Energy-Efficient Reconstruction of Environmental Data with a Multihop Wireless Sensor Network

Felipe da Rocha Henriques, Lisandro Lovisolo and Marcelo Gonçalves Rubinstein

***Abstract*— In this work, real environmental data are monitored with a multihop Wireless Sensor Network (WSN). An algorithm for energy conservation of sensor nodes is used to extend the network autonomy, increasing its lifetime. Temperature and humidity signals are sensed by nodes and transmitted to a sink node. Both signals are reconstructed at the sink node, with samples that it receives from sensor nodes. The algorithm considers the variation rate of the monitored data, in order to manage the need for communication. Thus, it aims at reducing the amount of transmissions and nodes can sleep between transmissions, to save energy. Simulations are performed, and we observed that both signals could be reconstructed, with a significant decrease in the amount of transmissions and an increase in the network lifetime. Moreover, a high network connectivity is obtained, with a packet delivery ratio greater than 90%.**

***Keywords*— Wireless Sensor Network, Reconstruction, Energy.**

I. INTRODUCTION

Wireless communications have become ubiquitous. The search for mobility and the easy deployment of wireless networks are some of their advantages. These networks can be classified as: Infrastructure, in which nodes are associated with a base station; and ad hoc, with nodes that can directly communicate with each other, without being controlled by the base station [1].

A Wireless Sensor Network (WSN) is a special kind of ad hoc network in which its nodes can collect data, as temperature, pressure or humidity, communicate with each other, and transmit them to a sink node. Sensor nodes have a sensing unit, and three other basic units: A processing unit; a communication unit, for transmission and reception tasks; and an energy unit, composed by a battery. Sensor nodes have an autonomy, operating as long as their batteries have energy [1].

There are several applications for WSN, like: Enemy troops monitoring, sensing of biometric data from patients in a hospital, industrial and home automation, or monitoring of regions like caves or forests [2]. In this work, a WSN is used to sense environmental data: Temperature and humidity. These data were gathered by a WSN located in the Intel Berkeley Research lab, and are available in [3].

Generally, nodes in a WSN are spread in areas of difficult access, and it may be impossible to change nodes batteries, when their energy ends. Thus, strategies to extend the network autonomy, by the conservation of nodes energy, are important in a WSN. Communication is the task that spends more energy in a sensor node [1]. So, it may be advantageous to process data, aiming at reducing the amount of information, in order to save energy. In [4], a survey of energy saving methods for WSNs is presented, including a taxonomy of energy saving schemes.

Two energy conservation algorithms for WSNs are proposed in [5]. We use a WSN to sense and reconstruct a process, and the proposed algorithms consider the variation rate of the process to manage the necessity for communication. Thus, they intent to reduce the amount of transmissions, and nodes sleep between subsequent transmissions, entering in an inactivity state, in order to save energy [6]. The monitored process is reconstructed at the sink, with samples received from sensor nodes.

In [5], the monitored process was simulated using a synthetic-generic function. We intent to extend and validate the results in [5] by using one of the proposed algorithms to sense real environmental data. In this work, we considered the Algorithm 2 proposed in [5], because the results showed that this algorithm may keep the reconstruction error of the monitored process (ε) to be less than a predefined threshold, the acceptable maximum reconstruction error (δ). Moreover, the reconstruction has to be energy efficient, as we intend to increase the network lifetime, which is defined as the time until the energy of the first node ends [7].

This work is structured as follows: In Section II, the algorithm used and the signals to be reconstructed are briefly described; in Section III, the energy model, and simulation aspects are presented; Section IV shows the obtained results; and conclusions are presented in Section V.

II. RECONSTRUCTION OF ENVIRONMENTAL DATA

In this work, an algorithm for energy conservation is applied to sense temperature and humidity signals gathered by a WSN located in the Intel Berkeley Research lab. In this WSN, 54 sensor nodes were sensing environmental data for more than one month.

We define the temperature and humidity signals as functions that depend on the sensor nodes coordinates x_i and y_i, and time t, in which i is the index of a given node S_i, i.e., $s_T(x_i, y_i, t)$ and $s_H(x_i, y_i, t)$, respectively. A node that is deployed close to a light source, for example, tends to measure samples with larger temperature values. Basically, each node measures samples of the signals, and the algorithm uses the variation rate of the monitored signal to estimate the

*Felipe da Rocha Henriques, Lisandro Lovisolo and Marcelo Gonçalves Rubinstein, Programa de Pós-Graduação em Engenharia Eletrônica (PEL), Universidade do Estado do Rio de Janeiro (UERJ), Brasil, *Departamento de Telecomunicações, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), Petrópolis, Brasil. E-mails: *henriquesfelipe@telpet.com.br, lisandro@uerj.br, rubi@uerj.br.

next transmission, and the inactivity period of S_i. The node considers that the process varies linearly, and uses the last two samples to perform the estimation [5].

Moreover, one constraint is imposed: The reconstruction error to be less than a given threshold (δ). Thus, sensor nodes can estimate a future measurement that will be transmitted $\hat{s}_i^t(k+1)$ (and that is assumed to be received at sink node), such that $\varepsilon \leq \delta$, using measured and transmitted samples. The sink uses a first order interpolator to reconstruct the signals, with the samples that it receives from sensor nodes. More details about the estimator are presented in [5].

Sensor nodes are both information sources, measuring samples from the signals, and routers (relays), forwarding packets from their neighbors. It is assumed that S_i has an inactivity period IP_i, during which the node does not measure, process, receive or transmit. Before S_i enters the sleeping mode, it verifies if it has neighbors that use it as a router. If this does not occur, S_i sleeps for $\gamma \times IP_i$ seconds. If there are neighbors that use it as a router, then S_i sleeps for:

$$IP_i = \gamma \times min(IP_i, IP_k), k \in S_{i,\#}, \quad (1)$$

where each IP_k represents the inactivity period of each neighbor of S_i. The sleeping period reduction factor γ ($0 < \gamma < 1$) is used to increase the probability of S_i being awake to forward packets from its neighbors [5].

The algorithm is distributed and it runs directly in the application layer of each sensor node. The flow of the algorithm is presented in **Algorithm 1**, in which n is the current instant of measurement or transmission, EC_i represents the energy of node S_i, m indicates a measured sample, and nodes initially have an inactivity period of 0.1 second, the timebase used in the simulations. Furthermore, line 3 of **Algorithm 1** indicates that it runs while node S_i has energy, in consonance with the adopted concept of lifetime.

Algorithm 1 Algorithm implementation at sensor S_i.

1: $n \leftarrow 1$
2: $IP_i \leftarrow 0.1$
3: **while** $EC_i > 0$ **do**
4: S_i measures $s_i^m(n)$
5: **if** $n = 1$ **then**
6: S_i transmits $s_i^m(n)$
7: **else**
8: S_i transmits $s_i^m(n)$
9: S_i calculates IP_i using linear estimation
10: **if** S_i has packets to forward from # neighbors **then**
11: S_i transmits packets from its # neighbors
12: $IP_i = min(IP_1, IP_k), k \in \#$
13: **end if**
14: S_i sleeps for $IP_i = \gamma \times PI_i$ seconds
15: S_i wakes up after IP_i seconds
16: **end if**
17: $n \leftarrow n$+1
18: **end while**

Figure 1 shows the layout of the WSN of the Intel Berkeley lab, with its 54 sensor nodes [3]. In this network, the (x,y) coordinates of nodes S_{16}, S_{24}, S_{50} and S_{42}, for example, are (1.5,2.0), (1.5,30), (38.5,1.0) and (39.5,30). These coordinates (in meters) are relative to the upper right corner of the lab.

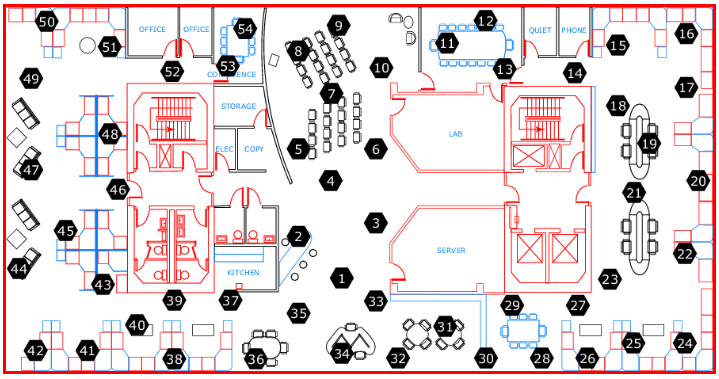

Fig. 1. Layout of the WSN Berkeley lab.

III. Energy Model and Simulations

The energy model used is a state-based model, in which nodes may operate in two states: Inactive, saving its energy, or active. The active state has four operation modes: Measuring, processing, transmission, and reception. The energy model considers the packet payload size and it is based on [8], in which it is observed that there is a linear relation between the energy consumption in the transmission mode and the size of the packet payload.

The energy consumption of a node can be estimated, as a function of the period of time in which the node stays in the different operation modes.

$$\begin{aligned}\hat{E}_C =& t_I \times C_I + t_A \times C_A \\ &+ t_M \times (C_A + C_M) + t_P \times (C_A + C_P) \\ &\quad + t_R \times (C_A + C_R) + t_T \times (C_A + C_T), \end{aligned} \quad (2)$$

in which t_I, t_A, t_M, t_P, t_R and t_T are, respectively, the cumulative sum of the time intervals in which a node remains in inactive and active states, and in measuring, processing, receiving, and transmitting modes. If a node is active, there is an increment in its energy consumption, depending on the task it is performing. The associated consumptions C_s of each one of the states and modes are presented in Table I.

The simulations were performed in TrueTime 1.5 [9], a simulation environment based in MatLab/Simulink and the network standard was the IEEE 802.15.4 [10].

In this work, a multihop communication model is considered, and the Ad-hoc On Demand Distance Vector (AODV) [11] routing protocol is employed.

For the scenario, we use a WSN with fifteen of the fifty-four nodes of the WSN of [3]. We consider the sink node deployed in the same position of node S_{20} of the Berkeley Research lab, in the (0.5,17) coordinate, and this position is fixed in all tests. In each simulation run, the positions of the fourteen sensor nodes are sorted, using the other fifty-three node positions. Each sensor node has temperature and humidity data.

The metrics used to evaluate the algorithm are: The percentage decrease in the amount of transmissions and the network

TABLE I

STATIC SIMULATION PARAMETERS.

Node initial energy (J)	2.00
Transmission power (dBm)	-5
Reception sensibility (dBm)	-66
Radio range (m)	40
C_I: Inactive state Cons. (mJ/s)	1.80
C_A: Active state Cons. (mJ/s)	10.00
C_M: Measuring mode Cons. (mJ/s)	18.00
C_P: Processing mode Cons. (mJ/s)	18.00
C_R: Rx mode Cons. (mJ/s)	62.40
C_T: Tx mode Cons. (mJ/s)	58.62
Payload size (Byte)	1

lifetime increase, both with respect to a network without any kind of energy management scheme; the reconstruction error of the monitored signals, $e_T(x_i, y_i, t)$ and $e_H(x_i, y_i, t)$, for temperature and humidity, defined in equations (3) and (4), in which $s_T(x_i, y_i, t)$ and $s_H(x_i, y_i, t)$ are the monitored signals, $\hat{s}_T(x_i, y_i, t)$ and $\hat{s}_H(x_i, y_i, t)$ are the reconstructed ones; and the packet delivery ratio, the ratio between the amount of receptions and the amount of transmissions in the network, defined in equation (5).

$$e_T(x_i, y_i, t) = s_T(x_i, y_i, t) - \hat{s}_T(x_i, y_i, t), \tag{3}$$

$$e_H(x_i, y_i, t) = s_H(x_i, y_i, t) - \hat{s}_H(x_i, y_i, t), \tag{4}$$

$$PDR = \frac{\text{Received packets}}{\text{Transmitted packets}}. \tag{5}$$

Each simulation was run ten times, for both temperature and humidity signals, and a 95% confidence interval for the mean is used, represented by vertical bars in the graphs. Thresholds (δ) of 1%, 2%, 5% and 8% were considered, allowing maximum reconstruction errors up to these values. Moreover, we consider $\gamma = 0.5$ in the simulations, in order to increase the probability of a node being awoken to forward packets from its neighbors [5].

IV. Results and Discussion

A. Energy conservation

Figure 2 presents the percentage reduction in the amount of transmissions and the lifetime increase of the network, in function of the δ threshold, for the monitored environmental data: Temperature and humidity. Both metrics are evaluated with respect to a network without any energy management strategy. Figure 2 shows that the increase in the threshold leads to greater transmission decrease and an increase in the network lifetime. The increment of δ allows transmissions of samples with greater percentage errors, and nodes can sleep for larger periods of time, which means that less transmissions are done, saving energy.

B. Reconstruction error

Figures 3 and 4 show the Cumulative Distribution Function (CDF) of the reconstruction error of the monitored signals, for temperature and humidity, respectively. The CDF represents the probability of the reconstruction error (ε) being less than a given threshold, i.e., $P(\varepsilon < \delta)$.

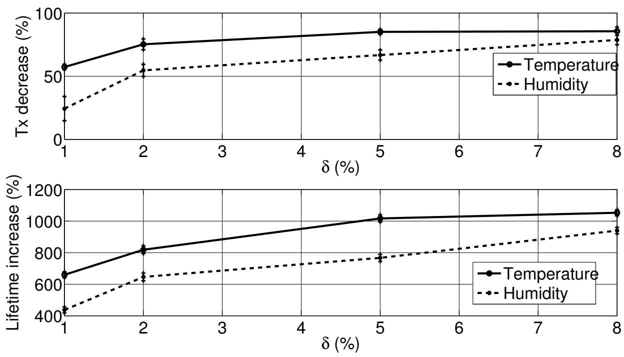

Fig. 2. Transmissions decrease and lifetime increase $\times$ δ.

The increase of δ leads to the increase of the reconstruction error (ε). This behavior is in consonance with results presented in Figure 2. A higher network lifetime can be obtained, but with less transmissions, and this leads to an increase in the reconstruction error, because there will be less samples to reconstruct the signal.

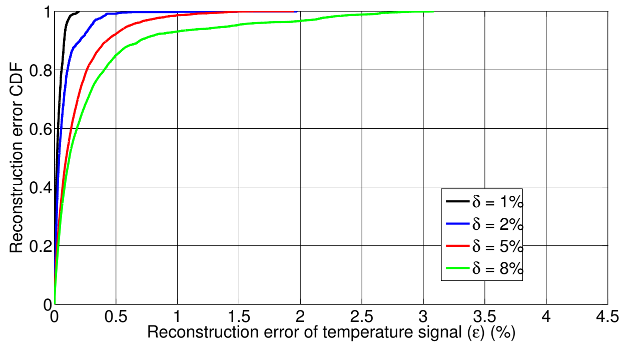

Fig. 3. Reconstruction error CDF of the temperature signal.

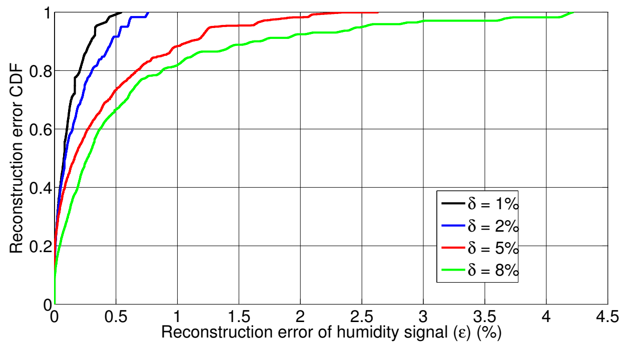

Fig. 4. Reconstruction error CDF of the humidity signal.

As presented before, the algorithm imposes the constraint that the reconstruction error has to be less than the threshold ($\varepsilon \leq \delta$). Figure 5 shows the maximum reconstruction error, in function of δ, for both monitored signals. It can be verified that the constraint is satisfied in all cases.

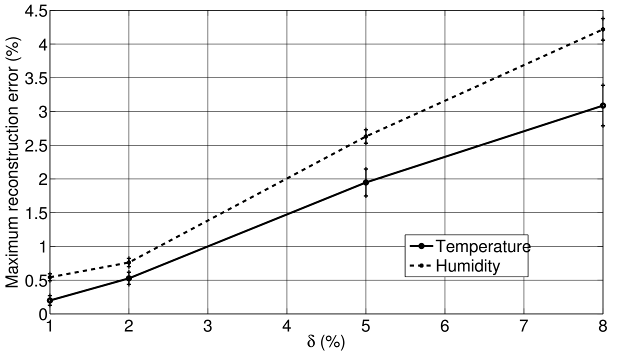

Fig. 5. Maximum reconstruction error $\times$ δ, for both signals.

It can be observed in Figures 3, 4 and 5, that the reconstruction errors obtained for humidity data are greater than the ones obtained for temperature data. In order to investigate this problem, we evaluate the variation rate of the monitored signals. We define the variation rates as the partial derivative of both signals, in function of time:

$$s_T^{'}(x_i, y_i, t) = \frac{\partial s_T(x_i, y_i, t)}{\partial t}, \quad (6)$$

$$s_H^{'}(x_i, y_i, t) = \frac{\partial s_H(x_i, y_i, t)}{\partial t}. \quad (7)$$

Figure 6 shows the variation rate of the monitored signals, for one node (S_1) in one simulation run, for $\delta = 2\%$. Table II presents the average and maximum variation rates of the monitored signals, for all sensor nodes in one simulation run, for $\delta = 2\%$. We verified that the monitored humidity signal presented a variation rate larger than the monitored temperature signal, for both average and maximum values.

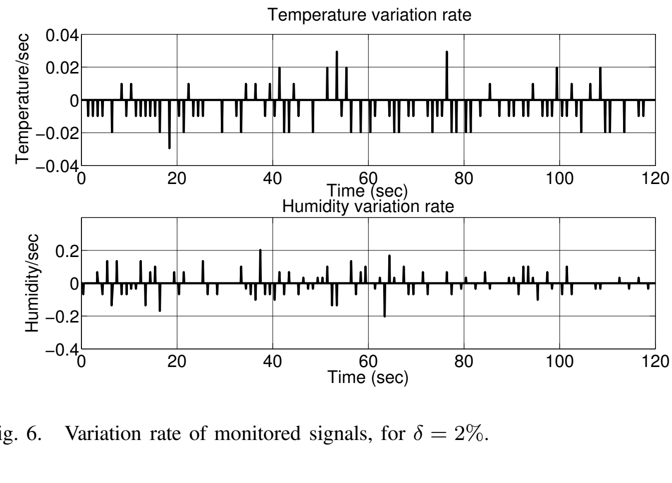

Fig. 6. Variation rate of monitored signals, for $\delta = 2\%$.

The algorithm used in this work in simulations considers the variation rate of the monitored signal. The greater the variation rate of the signal, the higher the number of required transmissions. Thus, as the monitored humidity signal presented a rate larger than the temperature one, there are more transmissions,

TABLE II

VARIATION RATE OF MONITORED SIGNALS, FOR $\delta = 2\%$.

Variation rate	Temperature	Humidity
Average	0.00002	0.00051
Maximum	0.0294	0.2025

in the humidity monitoring case, leading to a less significant increase in the network lifetime, as shown in Figure 2.

Moreover, for signals with lower variation rates, nodes tend to sleep for larger time periods. This behavior can be observed in Figure 7, which presents the inactivity period for node S_1 with $\delta = 2\%$, in one simulation run, and in the Table III, which shows the average and maximum inactivity periods for all nodes in one simulation run.

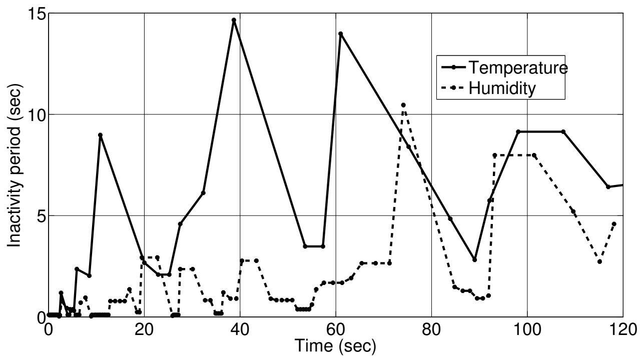

Fig. 7. Inactivity periods for S_1 node with $\delta = 2\%$.

TABLE III

INACTIVITY PERIODS FOR BOTH MONITORED SIGNALS, FOR $\delta = 2\%$.

Inactivity period	Temperature	Humidity
Average	3.7118 sec	1.1659 sec
Maximum	14.6594 sec	10.4643 sec

Figures 8 and 9 show the monitored and the reconstructed temperature and humidity signals, for $\delta = 1\%$ and for $\delta = 2\%$, respectively. We can see that the reconstructed signals closely follow the monitored ones.

C. Packet delivery ratio

Figure 10 shows the packet delivery ratio, in function of δ, for both signals. For larger values of δ, nodes sleep more, and this may impact the connectivity of the network. Moreover, we verified a small decrease in this metric, for fixed thresholds, in the case that nodes were monitoring humidity. As this signal presented a larger variation rate than the other, more transmissions are required, which may cause more dispute to access the medium, and consequently collisions.

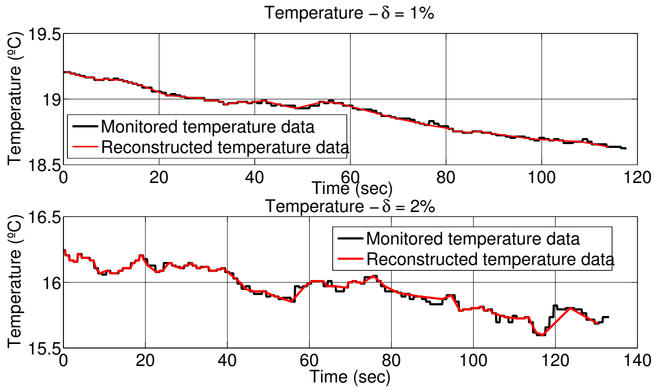

Fig. 8. Monitored and reconstructed temperature, for $\delta = 1\%$ and for $\delta = 2\%$.

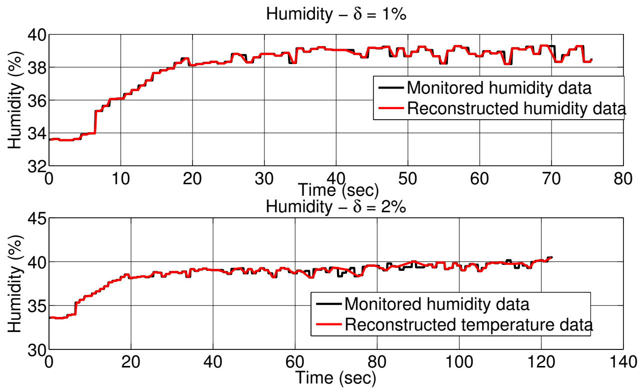

Fig. 9. Monitored and reconstructed humidity, for $\delta = 1\%$ and for $\delta = 2\%$.

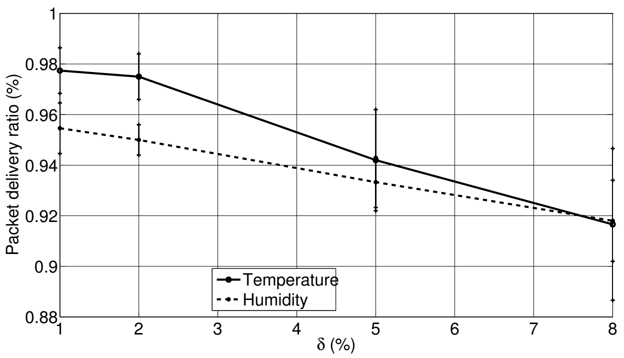

Fig. 10. Packet delivery ratio $\times$ δ.

V. Conclusions

We use a WSN to sense and reconstruct two environmental signals: Temperature and humidity. These signals were obtained from the database available in Intel Berkeley Research lab [3]. An energy conservation algorithm that is distributed and runs directly in the application layer of each sensor node is used. The algorithm considers the variation rate of the monitored signals to make a linear estimation of future measurements and the inactivity periods of sensor nodes. The algorithm also imposes a constraint, in which the reconstruction error of the monitored signals has to be less than a predefined threshold.

We run simulations, considering a WSN with fifteen of the fifty-four nodes of the Berkeley lab, and results show a significant increase in the network lifetime. Both temperature and humidity signals could be reconstructed, with errors less than the threshold. We observed that the humidity signal varied more than the temperature signal. As the algorithm tracks the variation rate of the monitored signal, more transmissions were required, when the WSN sensed the humidity signal. We also evaluate the network connectivity, obtaining packet delivery ratios larger than 90%.

For future works, we intent to implement the considered algorithm in a real wireless sensor network, and validate the results obtained with simulations.

Acknowledgements

This work has been supported by FAPERJ, CNPq and CAPES.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey”, *Computer Networks*, vol. 38, pp. 393–422, 2002.
[2] A. Mainwaring, J. Polastre, R. Szewczy and D. Culler, “Wireless Sensor Networks for Habitat Monitoring”, *ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, 2002.
[3] Intel Berkeley lab wsn, *http://db.csail.mit.edu/labdata/labdata.html*.
[4] V. K. Sachan and S. A. Imam and M. T. Beg, “Energy-efficient Communication Methods in Wireless Sensor Networks: A Critical Review”, *International Journal of Computer Applications*, vol. 39, no. 17, 2012.
[5] F. R. Henriques, L. Lovisolo and M. G. Rubinstein, “Algorithms for Energy Efficient Reconstruction of a Process with a Multihop Wireless Sensor Network”, *Proc. of the 4th IEEE Latin American Symposium on Circuits and Systems*, 2013.
[6] P. S. Sausen, J. R. B. Sousa, M. A. Spohn, A. Perkisich and A. M. N. Lima, “Dynamic Power Management with Scheduled Switching Modes in Wireless Sensor Networks”, *Proceedings of the 15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS’07)*, pp. 1–9, 2007.
[7] Z. M. Wang, S. Basagni, E. Melachrinoudis and C. Petrioli, “Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime”, *Proc. 38th Annual Hawaii International Conference on System Sciences (HICSS’05)*, pp. 3–6, 2005.
[8] Y. Panthachai and P. Keeratiwintakorn, “An Energy Model for Transmission in Telos-Based Wireless Sensor Networks”, *Proceedings of the 4th International Joint Conference on Computer Science and Software Engineering*, 2007.
[9] A. Cervin, D. Herinksson, B. Lincoln, J. Eker and K.-E. Arzen, “How Does Control Timing Affect Performance?”, *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 16–30, 2003.
[10] P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta and Y. F. Hu, “Wireless Sensor Networks: a Survey on the State of Art and the 802.15.4 and ZigBee Standards”, *Computer Communications*, vol. 38, num. 7, pp. 1655–1695, 2007.
[11] C. Perkins, E. Belding-Royer and S. Das, “Ad Hoc On Demand Distance Vector Routing (AODV)”, *RFC 3561*, 2003.