An Energy Consumption Model for a Wireless Sensor Network Node based on the division of the Duty Cycle

José. M. Alcalá, and Álvaro Hernández
Department of Electronic
University of Alcalá
Alcalá de Henares, Spain
jmanuel.alcala@depeca.uah.es

Victor Cionca, Michael Hayes, and Brendan O'Flynn
WSN Group
Tyndall National Institute
Cork, Ireland
victor.cionca@tyndall.ie

***Abstract*— As the energy consumption in an energy harvesting wireless sensor network is constrained, applications that can run on the nodes are limited as well. Thus, it is not only important to know how much energy can provide a harvesting system, but also how much energy will consume a specific application running on a node, and match both models to achieve a functional system from an energy point of view. Much literature can be found regarding energy harvesting models to predict energy produced but little about how this energy will be used. This paper presents an energy consumption model for nodes in a Wireless Sensor Network (WSN), which depends solely on the Duty Cycle. Likewise, applications are characterized for their duty cycle and this parameter will determine if a particular application can run on the node. The novelty of this model is the division of the duty cycle into smaller duty cycles for each of the WSN application components, allowing more control over the energy consumption. To the best of our knowledge, no other related studies have been carried out so far and we believe that this approach will contribute to develop applications for energy constrained system according to a maximum duty cycle. This approach has been tested against real data showing a relative error up to 3.14%.**

Keywords-wireless sensor networks; energy consumption model; energy harvesting; efficient energy; duty cycle; WSN applications

I. INTRODUCTION

The deployment of large Wireless Sensor Networks (WSN) provides greater granularity over different parameters (temperature, humidity, light intensity, Carbon diOxide (CO_2) levels, occupancy, etc.) in commercial and residential buildings. Consequently, this increases the control and monitoring of systems such as Heating, Ventilation, and Air Condition (HVAC), lights, doors and windows and so on. However, the maintenance of WSN can be a tedious process as batteries need to be replaced and, in most cases, sensor devices are located in places with difficult access. Hence, energy harvesting methods are widespread studied to provide long-term energy to autonomously power a mesh sensor network with a large number of nodes.

Energy harvesters can use sources from within the building such as indoor lights and vibrations, but all these sources generate relatively low and variable power, which is difficult to use. Developing low power hardware and protocols is no longer enough. The node must be aware of the amount of energy it consumes to be able to plan future activities against predicted levels of generated energy. For instance, having indoor lights as source, the power management must predict that, at night, lights are off and, consequently, reduce network activities to the minimum during this period to keep the nodes alive. Because the duty cycle determines the network activity, the energy consumption in a WSN node has a tight relation to this parameter.

Our contribution is an approach to an Energy Consumption Model based on an in-depth analysis of the duty cycle. Instead of considering a black-box model where a unique node duty cycle determines the energy consumption, our model splits the duty cycle into smaller duty cycles where each of them is attached to a specific activity performed by the node. Thus, the discretization of the duty cycle allows a greater granularity over the energy consumption and the customization of the activities depending on the available and expected energy.

To validate the model, several tests have been carried out, divided into two different experiments: firstly, an energy consumption model for a single node communicating directly to the base station has been developed and evaluated; then, the model has been adapted for a node as part of a WSN. Devices used for these tests are Tyndall motes transmitting on 2.4 GHz running TinyOS and using the Collection Tree Protocol (CTP).

In the next section, related work will be presented, followed by the experiments and modeling of the energy consumption in a node; two cases are considered: in section 3, a node communicating directly to the base station and in section 4, a forwarding node in a WSN. Finally, a discussion about the model is given in section 5, with future work and conclusions closes the paper.

II. RELATED WORK

Reducing the power consumption of wireless sensors or extending their lifetime has always been at the focus of the research community. This is handled across the levels of the platform, with low-power hardware, efficient Medium Access Control (MAC) protocols [1] and [2] that minimise the amount of time the radio is kept active, routing protocols [3] that attempt to even out the energy consumption throughout the network or data aggregation techniques [4] that reduce the overall amount of network traffic.

Energy harvesting technology [5] adds a new dimension to power management: the variability of the power source. This must be efficiently used by updating the system performance: increased during high energy periods and reduced otherwise. There are many approaches in literature that address the control of system duty cycle based on available power. Possibly the most comprehensive is by Kansal, Hsu, Zahedi and Srivastava [6], who model both the energy input and the behaviour of the node as a duty cycled load. The power supply is modelled as a bursty energy source and a storage element with leakage current. In an older paper, Kansal and Srivastava [7] propose a power management framework that could perform task scheduling based on the energy input and usage model. However, it is unclear whether the complete framework was implemented and validated. Vigorito, Ganesan and Barto [8] also address the problem of managing the duty cycle of the node, but relying only on past values for harvested energy. Their algorithm is based on adaptive control theory.

In [9], Fan, Zheng and Sinha approach the problem of data rate management in energy harvesting powered multi-hop networks. They present algorithms to determine data rates for each node, based on expected traffic flow. The algorithm works best in static topologies.

The ideal is being able to predict the power consumption of a node, based on its hardware and firmware. This would allow network deployers to determine how long the node will function, and if the lifetime is not satisfactory, to update parameters in the firmware in order to extend it. There are many power consumption models that follow closely the hardware platform (power consumption per clock cycle) [10] and physical radio layer [11]. PixieOS [12] is a WSN OS that uses resource-aware programming in order to increase the efficiency.

All of the papers enumerated here consider the node a single duty cycle load, and discuss how that duty cycle can be modified. A WSN application has many intrinsic duty cycles and changing the overall duty cycle is non-trivial. One could impose a specific duty cycle at low levels in the software stack but it would affect the functionality of not only the node, but the rest of the network. This paper proposes a power consumption model that takes into account the duty cycles of the power hungry firmware components. By changing those duty cycles, the power consumption will be modified and the functionality of the node and network will not be compromised.

III. ENERGY CONSUMPTION IN A SINGLE NODE

As a first approach, the model was tested in a simple scenario where an isolated node communicates directly to the base station without forwarding any packets. Therefore, the application has less components and less complexity. Further, developing this model allows isolating the node's own activities from the activities of maintaining the topology and routing packets.

A. *Energy Consumption Model*

In a scenario where the node communicates directly to the base station, "send" and "sense" activities are the main tasks performed. Consequently, the energy consumption can be expressed as follows:

$$E = E_{sense} \text{ x } D_{sense} + E_{tx} \text{ x } D_{tx} + E_{sleep} \text{ x } (1 - D_{sense} - D_{tx}) \quad (1)$$

The total energy consumption for a single node can be estimated by the sum of the different components of the application and these components are determined by their duty cycle. Thus, E_{sense} is the energy consumed every time the node is sensing and it is empirically determined as well as E_{tx}. D_{sense} is the duty cycle for sensing, *tx* refers to the send component and *sleep* to the sleep modes of the node's microcontroller.

This model was tested against real data and results are presented in the following section.

B. *Validation of the model*

For testing the model, a single node is set to communicate directly to a base station. The application running on the node senses light intensity with a T_{sense} period and transmits directly to the base station an average of each S samples. A power analyser is used to measure the real power consumption of the node.

Three different tests have been carried out, lasting ten minutes each, to evaluate the model for different values of the sense and send duty cycles, as follows:

- Test 1: D_{sense}=0. 34% and D_{tx}=0. 19%
- Test 2: D_{sense}=2.75% and D_{tx}=0. 39%
- Test 3: D_{sense}=0.017% and D_{tx}=0.002%

In Test 3, the activity is very low so it will show how the model fits when the sleep mode is dominant.

A simulation of the model was done to compare it to the real data. In Figure 1, the evolution of the energy consumption shows that the model fits quite well with the real data. There are two main sources of error: CPU activity for handling timer overflows – these are the smallest spikes in the real data plot; and a hardware irregularity causing a slight "bump" in the current consumption right after sending a message. Both could be easily integrated into the model but they are platform specific and it is more valuable to achieve a general model. The accumulative energy consumption plots in Figure 2 shows that the real energy consumption is linear and it can therefore be modeled using a linear equation as shown in (1), where the duty cycle determines the slope of the figure. The difference between slopes of the

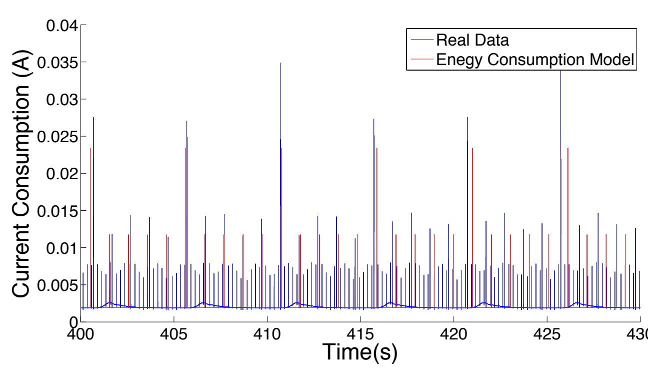

Figure 1. Simulation for Test 1. D_{sense}=0.0034 and D_{tx}=0.0019

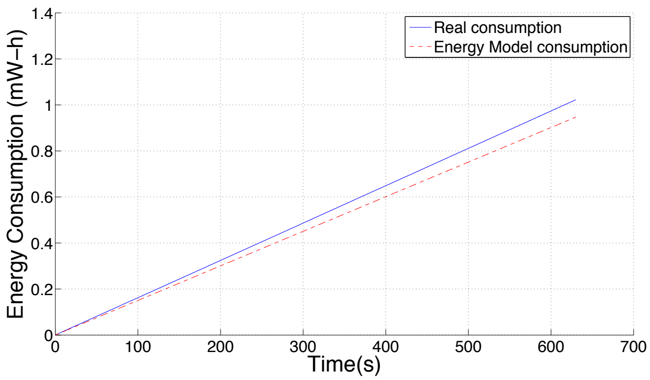

Figure 2. Acumulative Energy Consumption for Test 1.

real data and the model is due to the two error sources indicated above. An analysis of the total energy consumption based on duty cycles is shown in Figure 3. For each test, the total real consumption is compared to the energy consumption model, which is divided in the consumption of the different application components. Thus, it can be observed that the main consumption is due to the sleep mode, which is not the common case on low-power devices but, in this case, as the goal is to obtain an accurate model of the energy consumption and not a low-power system, this aspect was not taken into account. The sleep consumption should be a known constant, which can be modelled with a minimum error.

The model seems to fits perfectly on Test 3, as duty cycles for sensing and sending are negligible and almost all the consumption is due to the sleep mode. For test 2 and 3, it can be observed how the energy consumption for sending and sensing proportionally increase as their duty cycles increase as well. Especially test 2 shows the importance of the analysis of the duty cycle that the model carries out, although the energy consumption for a single sending activitiy is 2 times higher than a sensing activity, the overall is a higher energy consumption on the sensing mode as the duty cycle is 7 times bigger than the duty cycle for the sending mode. The relative error is shown on top of the bars.

IV. ENERGY CONSUMPTION IN A WSN NODE

For a node in a Wireless Sensor Network, the previous model does not fit with the real consumption because there are other important activities to take into account: maintaining the topology of the network and forwarding network traffic. In this section, the energy consumption model for a WSN node will be presented, considering the main activities by their duty cycles and then, the model will be tested against the consumption of a real node in a network.

A. Energy Consumption Model

A generic application in a WSN node performs the following major activity cycles:

- Periodically checking for incoming packets at the MAC layer: switch on the radio, check for incoming packets, switch off the radio.
- Sending **periodic beacons** at the routing layer to maintain the network topology.
- **Forwarding packets** of upstream neighbours towards the sink, at the routing layer
- Performing its own periodic activities: sensing, sending packets, etc.

From these, the MAC cycle, routing beacon and application activity cycle are predetermined, hard-coded in the node's firmware. Therefore they can be controlled. The only one that cannot be controlled is the rate of packet forwarding, as this depends on the network dynamics.

In a static WSN with a static routing tree, the packet forwarding rate could be determined for each node, based on the structure of the network sub-tree routed at that node and the activity cycle of the application components, as shown in (2) where the number of forwarded packets is proportional to the time (t) (with t_o as the initial time) and T_{app} and N_{sons} are the period of the application and the number of sons for that node, respectively. Therefore, in a static WSN it should be possible to provide an accurate estimate of the network´s energy consumption depending on the different components of the application, as it shows in the (3). E_{mac}, $E_{beacons}$ and E_{app} are predetermined and empirically obtained and stand for the energy of the periodic MAC check, periodic beacons and forwarding packets of a duty cycle; D_x stands for the duty cycle of the component X and P_{tx} is the probability of successful packet transmission.

$$Packets_{(fwd)}(t) = 1/T_{app} \text{ x } N_{sons} \text{ x } t + Packets_{(fwd)}(t_o) \quad (2)$$

$$E = E_{mac} \text{ x } D_{mac} + E_{beacons} \text{ x } D_{beacons} + E_{app} \text{ x } D_{app} \text{ x } 1/P_{tx} + E_{app} \text{ x } D_{app} \text{ x } (1+1/P_{tx}) \text{ x } N_{sons} + E_{sleep} \text{ x } [1 - D_{mac} - D_{beacons} - D_{app} - (1+1/P_{tx}) \text{ x } N_{sons} \text{ x } D_{app}] \quad (3)$$

The energy consumption model for a static network will be applied to a real-life, dynamic network when the network is stable, and it will be tested against real data to evaluate the error of the approximation. To simplify the model, the probability of successful packet transmission will be set to

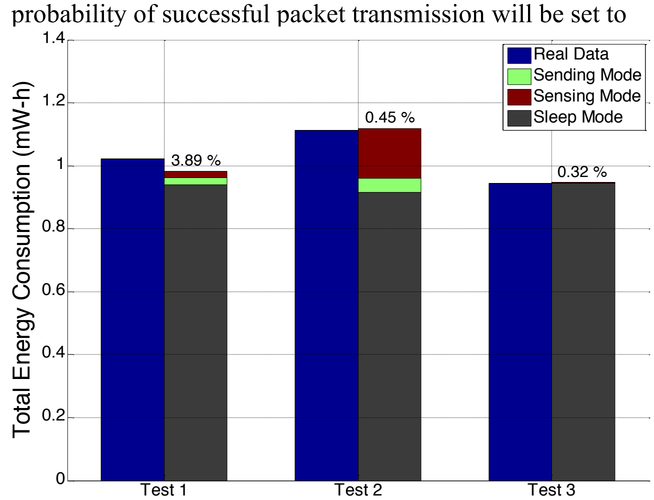

Figure 3. Comparison of the total energy consumption between the real data and the energy consumption model spliting into its components.

one. To consider any possible congestion in the network, a congestion factor "C" for forwarding packets will be introduced in (3). This is a probabilistic factor that resizes the energy consumption model depending on the network congestion. When the one of the nodes is congested its forwarding buffer fills up and the node starts dropping packets. This means that while sons keep sending packets, the parent node will not spend energy forwarding them and this has to be considered in the model. The congestion factor is the complement of an occupancy factor, which represents the rate of forwarding packets divided by the maximum rate that can be performed; the rate is the number of descendants for that node divided by the application activity period and the maximum rate is represented by the inverse of the MAC check period. Combining all together, the congestion factor is shown in (4).

$$C = 1 - [N_{sons} \times T_{mac} \;/\; T_{app}] \quad (4)$$

The final Energy Consumption Model obtained after these modifications is as shown in (5). This will be the model used and tested for validation.

$$E = E_{mac} \times D_{mac} + E_{beacons} \times D_{beacons} + E_{app} \times D_{app} + E_{app} \times D_{app} \times 2 \times C \times N_{sons} + E_{sleep} \times [1 - D_{mac} - D_{beacons} - D_{app} - 2 \times N_{sons} \times D_{app}] \quad (5)$$

B. Validation of the Model

In this section, the energy model consumption stated in (5) will be tested against real data and then a relative error will measure the accuracy of the model. For that, a dynamic wireless sensor network was set up with 6 nodes. After stabilization, the topology is as shown in Figure 4. The energy consumption on the forwarding node 1 has been monitored with a power analyser; node 0 is the base station and the network has a maximum of three hops. The Collection Tree Protocol (CTP) is responsible for routing and the application collects intrinsic network parameters such as "timestamp of the collection", "id of the node", "total number of sent packets", "number of packets sent to node X (array)", "total number of packets forwarded", "number of packets forwarded to node X (array)", "number of forwarding request" and "number of unacked packets". After stabilization, the energy consumption on node 1 was monitored for thirty minutes. I_{mac}, I_{sleep} $I_{beacons}$,and I_{app} were

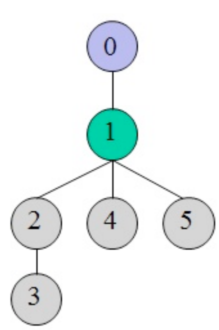

Figure 4. Network topology after stabilization.

empirically determined. Considering that the length of a beacon pulse must be at least equal to T_{mac} as beacons are broadcasted and they should be listened by all nodes in the network, the duty cycle for every component on the WSN application is as follows:

- $D_{mac} = 4.75\%$, $D_{beacons} = 1.6\%$, $D_{app} = 0.95\%$ and $D_{fwd} = 2.57\%$

Where D_{fwd} is 2 x C x N_{sons} x D_{app}, C is 0.34 and N_{sons} is 4.

Equation (2) was modeled and tested against real measurements after stabilization. Figure 5 shows the number of packets forwarded by node 1 and node 2 (forwarding nodes on the network) during the experiment. The error is very low and the model fits well with the real data, so it can be stated that (2) is valid and forwarded packets in a dynamic network can be approximately predicted using a model to predict forwarded packets in a static network when the environment is stable. Therefore, the duty cycle for forwarded packets in a dynamic network can be roughly estimated as well as the energy consumption produced by this component.

Figure 6 is a comparison of the real energy consumption and the energy predicted by the Energy Consumption Model. The predicted energy is disambiguated by its duty cycle, with each main activity represented on the model. Again, the major energy consumption is in sleep mode (it can be changed for low power devices), then the periodic medium check, forwarding packets (which is saturated by the congestion factor as the network is congested), sending beacons and the own application activity (which is very low due to simplicity of the tested application: it collects some parameters and sends them to the parent). The figure gives more detailed information about the energy consumption than if only the active mode and the sleep mode were considered.

Figure 7 is an indicative of the possible congestion on the network. It shows the number of unacknowledged packets on the different nodes. Node 3 has a large amount of unacknowledged packets that could be a consequence of dropped packets at Node 2; the same can be said about nodes 2, 4 and 5. Therefore, it may be possible that Nodes 1 and 2 are congested and the congestion factors need to be applied. Finally, to validate the model, the relative error between the model and the real data must be determined. Figure 8 compares total energy values measured and modeled. The

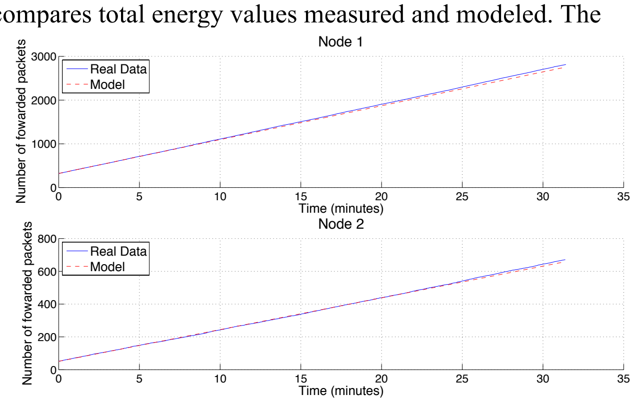

Figure 5. Number of forwarded packets per node and per time.

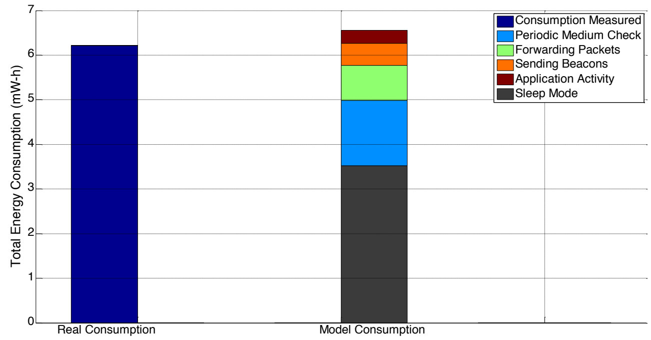

Figure 6. A comparison of the real consumption and the Energy Model Consumption by duty cycles, for node 1.

model equation presented on (5) has a relative error of 5.4 %. Also, the experiment was simulated using Matlab/Simulink based on (5). The congestion factor was modeled introducing the occupancy factor in a Bernoulli binary generator block. With this, the relative error drops to 3.1408 %. This simulation gives more detailed information about the error evolution over the time, Figure 8. Absolute error is linearly dependent on time, however, the relative error tends to be constant (around 3.2%).

V. DISCUSSION

A more precise energy control can be carried out using the energy consumption model presented in this paper. Thus, WSN applications can be customized and modeled to satisfy energy constraints decreasing duty cycles of each application component. Figure 6 gives a good indication of what consumes the most current and therefore it is possible to adjust the functionality of application components to reduce it. For instance, in this case, it can be stated that, in order to reduce energy consumption, the sleep consumption should be reduced but also the periodic medium check duty cycle should be reduced as the network is stable and the energy consumption for this component is considerably high. The application activity is low enough so there is no need of adjusting it.

However, there is always a trade off on reducing the energy consumption: decreasing the MAC duty cycle, the network will have more latency and might break down; likewise, reducing the beacon duty cycle, the network will take longer to adapt to changes. Therefore, it is needed to highlight here that the energy consumption model presented in this paper does not aim to reduce or to give a fine-grained measure of the energy consumption but to discretize the energy consumption to determine what component consumes the most and how the consumption is distributed.

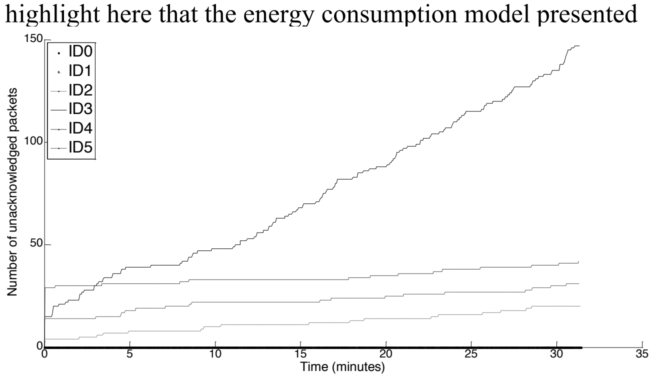

Figure 7. Number of unacknowledged transmissions per node.

TABLE I. COMPARATIVE TOTAL ENERGY OF THE EXPERIMENT FOR A WSN NODE.

	Total Energy comparative		
	Energy Consumption (mW-h)	***Absolute Error (mW-h)***	***Relative Error (%)***
Real Data	6.219	-	-
Model Equation	6.5553	0.33637	5.4088
Model Simulink	6.4143	0.19532	3.1408

VI. FUTURE WORK

External factors such as symbolic interference, noise and multipath effect have a strong effect over the network communications and topology and, consequently over the energy consumption. In future work, a study of external factors will be carried out in order to include them in the energy consumption model. These factors will be modeled using Link Quality Indicator (LQI), Ambient Noise and Received Signal Strength Indication (RSSI). Also, deploying the network, it should be considered that, depending on the distance, links between two nodes could be in one of three statuses: connected region, transitional region and disconnected region. The packet reception rate (PRR) becomes quite random on the transitional region so, for future work, it is needed to avoid this region [13].

VII. CONCLUSIONS

An energy consumption model for a WSN node based on the disambiguation of the duty cycle has been achieved and presented on this paper. The division of the energy consumption in different components provides a better understanding of the internal process in nodes and allows network deployers to customize the overall consumption by modifying the functionality of the individual components as well as detect any malfunction that produces overconsumption.

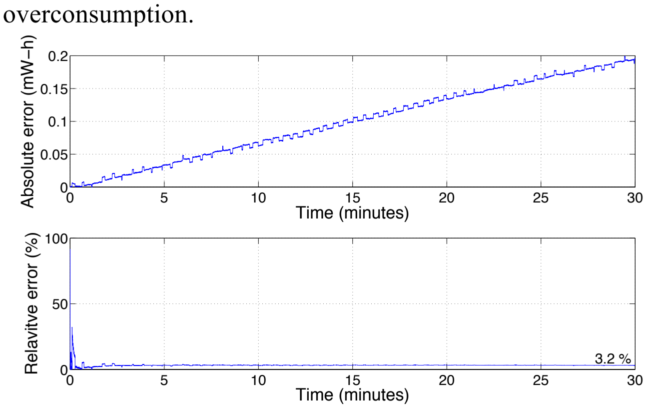

Figure 8. a)Absolute Error over the experiment. b)Relative Error over the experiment. Data measured for node 1.

The error obtained is low enough: using the model equation, the relative error is 5.4%; using a more stochastic process in Matlab/Simulink, the error drops until 3.14%. This model provides a more detailed view of the energy consumption than other models that use only active and sleep duty cycle.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Economy and Competitiveness (LORIS project, ref. TIN2012-38080-C04-01), the FPI/UAH fellowship programs and the Science Foundation Ireland (ITOBO Strategic Research Cluster).

REFERENCES

[1] R. Lin, Z. Wang, and Y. Sun, "Energy efficient medium access control protocols for wireless sensor networks and its state-of-art," *IEEE International Symposium on Industrial Electronics*, May 2004, pp. 669–674 vol. 1.

[2] K. Langendoen, A. Meier, and E. T. H. Zurich, "Analyzing MAC protocols for low data-rate applications". *ACM Transactions on Sensor Networks (TOSN),* August 2010 vol. 7, no. 1.

[3] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," *Proceedings. 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, 2002, pp. 129–136.

[4] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, 2002, pp. 575–578.

[5] R. J. M. Vullers and R. Van Schaijk, "Energy harvesting for autonomous wireless," *Solid-State Circuits Magazine, IEEE,* 2010, pp. 29–38.

[6] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, Sept. 2007, vol. 6, no. 4, p. 32–es.

[7] A. Kansal and M. B. Srivastava, "An environmental energy harvesting framework for sensor networks," *Proceedings of the 2003 International Symposium on Low Power Electronics and Design. ISLPED '03*, 2003, pp. 481–486.

[8] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," *Sensor, Mesh and Ad Hoc Communications and Networks,* June 2007, pp. 21–30.

[9] K.-W. Fan, Z. Zheng, and P. Sinha, "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks," *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, 2008, pp.239-252, 2008.

[10] E. Perla, A. Ó. Catháin, R. S. Carbajo, and M. Huggard, "PowerTOSSIM z : Realistic Energy Modelling for Wireless Sensor Network Environments," *Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks,* 2008, pp. 35–42.

[11] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 2006, pp. 286–295.

[12] K. Lorincz, B. Chen, J. Waterman, G. Werner-allen, and M. Welsh, "Resource aware programming in the pixie OS," *Proceedings of the 6th ACM conference on Embedded network sensor systems,* 2008, pp. 211–224.

[13] N. Baccour, A. K. Aa, L. Mottola, H. Youssef, C. A. Boano, and M. Ario, "Radio Link Quality Estimation in Wireless Sensor Networks : a Survey," *ACM Transactions on Sensor Networks (TOSN),* Sept. 2012, vol. 8, no. 34.