. Chernyshev, M., Baig, Z.A., Bello, O., Zeadally, S.: Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal* **5**, 1637–1647 (6 2018)
7. Chhimwal, P., Rai, D.S., Rawat, D.: Comparison between different wireless sensor simulation tools. *IOSR Journal of Electronics and Communication Engineering* **5**(2), 54–60 (2013)
8. Derhab, A., Ounini, F., Remli, B.: Mob-tossim: An extension framework for tossim simulator to support mobility in wireless sensor and actuator networks. In: *Internat. Conf. on Distributed Computing in Sensor Systems*. pp. 300–305 (2012)
9. Du, W., Navarro, D., Mieleveville, F., Gaffiot, F.: Towards a taxonomy of simulation tools for wireless sensor networks. In: *Inter. Conf. on Simulation Tools and Techniques*. pp. 52:1–7 (2010)
10. Fahmy, H.M.A.: Simulators and emulators for wsns. In: *Wireless sensor networks*, pp. 381–491. Springer (2016)
11. G Gupta, S., Ghonge, M., D P M Thakare, P., Jawandhiya, P.: Open-source network simulation tools: An overview. *Internat. Journal of Advanced Research in Computer Engineering and Tech.* **2** (2013)
12. Garg, K., Frster, A., Puccinelli, D., Giordano, S.: Towards realistic and credible wireless sensor network evaluation. vol. 89 (09 2011). https://doi.org/10.1007/978-3-642-29096-1_4
13. Godoy, D., Sosa, E., Daz Redondo, R., Bareiro, H.: Webshawn, simulating wireless sensors networks from the web. pp. 190–195 (10 2017). <https://doi.org/10.1109/WiMOB.2017.8115829>
14. Helkey, J., Holder, L., Shirazi, B.: Comparison of simulators for assessing the ability to sustain wireless sensor networks using dynamic network reconfiguration. *Sustainable Computing: Informatics and Systems* **9**, 1–7 (2016)
15. Imran, M., Abas, S., Halabi, H.: A survey of simulation in sensor networks. *Information Technology Internat. Symp, IEEE* **2** (2010)
16. Kellner, A., Behrends, K., Hogrefe, D.: Simulation environments for wireless sensor networks. Tech. rep., Inst. of Computer Science – Georg-August-Universit at Göttingen (2010)
17. Khan, M.Z., Askwith, B., Bouhaf, F., Asim, M.: Limitations of simulation tools for large-scale wireless sensor networks. In: *Internat Conf on Advanced Informat Networking and Apps*. pp. 820–825 (2011)
18. Krishna, K.H., Kumar, T., Babu, Y.S.: Energy effectiveness practices in wsn over simulation and analysis of s-mac and leach using the network simulator ns2. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. pp. 914–920. IEEE (2017)
19. Lahmar, K., Cheour, R., Abid, M.: Wireless sensor networks: Trends, power consumption and simulators. In: *Sixth Asia Modelling Symposium*. pp. 200–204 (2012)
20. Malavolta, I., Mostarda, L., Muccini, H., Ever, E., Doddapaneni, K., Gemikonakli, O.: A4wsn: an architecture-driven modelling platform for analysing and developing wsns. *Software & Systems Modeling* **18**(4), 2633–2653 (2019)
21. Minakov, I., Passerone, R., Rizzardi, A., Sicari, S.: A comparative study of recent wireless sensor network simulators. *ACM Transactions on Sensor Networks (TOSN)* **12**(3), 20 (2016)
22. Musznicki, B., Zwierzykowski, P.: Survey of simulators for wireless sensor networks. *Journal of Grid and Distributed Computing* **5**, 23–50 (09 2012)
23. Navarro, D., Mieleveville, F., Galos, M., Carrel, L.: Simulation of hardware and software in heterogeneous wireless sensor network. *Journal on Advances in Networks and Services Volume 7, Number 1 & 2* (2014)
24. Nayyar, A., Singh, R.: A comprehensive review of simulation tools for wireless sensor networks (wsns). *Journal of Wireless Networking and Communications* **5**, 19–47 (2015)
25. Nursetitov, N., Paulson, M., Reynolds, R., Izurieta, C.: Comparison of json and xml data interchange formats: a case study. *Caine* **9**, 157–162 (2009)
26. Papadopoulos, G.Z., Kritsis, K., Gallais, A., Chatzimisios, P., Noel, T.: Performance evaluation methods in ad hoc and wireless sensor networks: a literature study. *IEEE Communications Magazine* **54**(1), 122–128 (2016)

27. Perla, E., Huggard, M., Mc Goldrick, C., Carbajo, R., Cathin, A.: Powertossim z: Realistic energy modelling for wireless sensor network environments. In: Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (2008)
28. Pestic, D., Radivojevic, Z., Cvetanovic, M.: A survey and evaluation of free and open source simulators suitable for teaching courses in wireless sensor networks. In: Internat. Convention on Information and Communic. Tech, Electronics and Microelectronics. pp. 895–900 (2017)
29. Rahman, M., Pakstas, A., Zhigang Wang, F.: Network modelling and simulation tools. Simulation Modelling Practice and Theory **17** (2013)
30. ur Rehman Khana, A., Bilal, S., Othman, M.: A performance comparison of networks simulators for wireless networks. Internat. Conf. on Control System, Computing and Eng. (2012)
31. Robinson, S.: Conceptual modelling for simulation part i: Definition and requirements. Journal of the Operational Research Society **59**, 278–290 (2008)
32. Saginbekov, S., Shakenov, C.: Testing wireless sensor networks with hybrid simulators. arXiv preprint arXiv:1602.01567 (2016)
33. Sarkar, N.I., Halim, S.A.: A review of simulation of telecommunication networks: simulators, classification, comparison, methodologies, and recommendations. Cyber Journals pp. 10–17 (2011)
34. Socolofsky, T.J., Kale, C.J.: Tcp/ip tutorial. Tech. rep. (1991), no. RFC 1180
35. Stetsko, A., Stehlik, M., Matyas, V.: Calibrating and comparing simulators for wireless sensor networks. In: Internat. Conf. on Mobile Ad-Hoc and Sensor Systems. pp. 733–738 (2011). <https://doi.org/10.1109/MASS.2011.80>
36. Sundani, H., Li, H., Devabhaktuni, V., Alam, M., Bhattacharya, P.: Wireless sensor network simulators a survey and comparisons. Internat. Journal of Computer Net. **2**(5), 249–265 (2011)
37. Terrasson, G., Briand, R., Basrour, S., Dupe, V., Arrijurria, O.: Energy model for the design of ultra-low power nodes for wireless sensor networks. Procedia Chemistry **1**, 1195–1198 (2009)
38. Vasanthi, V.: Simulators and emulators used for wireless sensor network. International Journal of Advanced Research in Computer and Communication Engineering **6** (1 2017)
39. Yick, J., M., B., G., D.: Wireless sensor network survey. Computer networks **52:12**, 2292–2330 (2008)
40. Yu, F., Jain, R.: A survey of wireless sensor network simulation tools. Washington University St. Louis, Dep. of Science and Eng. (2011)

Authors

Michel Bakni 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.

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

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.

Guillaume Terrasson received the M.S. degrees in Microelectronics and the Ph.D degrees in Electronics from the University of Bordeaux 1, France, in 2004 and 2008 respectively. The Ph.D. was obtained in collaboration with the ESTIA Engineering School, Bidart, France and the TIMA laboratory, Grenoble, France. Since 2008, he is a researcher at ESTIA RECHERCHE. Qualified in sections CNU in 2010 (French Universities National Council) n63 (electrical engineering, electronics, photonics and systems), his research interests include critical embedded systems applied to aeronautics as well as wireless sensor networks. Guillaume TERRASSON has also managed collaborative European or National programs like Interreg POCTEFA.

Octavian Curea received the Engineer degree in Electrical Engineering at Polytechnic Institute Traian Vuia of Timisoara, Romania, in 1994. He pursued his studies in France by obtaining the M.Sc. degree in 1997, and the Ph.D. degree in 2001, at University of Le Havre, GREAH laboratory. In 2004 he joined ESTIA as associated profesor and he became full profesor in 2017. His research interests include the electrical microgrids from the point of view of energy management, communication networks, power electronics, power quality.