Wireless Sensor Network Modular Node Modeling and Simulation with VisualSense

V. Roselló, J. Portilla, Y. E. Krasteva, T. Riesgo
Centro de Electrónica Industrial, Universidad Politécnica de Madrid
{victor.rosello, jorge.portilla, yana.ekrasteva, teresa.riesgo}@upm.es

Abstract—**Wireless sensor networks (WSNs) have been studied deeply during the last years, but real deployments with thousands of motes are really critical in cost, time and reliability due to several problems: commissioning, debugging, etc. In this context, simulation is becoming mandatory to reduce the potential failures of the WSN before deployment, which is very costly. In this paper, a simulation approach for WSNs nodes is presented, based in the Ptolemy simulation environment, specifically in VisualSense. The simulation is carried out over specific hardware, a modular platform for WSNs, whose main feature is the modularity. Therefore, the simulation is also modular, making the process very flexible to modifications of the HW to be simulated and to the operation conditions.**

Index Terms—**Wireless sensor networks, simulation, modular architecture, Ptolemy, VisualSense.**

I. INTRODUCTION

WIRELESS sensor networks (WSNs) have been studied during the last decade in many topics related to them, ranging from the hardware (HW) of the nodes to security, low data rate to low power consumption communication protocols [1][2][3][4].

The target of a WSN is to take measures from the environment in a massive way, in order to monitor and sometimes act over this environment. The advantage of such a system is the possibility of measuring in difficult access points or, even more, measuring during a long period of time, processing the information and transmitting it to a central control point.

The WSN is composed of nodes whose HW has a critical impact in the system behavior. These nodes are capable of measuring, processing and communicating information to the network, and small size and smart behavior are requirements. They have to be autonomous because the network deployment is a very difficult task (sometime the network is composed of maybe hundreds or thousands of nodes). Therefore, it is very important to plan the deployment before the deployment itself is carried out because once the network is set it will be very costly to change a node or a set or nodes if the system crash.

Therefore, a pre-deployment stage must be taking into account. This stage is the simulation stage. Simulating the network in a model of environment can help enormously to obtain satisfactory results after WSN deployment.

Several simulators for sensor networks have appear in the state of the art, some of them are specific for a project and some of them are more generic. These simulators will be detailed in section II.

In the present paper a simulation framework for WSNs nodes is presented, based on the VisualSense simulation environment. The node used for modeling and simulating is a modular platform developed in the Centro de Electrónica Industrial. The details about this platform are commented in section III and the aspects about the simulation procedure using VisualSense for the modular platform in section IV.

II. WIRELESS SENSOR NETWORKS SIMULATORS

In this section an overview of simulation tools for WSNs will be done. Finally VisualSense [21][22] will be described, highlighting its main features.

General purpose network simulators like [6][7][11] may not be the most suitable option in order to get the best performance in the simulation of WSNs, because they are not ready to model the special characteristics of the sensor node like energy consumption, hardware models, mobility of the nodes, etc. With the aim of reproduce these special features, they were found a lot of dedicated tools for WSNs modeling and simulation.

The main requirements taken into account in the present work, when the simulation tool was chosen, were the next:
1) Open source solution: the simulation kernel should be based on open-source elements to avoid further costs if the deployment tool results in a commercial product.
2) Easy to integrate in a customized tool: input and output data formats.
3) Used by an active scientific and/or technical community.
4) Scalable to different sizes and different abstraction levels.
5) Availability of models or easy development of models of the following items:
   a. Network models
   b. Node / mote models
   c. Energy / power models
   d. Environment models
6) Consideration of application specific aspects, like sensors and traffic in the network.
7) Availability of model tools for mobile elements.

On the other hand, functional requirements of the simulator are also identified:
1) It must be capable of showing the amount of data that a specific node has to manage, to preview (and overcome) possible congestions in the network, which could lead to data losses
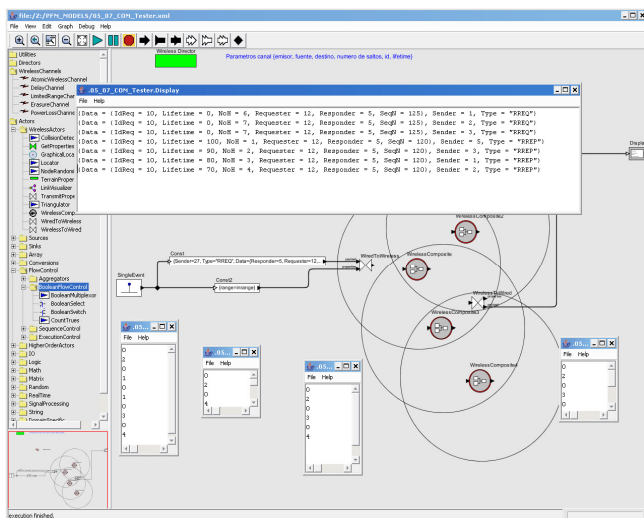
Fig. 1. VisualSense application window example

Fig. 2. Modular platform for wireless sensor networks, modeled and simulated in the present work.

2) The simulator should have communication protocols models, or provide the capability to implement different communication protocols, in order to get routing information and to be able to calculate traffic rates in the whole net and in each node.

3) The simulation tool must manage models based in finite state machines and discrete events modeling, in order to model the nodes in a functional approach. Therefore it will not be necessary to use application code or specific operative system.

4) The environment has to be modeled in a simple way. It must be able to mange models based on signal attenuation.

With all these aspects in mind, one of the main requirements of the simulator is that it contains "radio models" that are able to define the routing of the messages. Besides, it must be able to accept basic models (maybe expressed as state machines or automata) for the network elements.

Considering all these basic requirements, a search of the simulation tools in the state of the art was made. Following it is related the main reasons to the rejection of the simulation tools to, finally, warrant the selection of the VisualSense [20] platform as modeling/simulation environment.

There are a lot platforms based in a concrete operating system or specific type of node like [9][12][13][18]. Many other tools are not under maintenance or development anymore like [10][15][16][17], or they have some license fee [14][18].

VisualSense is an open source platform for "*Visual modeling for wireless and sensor network system*" [20]. It is built on top of Ptolemy II, the software framework for modeling, simulation, and design of concurrent, real-time, embedded systems of the Ptolemy project. It is a long life project and VisualSense is available as part of Ptolemy II since 2004.

VisualSense provides base models that can be modified using Java, or using the models provided to create new *composited actors*, models created in the GUI (graphic user interface), by sub-classing the elements provided in the framework. It also has the capability of combine models with behaviors defined in different domains (continuous time, discrete event, synchronous dataflow, finite state machines, etc).

Because of all these reasons, it was selected like the best suited framework to model and simulate a modular platform, which provides a lot of flexibility to simulate different abstraction level models for the same platform or to change easily the models for the different components of the hardware (power supply, sensors, communications, etc).

III. MODULAR PLATFORM FOR WIRELESS SENSOR NETWORKS

The WSN node which is modeled for simulation in the present work is the one presented in [5] (see "Fig. 2"). This node was designed with the modularity concept in mind.

Modularity allows dividing and encapsulating the functionalities included in the node. Therefore, future redesigns may involve only part of the platform, which is desirable considering the time to market. Moreover, researching using this platform turns open, due to the node flexibility, which makes possible the proof of several concepts minimizing the effort.

Modularity not only provides advantages like those commented in the previous paragraph but allows modeling the different layers in an isolated way. This is very useful because new design for the HW lead to new model for the simulation, but only the new layer must be studied.

The modular platform is composed of four main layers: processing, communication, power supply and sensors (more layers can be added, like memory). Every layer carries out a specific task, which encapsulates the functionality. In the following paragraph these layers are described briefly:

1) Sensors: This layer includes those elements which are intended to take measures from the environment. Today, 3 different layers have been developed for the modular platform. These layers include sensors of acceleration, temperature, humidity, light, infrared and deformation. These sensors supply data though analog and digital interfaces, depending on the type of sensor

2) Processing: This is the brain of the modular platform. It includes an 8052 core compliant microcontroller (uC) from Analog Devices (ADuC841) and a Xilinx XC3S200 Spartan 3 partially reconfigurable FPGA. The uC
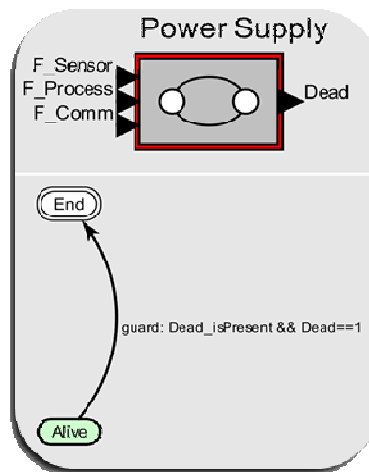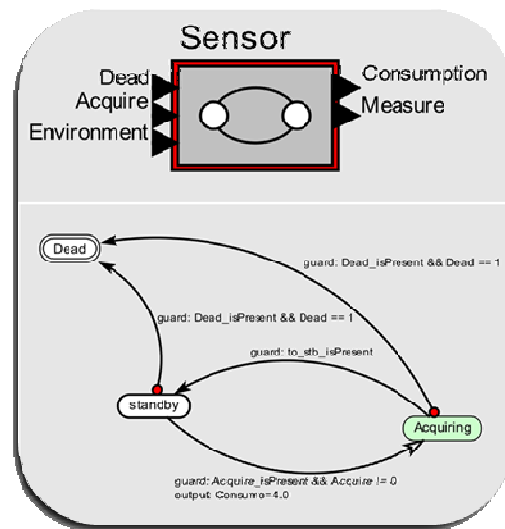
Fig. 3. Power Supply Model: Interface and FSM

Fig. 4. Sensor Model: Interface and FSM

manages communications and data from sensors with analog interface whereas the FPGA process the data from sensors with digital interface. Communication: The last version of this layer includes a low power low data rate ZigBee module (ETRX2 from Telegesis). This module is controlled by the uC through the UART port through an ATI commands interface. A layer version with Bluetooth is also available.

3) Power supply: This layer generates all the voltages needed within the modular platform. Three versions have been developed, the latest with a solar panel.

One of the features of the sensor layer is that it can include both digital and analog sensors. Signals from analog sensors are connected to the ADC of the uC. On the other side, signals from digital sensors are connected to the FPGA. In principle, the FPGA carries out all the processing related to digital sensors, to release the uC, which manage the communications and processes analog sensors.

Due to the modularity of the HW node, each layer can be modeled in a separated way, in order to be simulated. In this context, future simulations with new layers for the platform will require only modeling the new layer, so the process is simplified very much.

Every layer has been modeled as a state machine, and is described section IV.

IV. MODELING AND SIMULATING THE MODULAR PLATFORM FOR WIRELESS SENSOR NETWORKS

According to [24] the models based in Finite State Machines (FSMs), are a very good solution to create models for discrete events system, like WSNs.

As it was described in the previous section, the modular platform is composed of four main HW layers.

The models created for each layer will be presented next (power supply, sensor, processing and communications).

A. Power Supply:

As it is shown in "Figure 3", the Power Supply layer has been modeled using two basic states (Alive, End).

1) *Alive*:
   This state models the situation in which the power supply has energy available for the node. When a message arrives from one of the other models by the input ports, the remaining energy is updated. When an event makes the energy go sub zero, a dead message is send by the *Dead* ports, driving all the models to their *dead* State, and the Power supply model to its *End* state.

2) *End*:
   This is the final state, modeling when the system is out of battery. All the possible messages received in the input ports are rejected.

B. Sensor:

The sensor model is based in a three state FSM, as appears in "Figure 4". These states are (Acquiring, Standby and Dead).

1) *Acquiring*:
   In this state, sensor model reads the Environment input message, and send the Measure to the Processing model and the total consumption of the sensor model (it depends on the type and number of sensors being modeled).

2) *Standby*:
   Represents when the sensor layer is low power mode, when a message from de Processing model (in the Acquire port) arrives, the FSM goes to Acquire state, and sends its consumption rate to the Power Supply model.

   3) *Dead*:
   When a Dead message (from Power Supply mode) is received, the FSM evolves to this state from every other state. Describes when there is no battery in the node.

C. Processing:

The processing layer model is based on a four states FSM model. In "Figure 5" it is shown the basic diagram. The states used to simulate the processing layer are Standby, Sensing, Send and Dead.
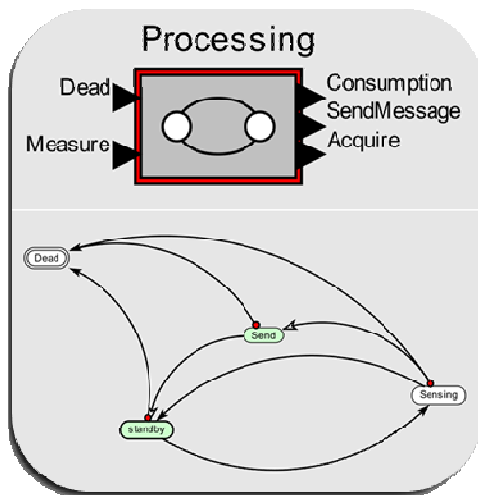
Fig. 5. Processing Model: Interface and FSM
(no transition values to get a more comprehensible diagram)

1) *Standby*:
   It simulates the processor low power consumption mode. When a wake up event is produced (by the processing layer itself), the model send a message with it consumption rate to the Power supply models, and goes to the Sensing state.

2) *Sensing:*
   Here the processing layer has send a Acquire message to the Sensor layer and is waiting for the Measure message. When it arrives, the FSM change to the Send state. An update power supply message with the new consumption rate is send to the Power Supply model.

3) *Send*:
   The Send state describes the operation carried by the processing layer to send a message. Basically is wake up the communication layer, and wait till the operation to be completed, that depends on the number of messages to be send and network status, (number of messages traveling in the neighborhood and possible transmission problems). When the operation is completed, a sleep message is send to the Communication model, just if a Sleepy End Devices (SED) is being modeled. Otherwise the communication layer of the node never goes to low power mode.

4) *Dead*:
   It shows the same situation of Dead state in the Sensor model, when there is no battery in the node.

   D. *Communications:*

   The communications layer, has the most complex layer, because it has to manage the interactions with the rest of the nodes, that is mainly create and maintain routes following as much as possible Zigbee specifications. The states used to simulate the communication layer are Free, CSMA, Sleep, Dead, RoutingMessage, ACKMessage and DataMessage.

1) *Free*:
   In this state a message is read and depending on the type of message (routing message, data message or ACKMessage) different actions will be carried out:
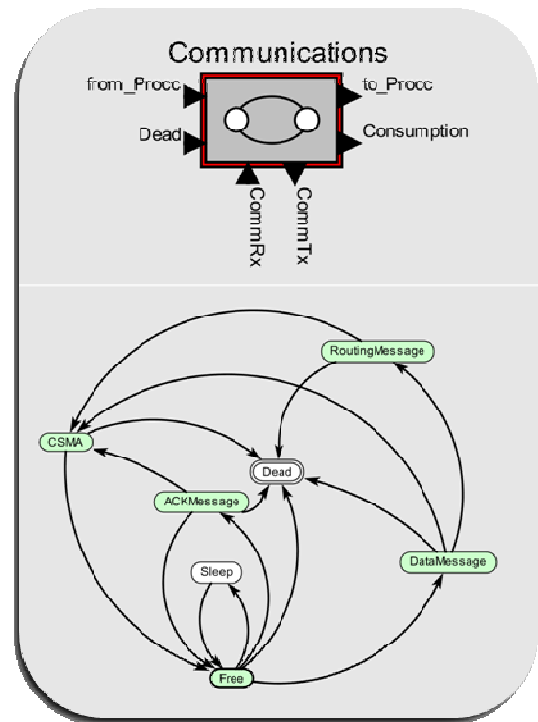

Fig. 6. Communications Model: Interface and FSM
(no transition values to get a more comprehensible diagram)

- Routing message: The message is processed (routing table update) and a suitable reply is sent if necessary.
- Data message: If the receiver is known the data is sent to the next node because the route is known as well or if not a root request message is sent.
- ACKMessage: To process the reception of an Acknowledge or to send an Acknowledge when it is required by the sender.

2) *CSMA*:
   In this state the medium access control for ZigBee (non beaconing networks) is modeled.

3) *Sleep*:
   In this state the low power state of the communication HW module is modeled. When a message from processing is received, the FSM jump to the Free and a message is sent to the Power Supply to update the remaining energy.

4) *Dead*:
   The FSM jump to this state when the Power Supply informs that no energy is available any more.

5) *RoutingMessage*
   This State is used to simulate the behavior of a node during a route creation process. The routing protocol used is the *Ad Hoc on Demand Distance Vector Routing* [25].

6) *ACKMessage:*
   When a message is received, even if the node is not the final destination of the message must be send an *ACKMessage* to notice the sender the correct reception of the message.
   When an *ACKMessage* is received, the message in acknowledged is pushed from the send queue

7) *DataMessage:*
   If a *DataMessage* is received and the node is the

destination, the data must be sent to the Processing Layer. Also and *ACKMessage* would be send to the originator node if requested.

If a *DataMessage* is order from the Processing Layer, it must be created, if a route to the destination is not known a route creation process will be started.

V. CONCLUSIONS

In the present work a modeling and simulation framework for modular nodes in wireless sensor networks has been presented. It has been showed the flexibility of the procedure derived from the modularity of the hardware, which implies modularity in the models.

All the models follow a theoretical approach, according to previous experience using the modular HW platform.

The features of VisualSense have been tested and validated, demonstrating its feasibility for modeling and simulation of WSNs modular nodes. The environment is general and open allowing easy development of models for new systems.

The modeling and simulation environments for real deployments, is currently being tested, using the hardware platform and measuring the actual performance of the simulation with real measurements.

The model created for the *AdHoc On Demand Distance Vector Routing* will be added as an example of routing protocol in next version of *VisualSense*. Nevertheless the model is already available for download and test as part of the beta version of the *Ptolemy II* release [25].

REFERENCES

[1] S. Yamshita, T. Shimura, K. Aiki, K. Ara, Y. Ogata, I. Shimokawa, T. Tanaka, H. Kuriyama, K. Shimada, K. Yano, "A 15x15, 1 µA, Reliable Sensor-Net Module: Enabling Application-Specific Nodes," *in Proc. of the 5th IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN'06)*, April 2006, pp. 383–390.

[2] J. Polastre, S. Szewczyk, D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research", *in Proc. of the 4th IEEE/ACM International Conference on Information Processing in Sensor Networks (IPSN'05),* April 2006, pp. 364-369.

[3] Xiaojiang Du; Hsiao-Hwa Chen, "Security in wireless sensor networks", *IEEE Wireless Communications*, vol.15, no.4, pp.60-66, August 2008

[4] M.H.F. Ghazvini, M. Vahabi, M.F.A. Rasid, R.S.A.R. Abdullah, W.M.N.M.W. Musa, "Low Energy Consumption MAC Protocol for Wireless Sensor Networks," *Second International Conference on Sensor Technologies and Applications, 2008 (SENSORCOMM'08)*, pp.49-54, August 2008.

[5] J. Portilla, A. de Castro, E. de la Torre, T. Riesgo, "A Modular Architecture for Nodes in Wireless Sensor Networks", *Journal of Universal Computer Science (JUCS)*, vol. 12, nº 3, March 2006, pp. 328-339.

[6] G.F. Riley "The Georgia Tech Network Simulator", In *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools For Reproducible Network Research*, Karlsruhe, Germany, August 2003.

[7] The ns Manual (formerly ns Notes and Documentation) 2009.

[8] Mannasim Framework Classes Manual, The Manna Research Group, 2006. (Mannasim).

[9] P. Levis, N. Lee, M. Welsh, D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications", In *Proceedings of the 1st international Conference on Embedded Networked Sensor Systems* (SenSys '03), Los Angeles, California, USA, November 2003.

[10] Rimon Barr , Zygmunt J. Haas , Robbert van Renesse, "JiST: an efficient approach to simulation using virtual machines" Research Articles, Software—Practice & Experience, v.35 n.6, p.539-576, May 2005. (JiST/SWANS)

[11] G. F. Riley, 2003. "The Georgia Tech Network Simulator". In *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools For Reproducible Network Research*, Karlsruhe, Germany, August 2003.

[12] B. L. Titzer, D. K. Lee, J. Palsberg, "Avrora: scalable sensor network simulation with precise timing", In *Proceedings of the 4th international Symposium on information Processing in Sensor Networks*, Los Angeles, California, USA, April 2005.

[13] J. Polley, D. Blazakis, J. McGee, D. Rusk, J. S. Baras, "ATEMU: a fine-grained sensor network simulator," *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 145-152, October 2004.

[14] Chen, G., J. Branch, M. J. Pflug, L. Zhu and B. Szymanski (2004). "SENSE: A Sensor Network Simulator", *Advances in Pervasive Computing and Networking*, Springer: 249-267.(SENSE)

[15] Sundresh, Sameer, WooYoung Kim, Gul, Alpha, "SENS: A Sensor, Environment and Network Simulator," in *Proceedings of the 37th Annual Simulation Symposium (ANSS'04)*, Arlington, VA, USA, April 2004.

[16] S. Park, A. Savvides, M. B. Srivastava, "SensorSim: a simulation framework for sensor networks", In *Proceedings of the 3rd ACM international Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Boston, Massachusetts, United States, August 2000.

[17] Y. Wen, R. Wolski, G. Moore, "Disens: scalable distributed sensor network simulation". In *Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, San Jose, California, USA, March 2007.

[18] G. Simon, P. Volgyesi, M. Maroti, A. Ledeczi, "Simulation-based optimization of communication protocols for large-scale wireless sensor networks," *Proceedings of IEEE Aerospace Conference*, vol.3, pp. 3_1339-3_1346, March 2003(Prowler)

[19] S. P. Fekete, A. Kroller, S. Fischer, D. Pfisterer, "Shawn: The fast, highly customizable sensor network simulator", *Fourth International Conference on Networked Sensing Systems (INSS '07)*, pp.299-299, June 2007. (SHAWN)

[20] P. Baldwin, S. Kohli, E. A. Lee, X. Liu, Y. Zhao, " VisualSense: Visual Modeling for Wireless and Sensor Network Systems," Technical Memorandum UCB/ERL M05/25, University of California, Berkeley, CA 94720, USA, July 15, 2005.

[21] R. Gonzalez. "Modeling Network Lifetime in a WSN using Visual Sense", poster presented at the 8th Biennial Ptolemy Miniconference, April 2009.

[22] D. Vieira, "Simulating the Calculus for Sensor Networks in VisualSense", poster presented at the 8th Biennial Ptolemy Miniconference, April 2009.

[23] T. Antoine-Santoni, J. Santucci,E. De Gentili, and B. Costa, 2008. "Discrete Event Modeling and Simulation of Wireless Sensor Network Performance". *Simulation 84 Issue 2-3*, pp.103-121, February 2008

[24] C. E. Perkins, E. M. Rogers, "Ad hoc On-Demand Distance Vector Routing." In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

[25] Ptolemy subversion repository
http://chess.eecs.berkeley.edu/ptexternal/