Ali. The Arab Journal of Sciences & Research Publishing, Vol. 2 - Issue (3): 2016, 4, 21 P. 51-66;
Article no: AJSRP/ H27116.

Simulation of Wireless Multimedia Sensor Networks (WMSN)

Hanafy M. Ali
Computers and Systems Department, Faculty of Engineering, Minia University, El Minia, Egypt.
*Corresponding E-mail: hmali@mu.edu.eg

Abstract

Abstract— Wireless multimedia sensor networks (WMSNs) became widely popular because of the availability of low-cost easy-to-use multimedia sensors and devices capable of retrieving and transmitting multimedia content like sound, images and video streams. Simulation of WMSN has followed techniques and tools that are mainly used by similar wireless sensor networks (WSN). This, unfortunately, was not sufficient for the data-intensive nature of WMSN. Therefore, it is still challenging to provide sufficient simulation frameworks for WMSNs. This paper presents a survey of available simulation frameworks and tools for WMSNs. The challenges facing the simulation process are first introduced. Then, the attributes of the ideal WMSN simulation are presented and reviewed. The general model of WMSN simulation environments are also presented and discussed. Finally, a detailed survey of available simulation frameworks and tools are presented. Tools that are inherited from wireless sensor network (WSN) and those made specially for WMSNs are also presented.

Key words: *Wireless Sensor Networks, Wireless Multimedia Sensor Network, Video Sensor Network, Simulation, Survey.*

---

1. INTRODUCTION

Wireless Sensor Networks (WSNs) contain small devices known as nodes that collect data from their physical environment and transmit them through the network to a central control station where they are analyzed and possibly decisions are made [1]. Nodes are usually equipped with a power-efficient microcontroller, a power supply source, a wireless communication device and one or more sensors to collect such data, examples are humidity, temperature, pressure or vibration sensors [2]. WSNs tend to have easy and cheap employment features and this expanded their usage to a wide application spectrum [3].

Wireless Multimedia Sensor Networks (WMSN) are wirelessly connected nodes that are able to retrieve and transmit multimedia content like video and audio streams. Nodes in WMSNs are usually characterized as nodes that need to do multimedia (video/audio operations) in the most reliable and power-efficient manner. The emergence of low-cost technologies in image/video sensory (CMOS image sensors and microphones), digital signal processing as well as wireless communication have the most impact on enabling and the spread of WMSNs [4], [5], [6]. While WSNs operate on simple scalar measurements, WMSNs deal with more complex vector-based data such as video and audio [7] where hundreds or thousands of media sensing devices, communicating by means of wireless transmission, form WMSNs that serves different applications.

A wide variety of applications of WMSNs in military and civil sectors includes, and not limited to, those relating to surveillance [8] [9] [10] [11], traffic monitoring and control [12] [13], health care [14] [15], environmental monitoring [16] and Industrial process control [17]. Research on WMSNs and WSNs share similar applications and technical problems, but in WMSNs, specific challenges are more complex:

- *High Bandwidth Demand*. Multimedia content, especially video streams, require transmission bandwidth that is orders of magnitude higher than that supported by currently available sensors [18] [19]. Many researches are currently oriented to enhancing the performance of current WSN protocols and standards [20] [21].
- *Power Consumption*. Energy is a primary concern in WMSN. Nodes usually run on non-rechargeable batteries and therefore, the expected lifetime is a fundamental element. Unlike WSNs, WMSNs consume more energy because of the extensive usage of multimedia sensors and the power dissipated in processing of multimedia-rich data. Therefore, there is a general demand for creating efficient architectures [22], algorithms, and protocols [23] [24] [25] [26].
- *Application-specific requirements*. The variety of applications served by WMSNs will have different requirements. Along with regular scalar data delivery that need to be supported, multimedia-data requires energy-efficient snapshots and streaming [27] [28]. Snapshots are triggered-data obtained in a short time periods while streaming contents are generated over longer time periods. In addition, other challenges arise in terms of supporting mobility [29], platform diversity [30], security and privacy [31] [32], multisensory image fusion [33] [34], image and video compression [35] [36].

The successful deployment of wirelessly connected sensor nodes is one of the major concerns for sensor network research. Many factors may case partial or complete failures to the deployment process: failures in supporting hardware or software, interference from the environment, and incorrectly optimized parameters of algorithms and applications [37]. The success rates become even worse for WMSN where deployment experiments involve multimedia transmission. Video and audio streams should be monitored and evaluated based on many Quality of Service (QoS) and Quality of Experience (QoE) standards. Problems like frame loss, jitter, delay, encoding/decoding errors, and distortion need to be detected and corrected before a successful operation of the network. Therefore, the pre-deployment process of WMSN need to be supported with extensive multimedia-aware simulation frameworks to improve its success rates.

For WMSN simulation frameworks, there are two kinds of tradeoffs. The first is between accuracy and performance of the simulated network/sensor models. Increasing the accuracy results in obtaining poor performance. The second tradeoff is between details and scalability. The more the details, the less scalable the model is. Therefore, we define the goals of the ideal WMSN simulation tool as:

- Providing models with a suitable number of details to be scalable
- Having a friendly graphical user interface (GUI)
- Running at highest possible performance compared to the desired accuracy level.

Currently available WSN simulators may be used to simulate WMSN. However, this requires more effort to adapt such simulation frameworks to the requirements of current WMSN. On the other hand, many extensions to available WSN simulators were developed to extend their functionalities in order to support WMSN. Although this seems to solve the problem, such extensions are usually developed as layers or plugins over the existing simulation kernels, which introduces many simulation-performance concerns and limits their scalability. In this paper, we made our effort to survey available simulation tools and frameworks that can be used for simulating WMSN.

2. The Generic WMSN Simulation Model

The development of WMSN simulation tools requires corresponding models have which include new components, not present in classical network simulators. This can be achieved by developing new simulation frameworks that captures the requirements of WMSN simulation or by extending available traditional network/WSN simulators with new models. This section describes a general component model for WMSN simulation tools. The presented model is suitable for most of the tools employed in the on-going research on WMSN. The model is logically composed of three sub models: The node model, the network model and the environment model.

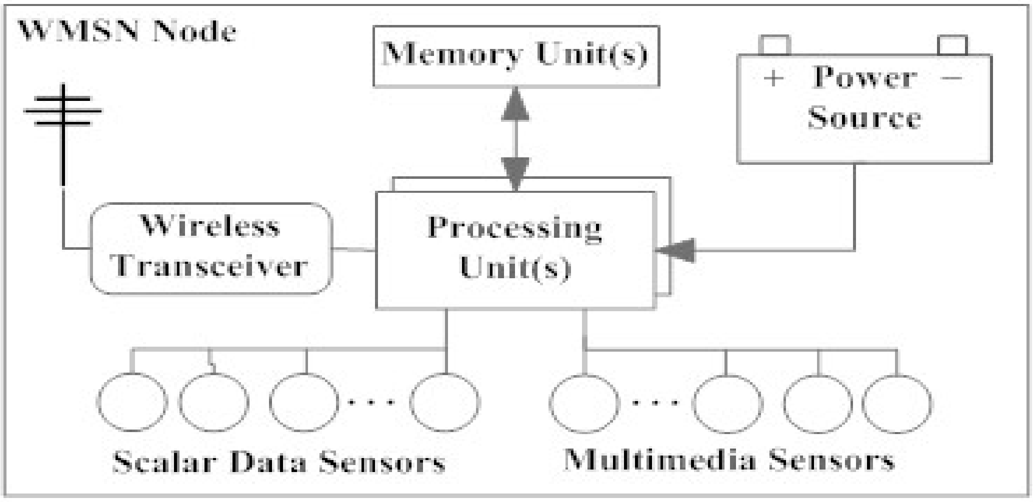

Fig 1. General structure of WMSN nodes.

2. A. WMSN Node Modem

A sensor node is main acting point in the WMSN. It consists of a power source, a microcontroller, memory, wireless transceiver and a set of multimedia or scalar –data sensors. Figure 1 describes the general structure of WMSN node.

The regular WMSN node will consist of a processing unit or more for data-intensive computation and communication. A memory unit is necessary for data storage, pre transmission data and inter-computation data. The power source is one of the most important parts that supplies all other parts with necessary power. It also incorporates a set of simple scalar data sensors like temperature sensors, humidity sensors … etc. As it is a multimedia sensor node, it also include a set of multimedia-data sensors like cameras, microphones … etc. The wireless transceiver is responsible for sending/receiving data between other nodes or possible a base station.

The simulation model of the WMSN node should include enough details so successfully simulate a set of interacting nodes with best possible performance and scalability. Whenever needed, the simulation model should include a representation of power source behavior regarding the tasks performed on the microcontroller, the power computation of multimedia sensors and wireless transceivers. The transceiver model should be able, when needed, to simulate low level details of network protocols and allows for packet inspection and tracing. The simulated microcontroller model may vary from low level specific microcontroller model that simulates machine language code to higher level model that simulates a dummy code. The simulated sensors must be of a good variety of available image and acoustic sources with support of different data attributes.

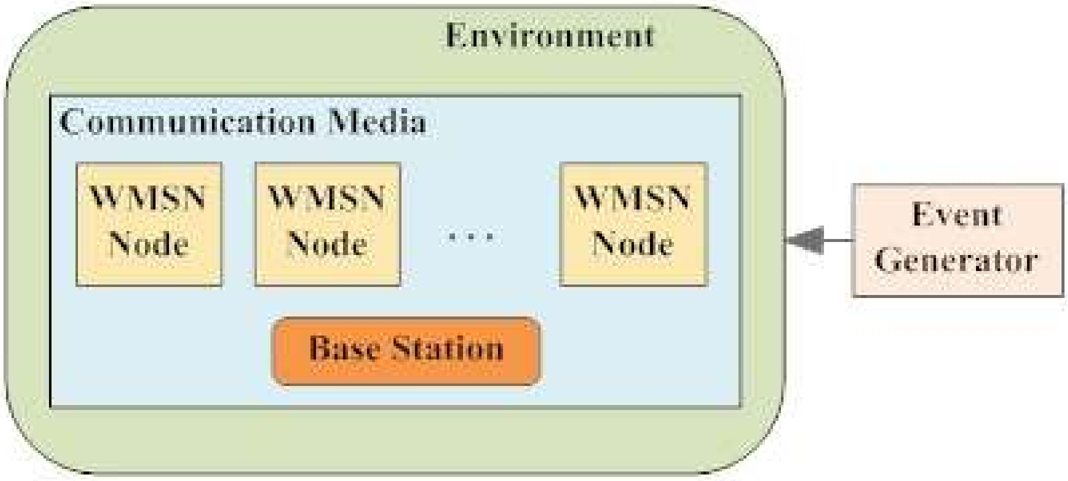

Figure 2. WMSN Network Model

2. B. WMSN Network Model

Figure.2 presents the generalized network model within WMSN simulation frameworks. The core component is the WMSN node described before. Nodes communicate together within the communication media using their transceiver modules. The communication media governs the used protocol and semantics of wireless communication. The environment is one of the major components affecting the simulation of generally all WSNs. It affects the communication attributes and sensors within nodes. The base station is a basically a special type node that collects data from another nodes. Its usage depends on the simulated network. The event generator triggers the environment attributes which in turn affects the node sensors

3. Selection of the Most Suitable Framework

Because of the wide range of simulation scenarios, sensor types, details of models and communication protocols, selecting the most suitable framework is task that should be given adequate attention. In WMSN, multimedia-awareness and mobility are essentials to fulfil researches' needs to verify and test their protocols and applications, and consequently, time and cost can be reduced dramatically. Nevertheless, current network frameworks that can be used for WMSN do not provide full support for the multimedia QoS/QoE and/or they do not provide a large set of mobility models.

Actually, most of the tools were originally designed to solve particular problems rather than providing a general framework for WMSN pre-deployment. Consequently, the selection of the most suitable framework for WMSN is a challenge that requires studying different tools' features and capabilities.

4. Available WMSN Simulation Frameworks

There are several traditional network simulators used by many researchers for WSN simulation. The most common ones are NS-2 [38], Omnet++ [39], TrueTime [40], OPNET [41], and GloMoSim [42] and they differ in the programming language, support for scripting, support for standard protocols, and their extendibility.

a. Omnet++

Omnet++ offers a powerful discrete-event simulation engine and a friendly GUI interface and high extensibility. It supports many extensions for mobility, wireless protocols, and energy models. Two of the most common extensions for WSN are INET [43], MiXiM [44], Castalia [45] and PAWiS [46]. Castalia supports modelling classical WSNs algorithms within a realistic real-data-based communication models while PAWiS simulates nodes with exceptionally low power consumption, yet have to drive sensors and radio communication.

Due to its high extensibility nature, many frameworks extended Castalia to support WMSN. WiSE-MNet [47] is based on Castalia/OMNet++ and provides a simulation environment for networks for WMSNs by addressing the need for co-designing network protocols and distributed algorithms. However, it does not provide video control and QoE support, which is a key characteristic to enable multimedia evaluation from the user's perspective [48]. WVSN [49] presents a Castalia-based simulation model for wireless video sensor networks by defining the sensing range of camera nodes with a Field of View (FoV) which is more realistic for WMSNs.

M3WSN [48] is another extension to Castalia that was proposed by Denis Rosário et al. as a WMSN framework. It extends Castalia with functionalities of bothWiSE-MNet and WVSN models and, additionally, supports transmission, control and evaluation of real video sequences in mobile WMSNs

b. NS-2

NS-2 is the most common network simulator. However, it does not have a direct support for WSN simulation. Extensions to NS-2 have been made to support ad-hoc networks but mobility of nodes requires more extensions. These extensions make dealing with mobile nodes difficult. Also external modules, like Mannasim [50], have been added to support the IEEE 802.11 and IEEE 802.15.4 standards. EvalVid tool is an extension that can be integrated with NS-2 to send a real time multimedia streaming over WSN and proposed by Chih-Heng [51].

c. TrueTime

TrueTime is a Matlab/Simulink toolbox that is originally designed for the simulation of distributed control algorithms. However, it provides basic support for the IEEE 802.15.4 standard making it ready for WSN simulation. It provides a simulation environment for network control and sensor network. A unique characteristic is that True Time could simultaneously simulate the computations within nodes, the power consumption of node batteries, the node dynamic changing (node communication range in specific transmission power, package sending or receiving, network congestion, package reception rate, etc.) [52].

d. OPNET

OPNET (Optimized Network Engineering Tools) is a commercial network simulator that can be used to model and simulate data networks. It can simulate heterogeneous networks that use various communication protocols. Chen et al. [53] used OPNET to implement many protocols for WMSN. Since Oct. 2012, OPNET is acquired by Riverbed and is now known as (SteelCentral NetModeler Suite) [54].

e. TOSSIM

TOSSIM [55] is a free tool, created from scratch and it is a discrete-event simulator for TinyOS WSNs, which is an operating system commonly used for WSN testbeds. While TOSSIM is a capable tool, it does not provide means for scenario management and statistics management, and does not have the wealth of link models available in OPNET [56]. Almeida et al. [57] used TOSSIM network simulator to test and implement their reliable transport protocol for wireless multimedia sensor networks.

f. SENSE

SENSE [58] (Sensor Network Simulator and Emulator) was developed originally for WSN simulation. This simulator addresses three important factors: extensibility, reusability, and scalability. It also takes into account the needs of different users.

Table I. Available WSN/WMSN used for multimedia-based simulation

Framework	Timing	Programming Language	Supported Platforms	GUI Support	WSN Extensions	WMSN Extensions
NS-2	Discrete-Event Simulation	C++, OTcl	FreeBSD, Linux, SunOS, Solaris, Windows (Cygwin)	No	Mannasim [50]	EvalVid [51]
NS-3	Discrete-Event Simulation	C++, Python	FreeBSD, Linux, SunOS, Solaris, Windows (Cygwin)	Yes	Already Supported	Custom Simulators [59]
OMNet++	Discrete-Event Simulation	C++, NED	Linux, Unix, Windows(Cygwin)	Yes	INET [43], MiXiM [44], Castalia [45], PAWiS [46]	WiSE-MNet [47], WVSN [49], M3WSN [48]
OPNET		C++	Windows	Yes	Custom Simulators [60]	Custom Simulators [61]
TOSSIM	Discrete-Event Simulation	nesC,Python,C++	Linux, Windows (Cygwin)	Yes	Already Supported	Custom Simulators [62]
TrueTime		Matlab	Windows, Linux	Matlab GUI Support	Custom Simulators [63]	Custom Simulators [63]
EmStar	Trace-Driven Simulation		Linux	Yes	Already Supported	Custom
Cooja	Discrete-Event Simulation	Java,C	Linux	Yes	Custom Simulators [64]	Custom Simulators [64]
J-Sim	Discrete-Event Simulation	Java, Tcl	Windows, Linux	Yes	G-JSIM [65]	Custom Simulators [66]
WSNet	Event-Driven	C++	Open Source	No	Already Supported	XS-WSNet [67]
ATEMU	Cycle-Accurate Instruction Level	C	Windows, Linux	Yes	Already Supported	
Avrora	Cycle-Accurate Instruction Level	Java	Windows, Linux	No	Already Supported	Custom Simulation [68]
GloMoSim	Prallel Discrete-Event Simulation	C	Windows, Linux	Yes	Custom Simulation [69]	Custom Simulation [70]

5. OMNet++, the Network Simulation Platform

OMNeT++ is a C++ based discrete event simulator for modeling communication networks, multiprocessors and other distributed or parallel systems. OMNeT++ is open-source, and it can be used either under the GNU General Public License or under its own license that also makes the software free for non-profit use [76].

The motivation of developing OMNeT++ was to produce a powerful open-source discrete event simulation tool that can be used by academic, educational or research-oriented commercial institutions for the simulation of computer networks and distributed or parallel systems. OMNeT++ tries to fill the gap between open-source, research-oriented simulation software such as ns (Bajaj et al. 2000) and expensive commercial alternatives like OPNET (OPNET Technologies, Inc.). A later section of this chapter presents a comparison with other simulation packages.

Unlike other simulation platforms, OMNeT++ was designed from the beginning to support network simulation in the large scale with the availability of extensions. This objective lead to the following main design requirements:

- To enable large-scale simulation, simulation models need to be hierarchical, and should be built from reusable components as much as possible.
- Simulation programs are infamous for long debugging periods. Thus, the simulation software should place large emphasis on easy traceability and debuggability of simulation models to reduce debugging time. (The same feature set is also useful for educational use of the software).
- The simulation software itself should be modular, customizable and should allow embedding simulation models into larger applications such as network planning software. (Embedding brings additional requirements about the memory management, restartability, etc. of the simulation).
- Data interfaces should be open: it should be possible to generate and process input and output files with commonly available software tools.

A. OMNeT++ Models and Modules

An OMNeT++ model consists of modules that communicate with message passing. The active modules are termed simple modules; they are written in C++, using the simulation class library. Simple modules can be grouped into compound modules and so forth; the number of hierarchy levels is not limited.

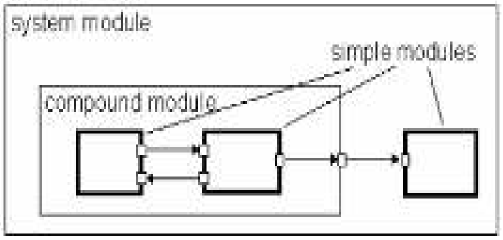

Figure 3: Model Structure in OMNeT++.

Figure 3 shows the model structure within OMNet++. Boxes represent simple modules (thick border), and compound modules (thin border). Arrows connecting small boxes represent connections and gates. Both simple and compound modules are instances of module types. While describing the model, the user defines module types; instances of these module types serve as components for more complex module types. Finally, the user creates the system module as an instance of a previously defined module type. When a module type is used as a building block, there is no distinction whether it is a simple or a compound module. This allows the user to transparently split a simple module into several simple modules within a compound module, or do the opposite, re-implement the functionality of a compound module in one simple module, without affecting existing users of the module type.

B. The NED Language

The user defines the structure of the model (the modules and their interconnection) in OMNeT++'s topology description language, NED. Typical ingredients of a NED description are simple module declarations, compound module definitions and network definitions. Simple module declarations describe the interface of the module: gates and parameters. Compound module definitions consist of the declaration of the module's external interface (gates and parameters), and the definition of sub-modules and their interconnection. A network definition basically defines a model as an instance of a module type.

NED and the OMNeT++ model structure were designed to promote reusable model components via module types and unlimited compound module hierarchy levels. NED also supports partitioning large NED files into several smaller ones via file inclusion. The OMNeT++ package includes a graphical editor (shown in Fig. 4) which uses NED as its native file format; moreover, the editor can work with arbitrary, even hand-written NED code. The editor is a fully two-way tool, i.e. the user can edit the network topology either graphically or in NED source view, and switch between the two views at any time.

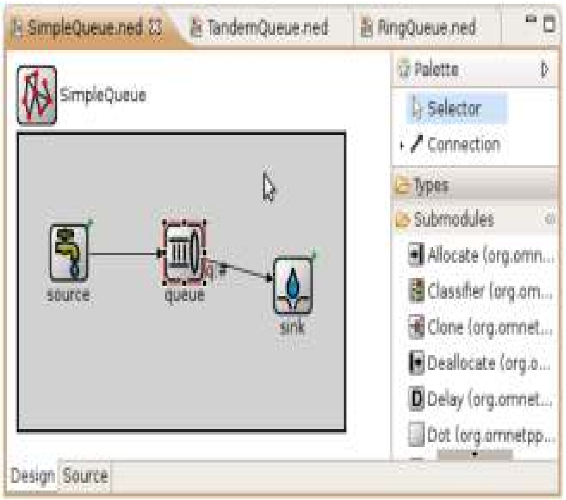

Figure 1: Graphical NED Editor of OMNet++

C. Internal Structure and Simulation Kernel

OMNeT++ simulation programs possess a modular structure. The logical architecture of OMNeT++ is shown in Fig 5.

The Model Component Library consists of the code of compiled simple and compound modules. Modules are instantiated and the concrete simulation model is built by the simulation kernel and class library (Sim) at the beginning of the simulation execution. The simulation executes in an environment provided by the user interface libraries (Envir, Cmdenv and Tkenv) – this environment defines where input data comes from, where simulation results go to, what happens to debugging output arriving from the simulation model, controls the simulation execution, determines how the simulation model is visualized and (possibly) animated, etc.

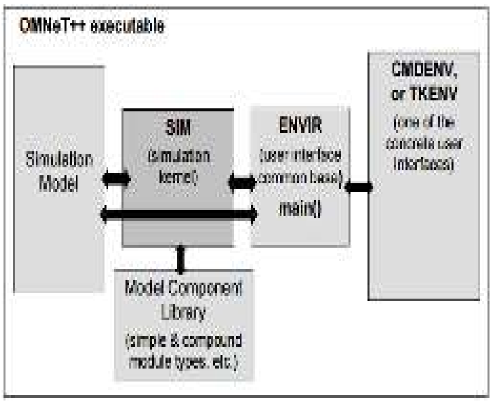

Figure 2: Logical Architecture of an OMNeT++ Simulation Program

One of the most powerful features of OMNet++ is that the user can replace the interface libraries and customize the full environment in which the simulation runs, and even embed an OMNeT++ simulation into a larger application.

D. GUI, Animation and Tracing

An important feature of OMNeT++ is the easy debuggability and traceability of simulation models. This is mainly implemented in Tkenv, the GUI user interface of OMNeT++. Tkenv uses three methods: automatic animation, module output windows and object inspectors. Automatic animation (i.e. animation without any programming) in OMNeT++ is capable of animating the flow of messages on network charts and reflecting state changes of the nodes in the display. Automatic animation perfectly fits the application area, as network simulation applications rarely need fully customizable, programmable animation capabilities. Figure 6 shows a screenshot of Tkenv and its capabilities.

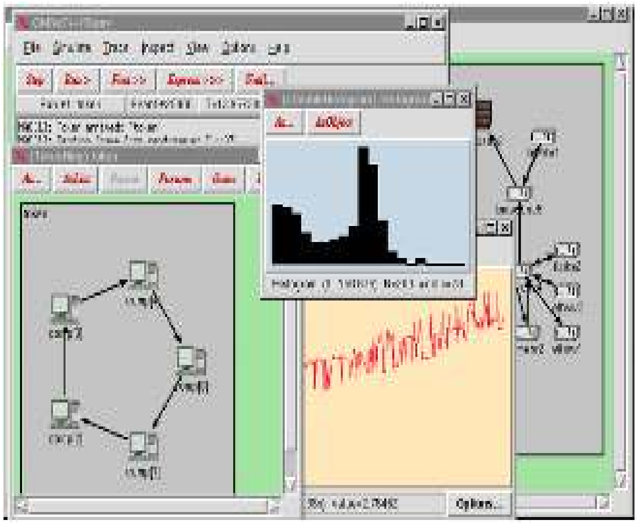

Figure 3: Screenshot of the Tkenv User Interface of OMNeT++.

E. Why OMNet++

There are numerous network simulation tools on the market today, both commercial and non-commercial ones. However, OMNet captured the interest for this research because of the following features:

- **Programmability that support Extensibility.** The user implements arbitrary new building blocks unlike many other platforms where he is confined to the predefined blocks implemented by the supplier. Some tools are not programmable by the user to this extent therefore they cannot be compared to OMNeT++.
- **Model libraries and available models.** Protocol models are available within OMNeT++ which is similar to many platforms like OPNET which has probably the largest selection of ready-made protocol models (including TCP/IP, ATM, Ethernet, etc.). NS also has a large number of protocol models, mostly centered on TCP/IP. CLASS only supports ATM networks. OMNeT++ currently has detailed TCP/IP, SCSI and FDDI models.
- **Support for structured, reusable simulation models.** Network simulation tools naturally share the property that a model ("network") consists of "nodes" (blocks, entities, modules, etc.) connected by "links" (channels, connections, etc.). Some simulation tools (Parsec, C++Sim) do not provide explicit support for topology description: in Parsec, one must program a "driver entity" which boots the model by creating the necessary nodes and interconnecting them. This solution does not enforce the separation of defining model structure from defining the functionality, and possibilities for model component reuse are rather poor. Other tools (NS, CLASS).
- **Performance.**
- **Debugging and Tracing Support.**
- **Source Availability.**

6. Castalia, the WSN Extension-enabling Framework

Castalia was started by A. Boulis [77] as a need for a WSN project. He and his team wanted to test some communication patterns in simulation before moving in real systems. In order to do that, they wanted accurate-enough radio/channel models so that the simulation results would become meaningful and guide them in their search. They used OMNeT++ as the base to build a reliable and fast event-driven simulator in order to focus on the models and overall design and not on the event-driven simulation engine. Shortly after, they have decided to build an open expandable and reliable framework for WSN simulation needs

Castalia was first developed as a simulator for WSN, Body Area Networks (BAN) and generally, networks of low-power embedded devices. It is an additional layer added to the famous OMNet++ simulation framework. It was from the first place designed for extensions [User Manual], so that other researches can use Castalia to build on it. The most powerful feature of Castalia is its ability to support new WSN platforms with:

1. An advanced channel model that is based on empirical measurements. The model implements a model for path loss rather than a simple connection between nodes.
2. The introduction of a radio model based on real low-power radios to support multi TX power levels with individual node variations.
3. The flexibility of the physical model and support for its extension.
4. Support of sensing device noise, bias and power consumption.

Therefore, developers will be able to extend the functionalities of the powerful OMNet++ simulator to levels beyond regular WSN simulations. Features like modularity, reliability, and speed are inherited directly from OMNeT++. However, Castalia is not a platform-specific framework, i.e. testing the actual binary code compiled for specific platform. It is meant to provide generic support for pre-implementation or to be extended for further simulation scenarios.

Figure 7 shows Castalia's basic modules and their interconnection. This forms the Castalia framework that is able to support basic WSN simulation and further extensions.

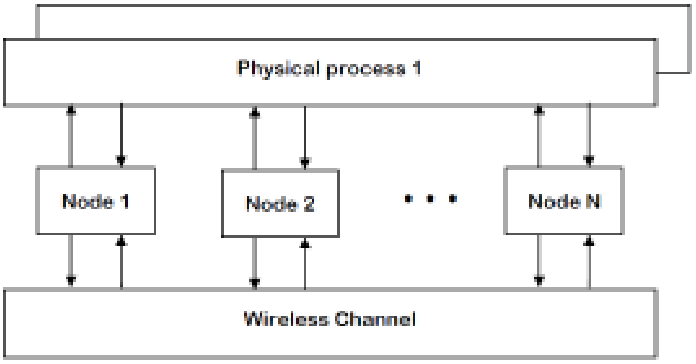

Figure 4: Castalia's Modules and their interconnection

The simulated nodes are by default connected to each other not directly but through the wireless channel. This means that when a node tries to send a packet to another node, it has to be sent to the wireless channel that routes the packet to the target node. This allows for path loss modeling with even complex temporal variations. In addition, interference is handled more properly.

On the other hand, nodes are directly connected to the physical processes (the simulation environment). Nodes can monitor physical processes by sampling them in space and time (i.e. sending a message to the corresponding module) to get their sensor measurements. The model supports multiple processes to be represented.

Nodes in Castalia are composite OMNet++ modules (i.e. composing of sub modules) implemented in NED language. Figure 8 shows the internal structure of the node model in Castalia. The sensor manager module is connected to a physical process to sample sensor measurements using messages (represented by solid lines). The application module represents the actual application of the WSN node and that most simulator user will configure to adapt it to their application needs. A composite communication module is presented to handle the connection from/to the wireless channel. The resource manager module represents the main resources of the WSN node and other modules communicate with it with simple function calls (dashed lines). For example, most modules contact it to inform it that power has been consumed.

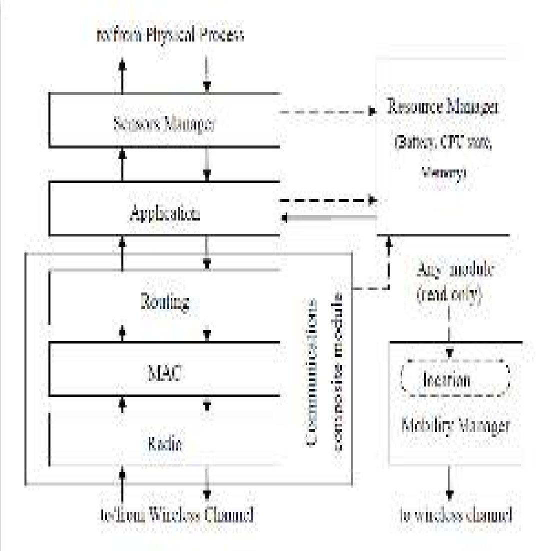

Figure 5: WSN node model in Castalia

Because of the high extensibility of Castalia, users can not only adjust the application module but also extend any sub module of the WSN node. For example, the communication's MAC and Routing modules, as well as the mobility manager module, can be changed to implement a new protocol or mobility pattern. In addition, all existing modules are highly tunable by many parameters.

A. The WiSEMNet Framework

Christian Nastasi and Andrea Cavallaro [52] proposed a simulation environment for networks for WMSNs, specially networks with sensors capturing complex vectorial data, such as for example video and audio. The proposed simulation environment allows modeling the communication layers, the sensing and distributed applications of a WMSN. This Wireless Simulation Environment for Multimedia Networks (WiSE-MNet) is based on Castalia/Omnet++ and is available as open source to the research community. The environment is designed to be extensible, and has a simple camera model that enables the simulation of distributed computer-vision algorithms at a high level of abstraction. They also demonstrate the effectiveness of WiSEMNet with a distributed tracking application.

B. The M3WSN Framework

Denis Rosário et. al. [53] presented an OMNeT++ framework, called Mobile Multi-Media Wireless Sensor Network (M3WSN). The proposed framework is based on OMNeT++ and Castalia to support multimedia transmission for fixed and mobile scenarios with QoE-awareness. They consider WiSEMNet and WVSN (specific frameworks) based on Castalia and therefore they decide to choose Castalia and integrate the functionalities of WiSE-MNet and WVSN due to their importance for WMSN experiments. Moreover, the M3WSN framework implements full support for delivering, control, and evaluating real video sequences in fixed and mobile scenarios with the aid of mobile traces. They believe that a new set of mobile multimedia-based solutions can be evaluated and optimized with the use of M3WSN.

7. Future WSN/WMSN Simulation Tools

Currently, challenges are facing the simulation of multimedia-based WSN which is not covered by traditional WSN tools. The performance of simulation is a major concern for many tools especially when the objective is to simulate a heterogeneous WMSN with nodes running algorithms that depend on separate video steams or/and with inter-frame calculation. One solution to improve or develop new tools that are capable of making benefit of parallel hardware. The current research of A. N. Kang et al [38] and H. Dutta et al [39] aim to develop parallel calculation simulators for independent sensor nodes simulation. This enables the use of GPUs with many cores and thus, real parallel and independent operations by dedicated GPU cores resolve the slowdown of the execution speed when numerous sensor nodes are used for simulations. This trend is an obvious reflection of the parallel nature of the simulation process where each node is executing independently. Hopefully, this would be considered by future versions of currently available tools, which, indeed, proved to improve the scalability, and performance of WSN/WMSN tools.

8. Conclusions

This paper presented a detailed survey of available WMSN simulation frameworks. First, the challenges and goals of WMSN simulation are defined. Then, the generic model of WMSN simulation is introduced. Also, available tools are presented and compared. Conclusions revealed that available WMSN simulation environments are not sufficient and far efficient. Therefore, dedicated frameworks for WMSN must be targeted by research and development.

Wireless Video Sensor Network (WVSN) simulator that consists of a set of sensor nodes equipped with miniaturized video cameras. They studied the problem of coverage by video sensors in randomly deployed WVSN and focused on the performance of various fast cover set construction strategies for enabling efficient scheduling of nodes in mission-critical surveillance applications. The simulation results show the performance of the various strategies in terms of percentage of coverage, network lifetime, intrusion stealth time and number of intrusion detection.

References

[1] P. Suriyachai, U. Roedig and A. Scott, "A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 14, no. 2, pp. 240-264, 2012.

[2] Y. Zhang, N. Meratnia and P. Havinga, "Outlier Detection Techniques for Wireless Sensor Networks: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 12, no. 2, pp. 159-170, 2010.

[3] I. Butun, S. D. Morgera and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 1, pp. 266-282, 2014.

[4] B. Harjito and S. Han, "Wireless Multimedia Sensor Networks Applications and Security Challenges," in *2010 International Conference on Broadband, Wireless Computing, Communication and Applications*, 2010.

[5] E. Gurses and O. Akan, "Multimedia communication in wireless sensor networks," *Ann. Telecommun.,* vol. 60, no. 7, p. 799–827, 2005.

[6] S. Misra, M. Reisslein and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys & Tutorials,* vol. 10, no. 8, pp. 18-39, 2008.

[7] C. Nastasi and A. Cavallaro, "WiSE-MNet: an experimental environement for wireless multimedia sensor networks".

[8] X. Wang, S. Wang and D. Bi, "Distributed visual-target-surveillance system in wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. B,* vol. 39, no. 5, p. 1134–1146, Oct. 2009.

[9] W.-T. Chen, P.-Y. Chen, W.-S. Lee and C.-F. Huang, "Design and implementation of a real time video surveillance system with wireless sensor networks," in *IEEE 67th Veh. Technol. Conf.: VTC2008 Spring*, Marina Bay, Singapore, May 2008.

[10] S. Sert, A. Yazici and A. Cosar, "Data fusion and processing in Wireless Multimedia Sensor Networks: An analysis for surveillance applications," in *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, 2014.

[11] M. Alaei and J. M. Barcelo-Ordinas, "A hybrid cooperative design for energy-efficient surveillance in Wireless Multimedia Sensor Networks," in *European Wireless, 2012. EW. 18th European Wireless Conference*, 2012.

[12] S. Bouaziz, M. Fan, A. Lambert, T. Maurin and R. Reynaud, "PICAR: experimental platform for road tracking applications," in *IEEE Intelligent Vehicles Symposium*, June 2003.

[13] J. Campbell, P. Gibbons, S. Nath, P. Pillai, S. Seshan and R. Sukthankar, "IrisNet: an Internet-scale architecture for multimedia sensors," in *ACM Multimedia Conference, 2005*, 2005.

[14] F. Hu and S. Kumar, "Multimedia query with QoS considerations for wireless sensor networks in telemedicine," in *Intl. Conf. on Internet Multimedia Management Systems*, Orlando, FL, 2003.

[15] A. Reeves, "Remote Monitoring of patients suffering from early symptoms of Dementia," in *Intl. Workshop on Wearable and Implantable Body Sensor Networks*, London, UK, 2005.

[16] R. Holman, J. Stanley and T. Ozkan-Haller, "Applying video sensor networks to nearshore environment monitoring," *IEEE Perv. Comput.,* vol. 2, no. 4, p. 14–21, 2003.

[17] A. A. K. S., K. Øvsthus and L. M. Kristensen, "An Industrial Perspective on Wireless Sensor Networks — A Survey of Requirements, Protocols, and Challenges," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 3, pp. 1391-1412, 2014.

[18] I. Akyildiz, T. Melodia and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks,* vol. 51, no. 4, pp. 921-960, 2007.

[19] F. Sousa, R. Campos and M. Ricardo, "Energy-efficient wireless multimedia sensor

networks using FM as a control channel," in *2014 IEEE Symposium on Computers and Communication (ISCC)*, 23-26 June 2014.

[20] M. Khanafer, M. Guennoun and H. T. Mouftah, "A Survey of Beacon-Enabled IEEE 802.15.4 MAC Protocols in Wireless Sensor Networks," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 2, pp. 856-876, 2014.

[21] M. A. Kafi, D. Djenouri, J. Ben-Othman and N. Badache, "Congestion Control Protocols in Wireless Sensor Networks: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 3, pp. 1369-1390, 2014.

[22] A. Seema and M. Reisslein, "Towards Efficient Wireless Video Sensor Networks: A Survey of Existing Node Architectures and Proposal for A Flexi-WVSNP Design," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 13, no. 3, pp. 462-486, 2011.

[23] N. A. Pantazis, S. A. Nikolidakis and D. D. Vergados, "Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 2, pp. 551-591, 2013.

[24] S. Mao, Y. Hou, H. Sherali and S. Midkiff, "Multimedia-Centric Routing for Multiple Description Video in Wireless Mesh Networks," *IEEE Network,* vol. 22, no. 1, pp. 19-24, Jan.-Feb. 2008.

[25] S. Sert, A. Yazici and A. Cosar, "Data fusion and processing in Wireless Multimedia Sensor Networks: An analysis for surveillance application," in *2014 22nd Signal Processing and Communications Applications Conference (SIU)*, 23-25 April 2014.

[26] C. Tunca, S. Isik, M. Y. Donmez and C. Ersoy, "Distributed Mobile Sink Routing for Wireless Sensor Networks: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 2, pp. 877-897, 2014.

[27] M. Kennedy, A. Ksentini, Y. Hadjadj-Aoul and G.-M. Muntean, "Adaptive Energy Optimization in Multimedia-Centric Wireless Devices: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 2, pp. 768-786, 2013.

[28] S. Misra, M. Reisslein and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys & Tutorials,* vol. 10, no. 4, pp. 18-39, 2008.

[29] Q. Dong and W. Dargie, "A Survey on Mobility and Mobility-Aware MAC Protocols in Wireless Sensor Networks," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 1, pp. 88-100, 2013.

[30] W. Z. Khan, Y. Xiang, M. Y. Aalsalem and Q. Arshad, "Mobile Phone Sensing Systems: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 1, pp. 402-427, 2013.

[31] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman and W.-C. Wong, "On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 3, pp. 1223-1237, 2013.

[32] M. Conti, J. Willemsen and B. Crispo, "Providing Source Location Privacy in Wireless Sensor Networks: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 15, no. 3, pp. 1238-1280, 2013.

[33] R. Blum and Z. Liu, "Multi-Sensor Image Fusion and Its Applications," in *CRC Press*, 2005..

[34] E. Nakamura, A. Loureiro and A. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Computing Surveys,* vol. 39, no. 3, p. 9/1–9/55, Aug. 2007..

[35] J. Karlsson, T. Wark, P. Valencia, M. Ung and P. Corke, "Demonstration of image compression in a low-bandwidth wireless camera network," in *Int. Conference on Information Processing in Sensor Networks (IPSN)*, 2007,.

[36] S. Rein and M. Reisslein, "Low-Memory Wavelet Transforms for Wireless Sensor Networks: A Tutorial," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 13, no. 2, pp. 291-307, 2011.

[37] A. Schoofs, G. M. P. O'Hare and A. Ruzzelli, "Debugging Low-Power and Lossy Wireless Networks: A Survey," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 14, no. 2, pp. 311-321, 2012.

[38] "Ns-2 project," 2010. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main Page.

[39] "OMNeT++ Project," 2010. [Online]. Available: http://www.omnetpp.org/.

[40] "Truetime project," Lund University, [Online]. Available: http://www.control.lth.se/truetime/.

[41] "Opnet," [Online]. Available: http://www.opnet.com/.

[42] X. Zeng, R. Bagrodia and M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks," in *Proceedings of the 12th Workshop on Parallel and Distributed Simulations*, Banff, Alberta, Canada, 1998.

[43] INET, "INET Framework," 2014. [Online]. Available: http://inet.omnetpp.org/.

[44] MiXiM, "MiXiM Project," 2011. [Online]. Available: http://mixim.sourceforge.net/.

[45] "The Castalia simulation model," [Online]. Available: http://castalia.npc.nicta.com.au/.

[46] "PAWiS - a simulation framework for Wireless Sensor Networks," [Online]. Available: http://pawis.sourceforge.net/.

[47] C. Nastasi and A. Cavallaro, "WiSE-MNet: an experimental environment for wireless multimedia sensor networks".

[48] D. Rosário, Z. Zhao, C. Silva, E. Cerqueira and T. Braun, "An OMNeT++ framework to evaluate video transmission in mobile wireless multimedia sensor networks," in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, 2013.

[49] C. Pham, "A video sensor simulation model with OMNET++," LIUPPA labs, University of Pau, France, December 16th 2011. [Online]. Available: http://cpham.perso.univ-pau.fr/WSN-MODEL/wvsn.html.

[50] "Mannasim: Wireless Sensor Networks simulation," National Council for Scientific and Technological Development (CNPq), Brazil, [Online]. Available: http://www.mannasim.dcc.ufmg.br/.

[51] K. Chih-Heng, "EvalVid: an evaluation tool-set which do realistic simulation for video streaming," 2008. [Online]. Available: http://140.116.164.80/~yufrank/YCY/myevalvid_rtp.htm. [Accessed May 2013].

[52] C. Chen, Z. Li, P. Song and K. Li, "True Time based Simulation for wireless sensor network".

[53] M. Chen, C.-F. Lai and H. Wang, "Mobile multimedia sensor networks: architecture and routing," *Journal on Wireless Communications and Networking,* vol. 159, 2011.

[54] Riverbed, "Riverbed to Acquire OPNET Technologies, Inc.," Oct. 2012. [Online]. Available: http://www.riverbed.com/about/news-articles/press-releases/riverbed-to-acquire-opnet-technologies-inc.html.

[55] P. Levis, N. Lee, M. Welsh and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, 2003.

[56] D. Sumorok, D. Starobinski and A. Trachtenberg, "Simulation of TinyOS Wireless Sensor Networks Using OPNET".

[57] J. Almeida, A. Grilo and P. Pereira, "Multimedia Data Transport for Wireless Sensor Networks," in *Next Generation Internet Networks, 2009. NGI '09*, 2009.

[58] G. Chen, J. Branch, M. J. Pflug, L. Zhu and B. K. Szymanski, "SENSE: A WIRELESS SENSOR NETWORK SIMULATOR," *Advances in Pervasive Computing and Networking,* pp. 249-267, 2004.

[59] A. -. A. I. T. S. &. S. Ltd, "Wireless and Multimedia - Network Simulations," 2014. [Online]. Available: http://www.aditess.com/wireless.html.

[60] H. C. H. W. Hua Wang, "Simulation and Modeling for Centroid Algorithm Using OPNET in Wireless Sensor Networks," *Emerging Research in Web Information Systems and Mining Communications in Computer and Information Science,* vol. 238, pp. 1-10, 2011.

[61] O. D. I. C. E. M. Aykut Yigitel, "Design and implementation of a QoS-aware MAC protocol for Wireless Multimedia Sensor Networks," *Computer Communications,* vol. 34, no. 16, p. 1991–2001, 1 October 2011.

[62] M. Hassan, H. Abid and S. Qaisar, "TRG: Trading rate for good-put in wireless multimedia sensor networks," in *2011 Eighth International Conference on Wireless and Optical Communications Networks (WOCN)*, 24-26 May 2011.

[63] J. W. M. C. D. L. Caifeng Zou, "Simulation Modeling of Cyber-Physical Systems Exemplified by Unmanned Vehicles with WSNs Navigation," *Embedded and Multimedia Computing Technology and Service Lecture Notes in Electrical Engineering,* vol. 181, pp. 269-275, 2012.

[64] F. Österlind, J. Eriksson and A. Dunkels, "Cooja TimeLine: a power visualizer for sensor network simulation," in *SenSys '10 Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 2010.

[65] P. Neves, I. Veiga and J. Rodrigues, "G-JSIM — a GUI tool for Wireless Sensor Networks simulations under J-SIM," in *ISCE 2008. IEEE International Symposium on Consumer Electronics, 2008*, 14-16 April 2008.

[66] N. Dimokas, D. Katsaros and Y. Manolopoulos, "Cooperative caching in wireless multimedia sensor networks," in *MobiMedia '07 Proceedings of the 3rd international conference on Mobile multimedia communications*, Brussels, Belgium, Belgium, 2007.

[67] A. Asim and S. Tixeuil, "XS-WSNet: Extreme scale wireless sensor network simulation," in *2010 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 14-17 June 2010.

[68] D. D. a. V. T. Biljana Stojkoska, "N-Queens-based Algorithm forMoving Object Detection inDistributed Wireless Sensor Networks," *Journal of Computing and Information Technology - CIT,* vol. 16, no. 4, p. 325–332, 2008.

[69] R. Alageswaran, O. Kiruthiga, T. Keerthika and B. Prakash, "Design and Development of Routing Protocol for WSN Simulation in GloMoSim," *Journal of Artificial Intelligence,* vol. 6, pp. 181-186, March 01, 2013.

[70] M. Chitnis, A FRAMEWORK TO SUPPORT WIRELESS MULTIMEDIA SENSOR NETWORKS, Pisa, Italy: Thesis: Doctor of Philosophy (PhD), Scuola Superiore Sant'Anna, January 2011.

[71] A. N. Kang, H.-W. Kim, L. Barolli and Y.-S. Jeong, "An Efficient WSN Simulator for GPU-Based Node Performance," *International Journal of Distributed Sensor Networks,* vol. 2013, no. 1, September 2013.

[72] H. Dutta, T. Failler, N. Melot, B. Pottier and S. Stinckwich, "An execution flow for dynamic concurrent systems: simulation of WSN on a Smalltalk/CUDA environment," in *SIMPAR '10*, Darmstadt, Novembre 2010.