OBIWAN: wireless sensor networks with OMNET++

E. Egea-López*, F. Ponce-Marín, J. Vales-Alonso,
A. S. Martínez-Sala, J. García-Haro
Departament of Information Technologies and Communications,
Polytechnic University of Cartagena, Spain.
* Corresponding author. Address: Campus Muralla del Mar, 30202, Cartagena, Spain.
phone: +34 968 326 553, fax: +34 968 325 973, e-mail: esteban.egea@upct.es

***Abstract*— Simulation is essential in WSN study. However, the nature of WSN makes it an unexpectedly complex task. The extremely large number of nodes, the need for an environment model, and the cross-layer dependecies of the models are some of the reasons for this complexity. Many of the existing simulation tools do not properly handle these issues. In this paper, OBIWAN, a new simulator for WSN is presented. In its design, critical issues like reusability, scalability and cross-layer dependencies have been addressed. The results on its capabilities and performance are provided through a case study involving several thousand nodes and an environment model.**

***Index Terms*— Simulation, Wireless Sensor Networks**

I. Introduction

In recent years extensive research has been conducted on Wireless Sensor Networks (WSN). WSN are formed by a large number of resource-constrained and inexpensive nodes, which has an impact on protocol design and network scalability. In addition, other important factors arise:

- The operation of the protocol layers are usually driven by physical sensor measurements. Hence, a model of the environment under study is needed.
- Energy is a primary concern, because nodes usually run on non-rechargeable batteries.

Most of the community has chosen simulation for their study. However, this natural approach also brings unexpected complexity, which is caused by several issues. First, the large number of nodes reduces simulation performance and scalability. Second, new aspects, inherent in WSN, must be included in simulators, e.g. a physical environment and an energy model, leading to different degrees of accuracy versus performance. Finally, cross-layer implementation of actual sensor protocols requires additional mechanisms and support from simulation frameworks.

Nowadays, two types of simulators for WSN can be found: classical network simulators (ns-2, J-Sim, Ptolemy) and specifical tools (TOSSIM, ATEMU), with their particular advantages and drawbacks [1]. In short, specific tools provide an extremely high fidelity but they are usually bound to a single platform (e.g., TinyOS/Mica2) and do not scale well. The availability of ready-to-use models has usually been the major advantage of classical tools. However, there is not a dominant approach for WSN and the proposed protocols differ greatly from classical protocols such as TCP/IP. This fact makes practically useless the library of available models for classical tools. Moreover, they lack accurate environment or battery models.

Even so, classical tools may be extended for WSN simulation. In fact, there exist such extensions for ns-2, the most widely used network simulator [1]. However, ns-2 object-oriented design introduces too much interdependence between modules, which makes the addition of new models difficult. Realizing this problem, other approaches [1], [2] have chosen a component-based design to promote *reusability* and *extensibility*. An additional point to be addressed is that information that would be isolated in the OSI model needs to be exchanged in sensor networks (cross-layer interdependence). Therefore, an efficient mechanism to share information between modules is necessary.

In this paper, we describe the structure and main design decisions of a new WSN simulator, called OBIWAN, developed with the OMNET++ simulation framework [1] and show preliminary results on its performance. Our simulator addresses critical design factors as *reusability and extensibility*, *scalability* and *cross layer dependencies*. Several WSN have protocols have been implemented and are available, together with an environment model. A case study is also presented: it involves several thousand nodes using a WSN MAC protocol (S-MAC [3]), and an application layer that reacts to the events generated by an environment model.

The rest of this paper is organized as follows, section II describes briefly other simulators. Section III provides the structure of the simulator, with a description of the implementation of its main blocks. In this section, the design decisions concerning the previously mentioned problems are discussed. Section IV shows the case study and the results obtained in the testbeds. Finally, section V concludes and describes possible future works.

II. Related Work

A detailed evaluation of simulation tools for WSN is available in our previous paper [1]. Available tools are classified into general-purpose tools as ns-2, J-Sim, JiST, OMNET++ and specific tools as TOSSIM or ATEMU. Our work extends a general-purpose framework, OMNET++ and takes advantage of its outstanding features. In comparison to similar

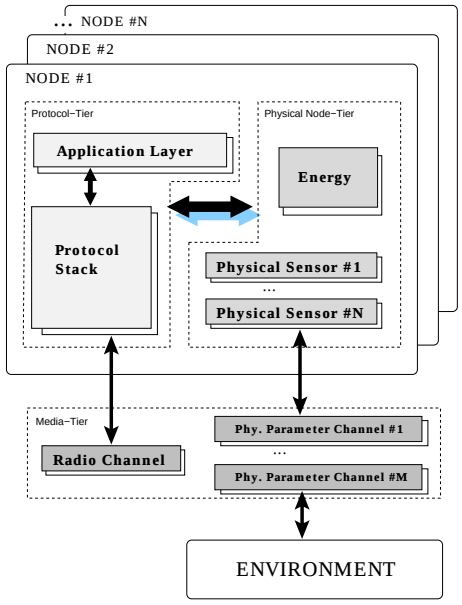

Fig. 1. General WSN model

approaches, OBIWAN offers: a cleaner way of handling cross-layer dependences than ns-2 and better performance than J-Sim, since it is implemented in Java, which renders a 41% worse execution time than ns-2 one [1].

Reference [2] discusses design decisions on the implementation of a WSN simulator. There, the need for a component-based design is justified. Our work, as well as using components, adds proved optimizations [4] to increase scalability and a cross-layer information exchange mechanism based on the *publisher/subscriber* paradigm.

III. Simulator Structure and Design Decisions

The general model for WSN used is shown in Fig. 1. OBIWAN has been developed within the OMNET++ framework. Its main modules are:

- Control. It internally includes a Manager, a Position and a StaticRouting module. The Manager collects statitistics and controls the simulation. Every other module in the simulation can be accessed through the public methods of Manager. Thus, all the modules have a pointer to Manager. Position assigns coordinates (e.g., following a normal distribution) to nodes at initialization. StaticRouting implements centralized routing algorithms. As can be seen, Control is in charge of any centralized processing. Every component of this module may be replaced with extended versions and new components can be easily added.
- Radio Channel. It includes the Propagation and Selector components. The former computes propagation losses (e.g. following a path-loss model). The Selector decides the nodes for which the propagation losses are computed. This way, interference and propagation models are decoupled and a number of optimizations can be seamlessly implemented in this module (see later in this section).
- Node. The communication stack, the energy (battery), mobility and sensor modules are part of this component. A basic class for all components, that just initializes and takes a pointer to the Manager, is available. Every new model implemented derives from the basic class.
- Environment. It generates physical events that will be captured by the sensors in nodes. This module reproduces the behaviour of the desired physical magnitude. It has two components: an event generator and an event manager.

Let us explain how the previously mentioned critical factors for WSN simulation are addressed in OBIWAN:

Reusability and Extensibility. OMNET++ is made upon a clear component-based design. Our simulator takes advantages of this feature of the framework. Every component in the model can be easily replaced just by editing the simulation configuration file. Moreover, other useful features of OMNET++ are available: script-based network configuration, XML model description support, a powerful GUI that simplifies debug and even its parallel simulation support. All of them make this framework interesting for a variety of users. For instance, it can be used as a highly descriptive educational tool due to its graphical capabilities .

Scalability. The extreme number of nodes involved in WSN makes scalability a major challenge in simulation. Computation of propagation losses in wireless simulation is the one of the most limiting factors of scalability. The worse case shows when a stochastic channel model is used or interferences are fully taken into account. In these cases, for every node and every packet transmission a reception event must be scheduled and propagation losses must be computed. However, the disk model and threshold model are common assumptions, which only need to compute losses for nodes within a given transmission range. Under these assumptions some optimizations to quickly search the set of affected nodes can be used [4]. OBIWAN uses the Selector component to implement them. Different propagation models (free space, path loss) can be seamlessly combined with these optimization techniques, providing flexibility to trade-off accuracy and scalability. Moreover, new optimization mechanisms can be readily investigated. OMNET++ implements, in addition, reference counting, which reduces the amount of memory needed for scheduling radio packets.

Cross-layer dependences and shared information. In WSN it is not clear *who* and *when* needs information. OBIWAN uses the *publisher/subscriber* paradigm to deal with cross-layer shared information. A component registers as an event listener of the modules from which needs information. Thus, modules inform of changes in certain internal variables by publishing events. For instance, the energy component is an event listener of the physical layer. Thus, every time the physical layer changes its state (e.g., it turns to sleep mode), it becomes informed and can drain the corresponding current from the battery. Let us note that this mechanism incurs a minimum overhead if no event listener is registered. Extending the information offered by a component is just a matter of deriving new event types.

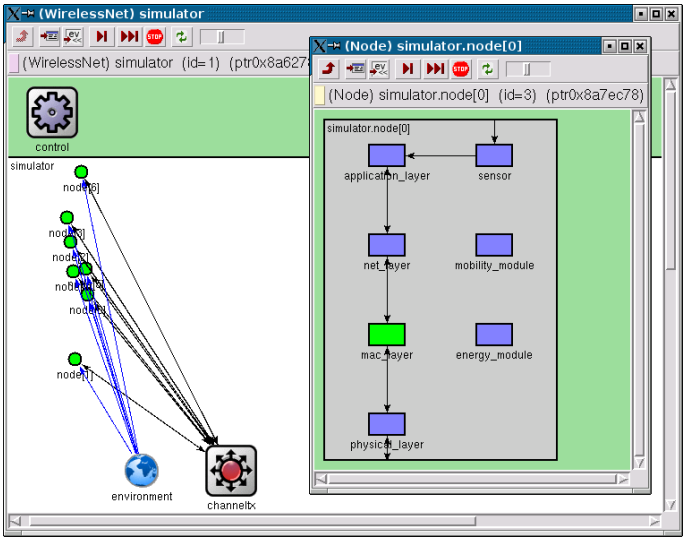

Fig. 2. Simulator OBIWAN

IV. Case Study

The OBIWAN simulator has been used to reproduce the deployment and operation of a wireless sensor network. Nodes use the S-MAC protocol. Routing is fixed statically by a MST (Minimum Spanning Tree) algorithm when the system is initialized and does not change during the simulation. Radio range is only computed at initialization. Afterwards, a list of nodes in range is used to deliver packets by the Selector (a classical disk model). In this setting, nodes are deployed according to an uniform distribution function over an area of 1000x1000 meters. The node closest to the center of the deployment area is selected as sink. Simulations last one day. There is a transient period of 30 minutes before collecting statistics. Physical events are scheduled according to an uniform distribution function between 0 and T' seconds We define the following parameters:

- R. It is the influence radius of the physical events. It gives us the area within which sensors can sense the physical event.
- δ. Average number of nodes per square meter.
- β. Number of nodes that sense, in average, a physical event.

From these parameters, we obtain the equation 1.

$$\delta = \frac{\beta}{\pi R^2} \tag{1}$$

We set R=35 meters guided by the experimental data obtained in table I. In this table it is shown the number of nodes for a given value of R and β. If the network is not connected, the value is in bold letters. A value of R=35 meters gives a reasonable number of nodes per simulation, which also shows that OBIWAN can seamlessly handle simulation of several thousands of nodes. It also makes the average number of nodes in range be above the maximum allowed by S-MAC (20 nodes). β ={2,5,10} have been selected because they provide a connected network.

We have simulated 3 scenarios varying the value of the parameter τ, defined in equation 2.

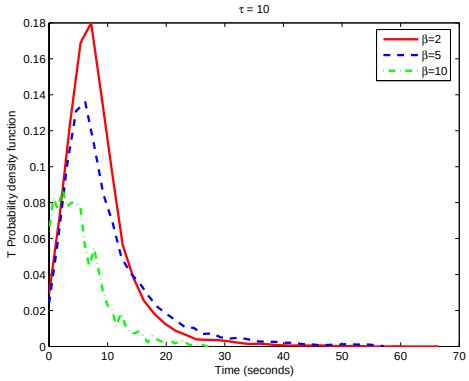

Fig. 3. Results for τ=10

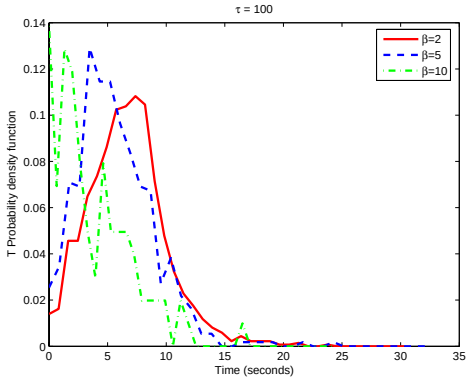

Fig. 4. Results for τ=100

$$\tau = \frac{T'}{Tc} \tag{2}$$

In equation 2, Tc is the duration of a cycle in protocol S-MAC (in our simulations, this value is 1.0278 seconds, with a duty cycle of 10%). Varying τ, we aim to study S-MAC behaviour when the frequency of generation of physical events changes. We have selected τ={1,10,100}. Moreover, for all the three scenarios, we have varied the number of nodes according to the indicated values of parameter β, and we have looked at the evolution of the network depending on the node density. The collected measurements of the following metrics are shown:

- T. Time interval from sensing a physical event to his arrival at the sink node. Figures 3 y 4 show its probability density function.
- ELP (Event Loss Probability). The probability of losing a physical event either because it has not been sensed (depends on the way the nodes are deployed), or because it has been lost in the network (full buffers, frames with errors, etc.). Table II.
- Latency. Time interval from generation of an event to its arrival at the sink.

A. Discussion

Two kind of conclusions can be extracted from this case study: those concerning the protocol under study and those

β **R**	**R=10**	**R=15**	**R=20**	**R=25**	**R=30**	**R=35**	**R=40**	**R=45**	**R=50**
β**=1**	3183	1415	796	509	354	**260**	**199**	**157**	**127**
β**=2**	6366	2829	1592	1019	707	520	398	314	**255**
β**=5**	15915	7074	3979	2546	1768	1299	995	786	637
β**=10**	31831	14147	7958	5093	3537	2598	1989	1572	1273

TABLE I

NUMBER OF NODES VERSUS PARAMETERS β AND R

	τ=1			τ=10			τ=100		
	β**=2**	β**=5**	β**=10**	β**=2**	β**=5**	β**=10**	β**=2**	β**=5**	β**=10**
ELP	0.8289	0.8892	0.9187	0.1466	0.5261	0.8909	0.1464	0.6682	0.9396

TABLE II

ELP

	τ=1			τ=10			τ=100		
	β**=2**	β**=5**	β**=10**	β**=2**	β**=5**	β **=10**	β**=2**	β**=5**	β**=10**
Max. T	1558.59	2987.87	1327.37	75.845	149.351	64.0851	23.8647	36.5096	16.5097
Mean T	397.007	722.513	143.554	8.947	10.45	5.716	6.656	5.688	3.688

TABLE III

LATENCY STATISTICS

Nodes	Basic model		+ Environment	
	simsec/sec	RAM [Mb]	simsec/sec	RAM [Mb]
500	30.16	17.77	26.15	17.9
1000	12.69	29.76	11.65	29.89
5000	1.834	122.6	1.761	122.7
10000	0.7973	233.7	0.7687	234.8

TABLE IV

PERFORMANCE RESULTS FOR DIFFERENT MODELS

applying to OBIWAN capabilities.

Regarding the protocol performance, it is shown that S-MAC does not handle properly this communication pattern. As can be seen, ELP and latency are too high, because of the high load in the network, that S-MAC cannot handle. Too many nodes are sensing simultaneously the events and try to communicate them. The underlying contention-based mechanism and low duty cycle used do not allow an efficient communication in a bounded time interval.

Regarding OBIWAN capabilities, it is shown that WSN simulation with a large number of nodes is possible in a flexible way. It also illustrates that a variety of scenarios can be easily simulated and different statistics collected. Moreover, to provide an estimate of its performance and scalability, another experiment has been conducted. A basic model, without environment and physical events, has been compared with the model of the previous case study. Table IV shows that the enviroment model used has a minimum influence on the simulation scalability. It also shows OBIWAN overall performance.

V. CONCLUSIONS AND FUTURE WORK

A new simulator for WSN has been presented. Its structure and main design decisions have been discussed. The results of a case study involving several thousand nodes show OBIWAN capabilities and preliminary performance results. A performance comparison with other simulation frameworks is left as future work. Other WSN protocols are being implemented at this moment, as well as battery, radio and environment models.

VI. ACKNOWLEDGMENTS

This work has been funded by the Economy, Industry and Innovation Council, with the SOLIDMOVIL project (2I04SU044), by Fundacion Seneca both from the Region of Murcia with the ARENA Project (00546/PI/04) and by the Spanish Research Council with the ARPaq project (TEC2004-05622-C04-02/TCM) and CSI-RHET project (TEC2005-08068–C04-01/TCM).

REFERENCES

[1] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Mariño, J. Garcia-Haro, "Simulation tools for wireless sensor networks", in *Proc. Int. Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2005)*, Philadelphia, PA, pp. 559–566, July 2005.

[2] Chen, G., J. Branch, M. J. Pflug, L. Zhu, B. Szymanski, "SENSE: A Sensor Network Simulator". *Advances in Pervasive Computing and Networking*, B. Szymanksi and B. Yener (eds.), pp. 249–267, Springer.

[3] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks", *ACM/IEEE Transactions on Networking*, vol. 12, pp. 493–506, 2004.

[4] V. Naoumov, T. Gross, "Simulation of large ad hoc networks", in *Proc. ACM Modeling, Analysis and Simulation of Wireles and Mobile Systems (MSWiM 2003)*, San Diego, CA, pp. 50–57, 2003.