Propagation of Agent Performance Parameters in Wireless Sensor Networks

J.C. Cuevas-Martinez, J. Canada-Bago,
J.A. Fernández-Prieto, and M.A. Gadeo-Martos

Abstract. Wireless sensor networks are composed of resource-constrained sensor nodes, powered by batteries, with limited CPU and memory, and wireless communication. In spite of the fact that sensor nodes are resource-constrained devices, several soft computing technologies have been adapted to them. In order to save battery, sensor nodes work in cycles based on awake and sleep modes. In this work we propose a method, based on a differential decision system, to calculate dynamic parameters that control the awake-sleep cycle in a multi-agent sensor structure and their propagation to other sensor nodes in a network. As an application of the proposed system, a sound pressure monitoring application is presented. Results have shown that the proposed method utilizes less work cycles than continuous measuring systems, saving battery and improving the lifetime of sensor nodes, with a reasonable lost of precision.

Keywords: Resource-constrained devices, multi-agent systems, wireless sensor networks.

1 Introduction

Wireless Sensor Networks (WSNs) [1] are composed of a large number of sensor nodes, powered by batteries, where each node consists of a processing unit with limited computational capability and memory, wireless communication, probes and actuarors. Their range of application is very wide: intelligent agriculture, industrial control and monitoring, environmental monitoring systems, surveillance, health monitoring, traffic monitoring, etc.

In order to prolong the sensor lifetime, sensor nodes usually operate in a work cycle in which they, first, execute the application (measure, calculate outputs, actuate, etc.), decide if it is necessary to connect with a base station or other sensors,

J.C. Cuevas-Martinez · J. Canada-Bago · J.A. Fernández-Prieto · M.A. Gadeo-Martos
Telecommunication Engineering Department
University of Jaen, Alfonso X El Sabio 28, 23700 Linares, Jaen (Spain)
e-mail: {jccuevas,jcbago,jan,gadeo}@ujaen.es

J.B. Pérez et al. (Eds.): Highlights in PAAMS, AISC 89, pp. 213–221.
springerlink.com

and then are configured in a sleep mode during an interval, which is calculated for each application and traditionally remains constant.

Although considerable research has been devoted to different WSN applications [2], rather less attention has been paid to adapting the **Sleep Mode Intervals (SMIs)** in each work cycle from the evaluation of the surrounding environment. In most cases, the estimated SMI is calculated with energy, coverage or routing restrictions [3]. The objective of this work is to obtain dynamic parameters related to awake-sleep cycle using a multi-agent sensor structure and their propagation among sensor nodes in a WSN in order to improve their lifetime. These dynamic parameters will allow sensor nodes to adapt their work cycles to the current conditions, reducing the number of work cycles and battery consumption, and therefore, increasing their lifetime. On the other hand, sharing the parameters with neighbour sensor nodes will improve the complete network behaviour.

The remainder of the paper is organized as follows. The following section deals with related work. Section 3 shows the sensor multi-agent structure and the system proposed to obtain dynamic parameters that will be propagated among sensors in a WSN. Section 4 presents the system adaptation to an application devoted to sound pressure monitoring. Section 5 shows the experimental results. Finally, conclusions are drawn in Section 6.

2 Related Work

WSNs [4] have become a new important area under study due to their possibilities, like mobility, distributed processing, monitoring and control of dangerous processes. However, WSNs have powerful constraints, mainly when sensor nodes are isolated (limited power source, computational capacity, wireless interference, routing, etc) that the applications for WSNs have to take into account. Therefore, that is one of the reasons of using intelligent systems to manage sensor nodes. Moreover, WSNs represent an ideal scenario to integrate intelligent agents that can accomplish complex applications despite the constraints of the WSNs [5].

Multi-agent theory and its applications have been under study for several years [6] and have become a real solution for a wide variety of complex problems [7]. In this work, the multi-agent system is embedded inside sensor nodes to manage certain decisions of the sensor. That is the case of WISMAP [8], a WSN application management protocol that defines a special multi-agent based framework over WSNs. WISMAP encloses communications, application process, data format, resource hierarchy and agent interaction inside and outside the sensor node. That framework shows that multi-agent systems can perfectly suit into a sensor node to manage WSNs resources efficiently. And one of those important resources is the battery, so sensor nodes usually have to stay most of the time in power saving modes and only in active mode for very short periods of time. Therefore, SMI is one of the parameters under study nowadays in WSNs.

3 Propagation of Agent Performance Parameters

The SMI represents an important aspect in WSNs due to the fact that it permits to prolong the lifetime of sensor nodes. Traditionally, the SMI is calculated for each application remaining constant during its lifetime or, in other cases, is calculated by sensor nodes individually. The main objective of this work is the propagation of dynamic parameters related to awake-sleep cycle among the sensor nodes in a WSN in order to improve their lifetime. Basically, each sensor node calculates its own next cycle SMI, changes to sleep mode for this time, evaluates it when returning to awake mode, and shares its value and evaluation with the rest of sensor nodes by mean of transmitting them via a base station (computer with an access point to the WSN). The shared values of SMIs and their evaluation allow sensor nodes to improve and adjust their values.

The next section presents the multi-agent structure used by sensor nodes that it is an enhancement of WISMAP structure. This structure includes a new subsystem with two new methods, one to obtain the SMI, and the other to evaluate the SMI. Section 4 presents an application of sound pressure monitoring.

3.1 Multi-Agent Structure for WSNs

The sensor node software is based on the multi-agent structure defined by WISMAP. This structure is composed of three agents: management, application control, and communication. The main objectives of the **management agent** are the execution of other agents and the control of the sensor sleep-awake cycle, in which it calculates a new SMI, programs the sensor node in sleep mode, and allows it to return to awake mode. Executed by the management agent, the **application control agent** allows the sensor node to control the execution of different applications (measurement, actuators, FRBS, etc). The **communication control agent** incorporates the application protocol, which allows sensor nodes to communicate with other sensor nodes, neighbouring sensors, and with a base station.

3.2 Differential Decision System

The proposed method to calculate the next SMI, the **Differential System (DS),** is based on the difference between the variable value measured in the present cycle and its values obtained in previous cycles, the region where the values fit and the battery level. The method consists of three steps:

1º) The sensor node measures the value of the object variable and calculates the difference between the new value and the measured in the previous cycle.

2º) After that, the sensor node verifies if the present value belongs to a different region than previous values. Besides, the sensor node may take into account if the present value belongs to critical regions.

3º) Taking into account the difference of values, change of zone, critical regions, battery level, and previous SMIs, the sensor calculates the next SMI.

The parameters to calculate the SMI depend on the application and have to be adapted to its objective. Section 4 presents one application of sound pressure monitoring where a particular SMI is calculated using the DS.

3.3 Proposed Success Conditions

We propose that the evaluation of the SMI can be based on a set of success conditions that are evaluated by each node when it returns to awake mode. Sensor nodes can take into account internal conditions, such as previous SMIs, scheduled tasks and the battery level, and external information, such as variable values, variable tendencies, alarms, and the neighbor or network SMI. On the one hand, critical regions can be defined in the range of input variables. For example, success conditions can include scenarios such as long increments in the value of the variable with small SMIs, short increments with large SMIs, large SMIs with low battery levels, and small SMIs in critical regions. On the other hand, failures conditions can include scenarios such as long increments with large SMIs, short increments with small SMIs, small SMIs with low battery levels, and large SMIs in critical regions. As we have indicated previously, the way on which the next SMI is evaluated depends on the application and has to be adapted to its objective.

4 Sound Pressure Monitoring Application

We have to remark that the main purpose of the solution explained above is to propagate the best SMI obtained among all the sensor nodes to avoid unnecessary awake cycles in every sensor node that is monitoring a particular magnitude in applications where power supply for sensors are not suitable or even impossible. Thus, the applications that are covered by this system have to be studied and modelled separately and must show an inertial behaviour like, for example, the temperature of a room, or the revolutions per minute of an engine.

The magnitude used to test the DS is the sound pressure. This magnitude is used to measure the loudness of sound in one area for a certain period of time. Sound pressure is a typical magnitude in acoustical pollution [9] but with the right tuning can be used to control other systems like working machinery [10]. Engines or moving parts produce similar sound patterns; therefore several ranges of sound pressure can be assigned to different loads of work or failures.

4.1 Differential System Parameters and Calculus

The parameters of the DS can be divided among three categories, related to the difference in measured values, sound pressure level and battery level:

- Maximum SMI (SMI_{max}): if the new SMI> SMI_{max} then SMI= SMI_{max}.
- Minimum SMI (SMI_{min} if the new SMI< SMI_{min} then SMI= SMI_{min}.
- Region hop (*rh*): if the new measure goes into a different region shown in Table 1 then *rh*= abs(current region value - previous region value).

Table 1 Differential system thresholds

Sound pressure	Lower critical	Lower normal	Upper normal	Upper critical	
	45 dB	55 dB	65 dB	75 dB	
Region value	-2	-1	0	1	2

- Steady (*st*): *st* grows one unit if the new measure does not change of region.
- Difference (d): the difference between the previous measure and the new one.
- Previous SMI (ps): this is the previous estimated SMI.
- New SMI (ns): the new SMI that is modified by the three phases of the DS.

Then, with *rh* and *st,* the base increment *i* is calculated as follows:

$$i = i - (rh * 0.1) + (st * 0.05) \tag{1}$$

- Delta (Δ) is the adjustment to *ps* due to the difference between the new measure and the previous one, and it can take three values:
  - $\Delta = +10\%$ if $d < 10$ dB
  - $\Delta = -10\%$ if $10 \text{ dB} \leq d < 20$ dB
  - $\Delta = -20\%$ if $d \geq 20$ dB
- Increment of duty time (*B*) due to battery level (β) is the last modifier applied to *ps* to obtain *ns*. *B* can take the following values:

Table 2 Increment of duty time (*B*) due to battery level (β)

β	β > 75%	50% > β ≤ 75%	25% > β ≤ 50%	10% > β ≤ 25%	β ≤ 10%
B	+ 0%	+ 10%	+ 30%	+ 60%	+ 100%

With all the modifiers the final *ns* value is:

$$ns = (ps * i)\ (1 + \Delta)\ \ (1 + B) \tag{2}$$

4.2 Quality Evaluation

To achieve the main goal of the system described in this paper, the propagation of performance parameters, every sensor calculates its own quality evaluation based on the previous SMI and the new region value of the taken measure. The evaluation gives an integer number *Q*, from -∞ to +∞. The algorithm to calculate Q uses two SMI values as bounds to define what is short (SMI_{short}) or long (SMI_{long}) compared to the possible SMI range. The algorithm of Q evaluation is the following:

- IF rh = 0 AND ps is NOT short THEN Q = Q + 1 ELSE Q = Q - 1
- IF rh=1 AND ps is NOT long THEN Q = Q + 1 ELSE Q = Q - 1
- IF rh=2 AND ps is short THEN Q = Q + 1 ELSE Q = Q - 2

Where:

short: SMI<SMIshort and SMIshort= SMImin + (SMImax - SMImin)/4
long: SMI>SMIlong and SMIlong= SMImin + (SMImax - SMImin)/2

When the level of quality reaches the stability threshold, established empirically in Q = 5, the sensor transmits to the base station the information of the SMI calculated and its evaluation. This message is stored in the base station and compared with other received from sensor nodes in the same area that the sending one. If the evaluation value is higher than the other values, the next time the base station would receive a sensor message, the response message will also carry that evaluation value and the SMI calculated by the previous sensor. Then, the SMI received will be used as the new SMI for the next estimation. In this approach, sensor nodes controlling the same area can collaborate to reach the most appropriate SMI and save processing time when surrounding sensors calculates a better SMI.

5 Results

The systems explained in previous sections have been tested in several scenarios to analyze the error accumulated in the measures when the SMI is not constant and short compared to the inertia of the system. The main purpose of these tests is to demonstrate that the DS can take enough measures to detect the evolution of inertial magnitudes, with less work cycles than Continuous Measurement Systems (CMSs). It is expected that the error introduced by DS will be insignificant and the sensor node lifetime will be prolonged.

5.1 Description of the Experiment

The system has been tested with pseudo-random sound pressure signals and real sound pressure measures obtained from working computers. Those kinds of systems are highly inertial and could model common systems like engines or conveyor belts, where sound should keep a constant pattern along time, only changing when the working duty increases or when there is a failure.

The range of the sound pressure values goes from 40 dBA to 90 dBA. Normal working values are around 60 dBA, so 40 dBA is close to silence, and 90 dBA is the louder sound pressure expected. In order to test the system and to get the sound pressure values, we have designed and implemented a simple analog circuit, equipped with an electret microphone, which has been incorporated to a Sun SPOT sensor.

5.1.1 Simulations

The pseudo-randomly generated signals, that represent inertial systems, have values of sound pressure from 40 dBA to 90 dBA.Every test shows one day of continuous working, with three different measures taken: every second (reference signal), every 20 seconds (CMS), and the measures taken from the DS. Each simulation is calculated with three battery charges, 99.9%, 50% and 10%.

5.1.2 Real System

In order to measure the quality of DS and avoid the effects of different battery consumption among sensor nodes, only one sensor was used to take the real sound pressure samples. The real system used is an opened computer case with one additional fan that can be manipulated to simulate failures. The total processing time of a sensor node with the DS running is around 1800 ms in a Sun SPOT. The normal processing time of a sensor node testing only the instant sound pressure takes around 1450 ms, so the penalty of the DS system can be assumed due to the reduction in the number times to wake up.

5.2 Experimental Results

The results of the experiments show that the DS can follow the evolution of the system with a reasonable loss of precision and with less awake cycles than the CMS. Figure 1 illustrates the evolution of both systems.

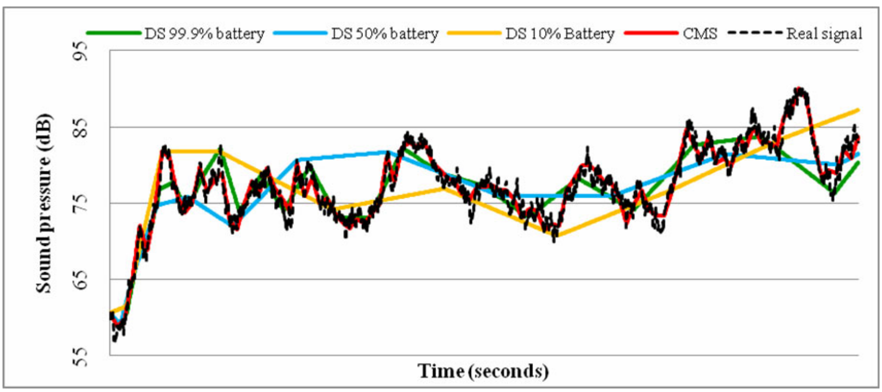

Fig. 1 Interpolated signals with pseudo-random measures for 2000 seconds

To obtain the interpolated signals the CMS woke up the sensor 100 times and the DS only 25 times (with 99.9% of battery charge), that is around a 65 % of power consumption reduction taking into account the total processing time.

5.3 Error Measures

It is clear that with fewer measures the possibility of losing significant values increases. Nevertheless, in WSNs the cost of losing measures can be more acceptable than excessive power consumption but only when the global error introduced by the DS was similar to the CMS error. The first error indicator is the absolute error of the interpolated signals generated through the measurement points of both systems compared to the real signal. The evolution of the absolute error for the

Table 3 Absolute error

System	DS 99.9% battery	DS 50 % battery	DS 10% battery	CMS
Absolute error	2,316%	3,693%	3,996%	1,127%

same period than Figure 1 is shown in Table 3. These experiments have been repeated for more than a hundred of different signals giving always similar values.

5.3.1 Quality Evaluation

As can be observed in Table 4, the best quality is obtained when the battery charge is almost full, and battery regulation does not apply yet (see section 4.2). Nevertheless, the result for 50% battery is good enough to reach 5 at the end of the interval.

Table 4 Evaluation of Q for the same period than Figure 1

	Instant (s)	20	208	407	790	1004	1259	1570	1749	2000
Battery	99.9 %	0	-1	1	4	6	7	9	8	10
	50.0 %	0	-1	1	1	1	2	3	4	5
	10.0 %	0	-3	-2	-3	-4	-5	-6	-6	-5

6 Conclusions and Future Research

This work has presented a method to obtain dynamic parameters related to the awake-sleep cycle in sensor nodes in order to prolong the lifetime of sensor nodes. Besides, a sound pressure monitoring has been developed to test the method. The results demonstrate that simple differential methods, with low computation cost, can save battery in continuous sensing applications with a reasonable loss of precision. On the other hand, further research is necessary in DS to model other real magnitudes and to tune the increment of the delta evaluation system and reduce the adaptation time of sleep mode interval to appropriate values. Future work will focus on the design and development of a FRBS devoted to infer the SMI.

Acknowledgments. This work has been partially supported by the University of Jaen and Caja Rural de Jaen (Spain) (project UJA 08_16_18) and the Ministerio de Ciencia e Innovación (project TEC2009-13619).

References

[1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cyirci, E.: Wireless sensor networks: A survey. Computer Networks 38(4), 393–422 (2002)

[2] Yick, Y., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Computer Networks 52(12, 22), 2292–2330 (2008)

[3] Cohen, R., Kapchits, B.: An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. IEEE/ACM Trans. Netw. 17(2), 570–581 (2009)

[4] Karlsson, B.: Intelligent Sensor Networks - an Agent-Oriented Approach. In: Workshop on Real-World Wireless Sensor Networks (2005)

[5] Cuevas-Martinez, J.C., Gadeo-Martos, M.A., Fernandez-Prieto, J.A., Canada-Bago, J., Yuste-Delgado, A.J.: Wireless Intelligent Sensors Management Application Protocol-WISMAP. Sensors 10(6), 8827–8849 (2010)

[6] Wooldridge, M.: An Introduction to Multiagent Systems. Wiley, New York (2002)

[7] Karl, H., Willig, A.: Protocols and Architectures for Wireless Sensor Networks. John Wiley & Sons, Chichester (2005)

[8] Filipponi, L., Santini, S., Vitaletti, A.: Data Collection in Wireless Sensor Networks for Noise Pollution Monitoring. In: Proc. of the 4th IEEE International Conference on Distributed Computing in Sensor Systems, Santorini Island, Greece, pp. 492–497 (2008)

[9] Tandon, N., Choudhury, A.: A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. Tribology International 32(8), 469–480 (1999)