Power Consumption Reduction in Wireless Sensor Networks through Optimal Synchronization

David Macii, Anton Ageev, Andrey Somov
DISI - Dipartimento di Ingegneria e Scienze dell'Informazione
University of Trento
Via Sommarive 14, 38100, Trento, Italy
E–mail: {macii,ageev,somov}@disi.unitn.it

***Abstract*—Operational lifetime of Wireless Sensor Network (WSN) nodes is crucial in a variety of monitoring applications. As the radio chip is usually the most power–hungry component in small, low–cost WSN devices, battery lifetime can be extended by reducing the duty–cycle of the radio module. In fact, the wireless chip could be switched on just to run the tasks of the considered application, while it could be kept in sleep mode for all the rest of time. Of course, this approach is viable only if the monitoring tasks are scheduled periodically and if the devices are synchronized, namely if they have a common notion of time. Indeed, if nodes were unsynchronized, some of them might wake up when others are still sleeping and some connections could not be established. Since inter–node time synchronization can be maintained within known uncertainty boundaries only by repeatedly adjusting local clocks, synchronization activities can be also scheduled periodically. In this respect, this paper describes an analytical criterion to establish the value of the synchronization period minimizing the average power dissipated by a WSN node. The proposed analysis is validated by means of both simulation and experimental results.**

I. Introduction

As known, battery lifetime represents a major limitation for the deployment of wireless sensor networks (WSNs), especially in outdoor applications [1]. In fact, replacing the depleted batteries of WSN nodes may be excessively expensive or impractical if a WSN consists of large number of devices or if the nodes are difficult to reach. Also, disposable batteries are very polluting and harmful for the environment. In last years, two kinds of complementary strategies have been adopted to tackle this problem, i.e. energy scavenging and power management strategies. The former techniques harvest the energy from the environment, e.g. by means of photovoltaic cells [2], vibration or motion–driven micro–electromechanical devices [3], thermoelectric converters [4], or particular Radio Frequency (RF) receivers [5]. Power management techniques instead rely on hardware and/or software solutions that temporarily disable those components which are unused or underused. In particular, since the radio section is generally the most power–hungry component of a WSN node, reducing the duty cycle of the wireless module as well as the amount of transferred data is essential to prolong battery lifetime. For example, the amount of dissipated power could be greatly decreased through energy–efficient routing schemes [6], or data aggregation techniques [7].

Of course, energy scavenging and power management solutions can be also combined together. For instance, the adaptive duty cycle algorithm described in [8] can utilize up to 58% more ambient energy compared to systems which do not rely on harvesting–aware power management. In applications requiring just periodic monitoring services, major power savings could be obtained if the WSN nodes were perfectly synchronized. In this case, all nodes could remain mostly in sleep mode and they could wake up only for the time that is strictly necessary to run the scheduled monitoring tasks. Unfortunately, the time values measured by different nodes tend to drift away from one another due to the limited frequency stability of the local oscillators. As a consequence, the time offsets of various nodes can be kept within given tolerance boundaries only by running repeatedly some synchronization procedure, such as those described in [9], [10] or [11]. Obviously, frequently repeated synchronizations lead to better timing accuracy. Accordingly, the probability that some nodes are still sleeping while others are already running their own instance of the monitoring task is small. However, the energy dissipation due to frequent synchronization activities may be considerably large. On the other hand, if synchronization events are rare, the related activities are not particularly expensive in terms of energy. Nevertheless, the time intervals in which nodes are active must be longer than expected in order to allocate the worst–case time differences between node timers. Of course, this extension of the radio module activity in turn increases the amount of power consumption.

Starting from this standpoint, in Section II, after a formal description of the problem mentioned above, the expression of the synchronization period minimizing the average power dissipated by a WSN node is derived analytically. In Section III some simulations validating the proposed analysis are shown. Finally, in Section IV some experimental results based on the synchronization protocol described in [11] are reported.

II. Problem Formulation and Model Description

Let us consider a WSN consisting of N identical nodes that are mostly kept in sleep mode and wake up occasionally just to run two independent tasks, i.e.:

- One *monitoring task* of duration d_m and period $T_m >> d_m$;

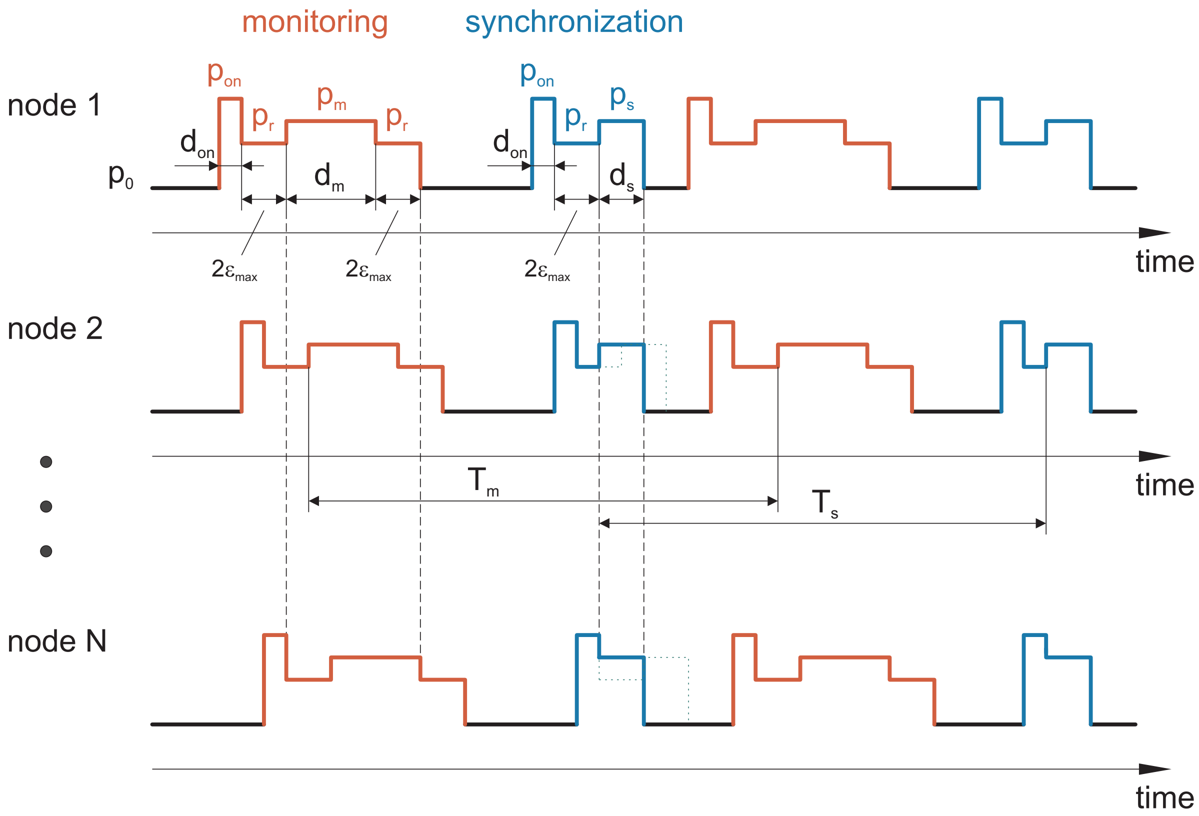

Fig. 1. Qualitative power consumption patterns during the execution of monitoring and synchronization tasks on various nodes of a WSN.

- One *synchronization task* of duration d_s and period $T_s >> d_s$, which is able to keep node time offsets in the range $[-\varepsilon_{max}, \varepsilon_{max}]$.

Of course $\frac{d_m}{T_m} + \frac{d_s}{T_s} < 1$, because the overall utilization of a single node must be strictly lower than 100%. Also, switching from sleep mode to active state takes some time d_{on}, during which the amount of current drawn by the device may be quite large [12], as shown in Fig. 1. Consider that in the proposed model both d_m and d_s include not only the time required to run the activities of a single node of the network, but also the time spent in waiting possible messages and packets coming from other nodes. Moreover, even if the actual duration of each task may change randomly as a function of time (e.g., due to variable traffic conditions), we will assume that both d_m and d_s correspond to the respective maximum (i.e., worst–case) values. In fact, this choice does not alter the basic reasoning underlying the proposed model and it is quite realistic because all nodes usually may be programmed to wait data for a preset time–out value.

In general, a major difference exists between the timing behavior of monitoring and synchronization tasks. The monitoring task can be usually regarded as composed by unrelated activities. This means that a node collects, processes and sends its data to other WSN devices as soon as data are ready, independently of what other devices are doing. In this respect, some lack of synchronization is beneficial to avoid channel contention and packet collisions whenever a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol is used. However, given that the uncertainty associated with the chosen synchronization procedure lies in the interval $[-\varepsilon_{max}, \varepsilon_{max}]$, in the worst–case the execution of an instance of the monitoring task on the node with the slowest clock rate may begin $2\epsilon_{max}$ seconds after the same instance running on the device with the fastest clock rate. Since in low–power mode the radio module of a generic node is disabled, it could happen that the data collected by the "fastest" node are sent to the "slowest" one when it is still sleeping. Similarly, the "slowest" device might attempt to transfer data to the "fastest" node after this one has already entered into sleep mode. This situation is clearly shown in the leftmost part of Fig. 1. Therefore, in order to avoid connection problems, the time interval during which the radio chip must be active should not just coincide with d_m, but it should be rather expanded by $2\varepsilon_{max}$ both before and after the execution of each instance of the monitoring task. This means that the overall duration of the time interval in which a generic node is ready both to run its own monitoring task and to receive data is $d_m + 4\epsilon_{max}$.

The scenario is quite different in the case of the synchronization task. In fact, any synchronization protocol relies on a set of coordinated activities which are usually triggered by the node starting the whole procedure. In the following, for the sake of simplicity, this node will be referred to as *synchronization master* (SM). When the SM broadcasts a synchronization packet, all nodes receiving this packet start executing the synchronization algorithm approximately at the same time. Although the time of execution of the synchronization procedure may differ slightly from node to node (e.g., due to possible differences among the activities performed by the various nodes) we can assume that the instances of the same synchronization task are simultaneous. Of course, a necessary condition to synchronize WSN nodes properly is that when the SM starts the procedure, the other devices must be already active. Therefore, for the same reasons explained above, all nodes should switch on $2\varepsilon_{max}$ seconds before the scheduled synchronization time. However, in this case prolonging the node activity by $2\varepsilon_{max}$ after the end of the synchronization procedure is unnecessary. Indeed, all activities roughly finish simultaneously and the residual time offsets immediately after synchronization can be assumed to be negligible. As a consequence, the duration of the interval during which the SM is active is exactly $d_s + 2\varepsilon_{max}$. This situation is shown in the central part of Fig. 1. Observe that the duration of the "active" interval of any WSN node different from SM is $d_s+\boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ represents the delay between the time when the node $i \neq SM$ switches on and the moment when the synchronization procedure actually begins. In general, $\boldsymbol{\varepsilon}$ is a random variable because it depends on the time offset between the node i and the SM. If the synchronization procedure is started by the "fastest" device, then $\boldsymbol{\varepsilon} \in [0, 2\varepsilon_{max}]$. Conversely, if the synchronization procedure is triggered by the "slowest" node, then $\boldsymbol{\varepsilon} \in [2\varepsilon_{max}, 4\varepsilon_{max}]$. In all other cases, (i.e. when the elected SM is any node with intermediate clock speed) the values of $\boldsymbol{\varepsilon}$ lie in $[\Delta, \Delta+2\varepsilon_{max}]$, where Δ represents the offset between the time of the SM and the time measured by the node clocked by the "slowest" oscillator. Of course, the distribution of $\boldsymbol{\varepsilon}$ depends on the probability density function of the timing errors as well as on the policy chosen to elect the SM. If we assume that the time offsets of the various nodes are uniformly distributed in the range $[-\varepsilon_{max}, \varepsilon_{max}]$ and that all nodes have the same probability of becoming the SM, it can be shown that $E[\boldsymbol{\varepsilon}] = 2\varepsilon_{max}$, where $E[\cdot]$ is the operator returning the expected value of its argument. Given that we are mostly interested in analyzing the average power dissipated by a generic node in different conditions regardless of its role in the network, in the following the duration of the "active" interval associated with the synchronization task will be set equal to the corresponding mean value, i.e. $d_s + 2\varepsilon_{max}$.

Starting from all the considerations explained above, if we refer to:

- p_0 as the average power consumption in sleep mode;
- p_{on} as the average power spent to switch on the radio module;
- p_m as the average power dissipated for running the monitoring task and for exchanging data with other devices;
- p_r as the average power consumed when a node is ready to receive messages from other devices;
- p_s as the average power associated with the execution of an instance of the synchronization task;

the overall average power dissipated by a WSN node over a time interval T equal to the least common multiple (LCM) of periods T_m and T_s results by definition from:

$$\overline{P} = \overline{P}_m + \overline{P}_s + \overline{P}_0 \tag{1}$$

where:

$$\overline{P}_m = \frac{d_m p_m + 4\epsilon_{max} p_r + d_{on} p_{on}}{T_m} \tag{2}$$

is the average power contribution due to the monitoring task,

$$\overline{P}_s = \frac{d_s p_s + 2\epsilon_{max} p_r + d_{on} p_{on}}{T_s} \tag{3}$$

is the average power contribution due to the periodic synchronization task,

$$\overline{P}_0 = p_0 \left(1 - \frac{d_{on} + d_m + 4\epsilon_{max}}{T_m} - \frac{d_{on} + d_s + 2\epsilon_{max}}{T_s}\right) \tag{4}$$

is the average power consumption when the node is in sleep mode. Notice that in (1) the effect of overlapping tasks is neglected. This is due to two reasons. First, the overlapping probability is generally quite small. Indeed, if the monitoring and synchronization tasks are assumed to be independent, the overlapping probability is simply equal to $\frac{d_m}{T_m}\frac{d_s}{T_s}$, and this quantity is usually much smaller than 1 if $d_m << T_m$ and $d_s << T_s$. Second, even when two instances of the monitoring and synchronization tasks overlap, we have just a small decrease in energy dissipation. Such a difference is due to the fact that the radio module switches on only once, rather than twice. Nonetheless, the time to run both task is still roughly equal to $d_m + d_s$, regardless of the adopted scheduling policy. Thus, the energy saving is just $d_{on} p_{on} + 2\varepsilon_{max} p_r$, which is negligible compared to the energy dissipated to run both tasks.

Expression (1) is significant because it can be used to estimate the synchronization period minimizing the average power dissipated by a WSN node. In fact, it is known that the limited frequency stability of local oscillators is the main uncertainty contribution affecting synchronization uncertainty in WSNs [13]. Moreover, the law describing how the time values measured by different nodes drift away from one another is approximately a linear function of time. Thus, if $|\nu_{max}|$ is the absolute value of the relative frequency offset of the least accurate local oscillator, the maximum time error just before performing a new synchronization is approximately $\epsilon_{max} \approx |\nu_{max}| \cdot T_s$. Usually, the value of ν_{max} for low–cost crystal oscillators lies within ±100 ppm. Thus, by replacing the expression of ϵ_{max} into (1), it follows that the derivative of (1) with respect to T_s is equal to zero for

$$T_s^* = \sqrt{\frac{T_m[d_s(p_s - p_0) + d_{on}(p_{on} - p_0)]}{4|\nu_{max}|(p_r - p_0)}} \tag{5}$$

which returns the optimal synchronization period. In conclusion, if the basic parameters of (1) are known, (5) provides an interesting criterion for energy–aware synchronization. Note that if $|\nu_{max}|$ tends to 0 (e.g., because the clock frequency skews are partially compensated by the adopted synchronization algorithm), T_s^* grows towards infinity. This is reasonable, because if $|\nu_{max}|$ is small, running frequently repeated synchronizations is unnecessary.

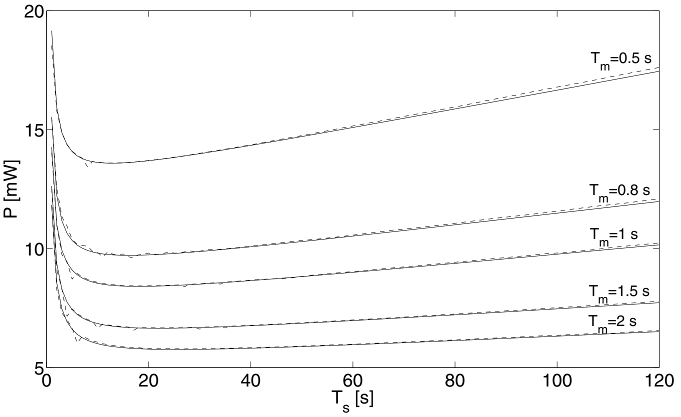

Fig. 2. Average dissipated power as a function of the synchronization interval T_s for different periods of the monitoring task (i.e., $T_m = 0.5, 0.8, 1, 1.5, 2$ s). The solid lines result from (1), whereas dashed lines are obtained through simulations. In all cases the value of the worst–case frequency offset is $\nu_{max} = 100$ ppm.

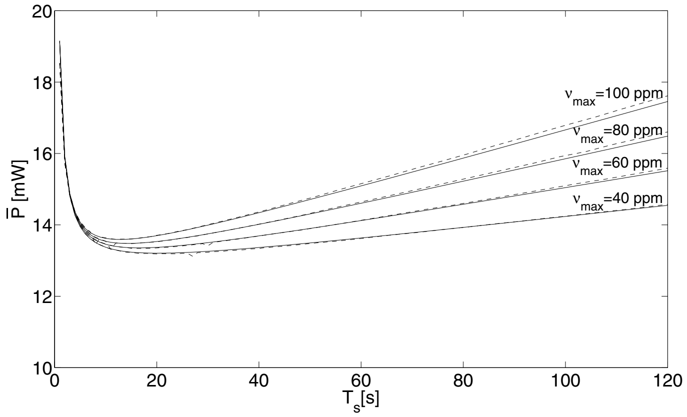

Fig. 3. Average dissipated power as a function of the synchronization interval T_s for different worst–case relative frequency offsets (i.e., $\nu_{max} = 40, 80, 60, 100$ ppm). The solid lines result from (1), whereas the dashed lines are obtained through simulations. In all cases the period of the monitoring task is $T_m = 0.5$ s.

III. SIMULATION RESULTS

In order to better understand the operative implications of (5), in this Section the results of some Matlab™ simulations are reported. The application developed for simulation purposes is able to generate a power waveform in accordance with the model described in Section II. All task scheduling operations (including early wake–up and late switch–off times) rely on a simulated digital timer running nominally at 32768 Hz. As a consequence, the output waveforms can be generated with 1–tick resolution. In order to make simulations closer to reality, the model of the local clock offers the possibility to adjust both its systematic relative frequency offset and the amount of random jitter. Accordingly, several simulations were repeated with different values of T_s, T_m and ν_{max} during variable observation intervals comprising multiple instances of both monitoring and synchronization tasks. The common values of the parameters used in all simulations are the following: $d_{on} = 100$ μs, $d_m = 92$ ms, $d_s = 130$ ms, $p_{on} = 57.4$ mW, $p_r = p_m = 56$ mW, $p_0 = 2.8$ mW, $p_s = 53$ mW. Such values are in reasonable accordance with the execution times and the power consumption contributions of WSN nodes working in different conditions. In Figs. 2 and 3 the power consumption values resulting from simulations (dashed lines) are compared with the corresponding theoretical curves (solid lines) obtained from (1). In particular, in Fig. 2 the average power curves are plotted as a function of T_s after setting $\nu_{max} = 100$ ppm and for five different values of T_m (i.e., $T_m = 0.5, 0.8, 1, 1.5, 2$ seconds). Dually, in Fig. 3 similar curves are plotted for a fixed T_m value (i.e., $T_m = 0.5$ s), while changing ν_{max} (i.e., $\nu_{max} = 40, 80, 60, 100$ ppm). Observe that, even if the simulations include the possibility of having overlapping tasks, the difference between each pair of simulated and theoretical curves is negligible. In the first set of simulations, the optimal synchronization period T_s^* ranges from 12.4 s when $T_m = 0.5$ s to 24.8 s when $T_m = 2$ s. Indeed, it doubles when T_m increases by 4 times, as expected. In the second set of simulations T_s^* grows from 12.4 s for $\nu_{max} = 100$ ppm up to 19.6 s for $\nu_{max} = 40$ ppm. The results shown in Figs. 2 and 3 confirm that in general the optimal power consumption is quite sensitive to ν_{max}. If ν_{max} becomes very small, synchronizations are rare and the total power consumption is mostly due the monitoring task, as it should be in the ideal case. Notice also that if the parameters of (5) are affected by some significant uncertainty it is preferable to overestimate T_s^*, because the amount of dissipated power tends to increase much more smoothly for $T_s > T_s^*$.

IV. EXPERIMENTAL RESULTS

The theoretical analysis described in Section II was validated experimentally using a small WSN consisting of five Crossbow TelosB nodes. Each device was programmed to run two concurrent applications, i.e. a simple humidity monitoring task and the synchronization procedure described in [11], suitably modified to switch off the radio chip when the node is supposed to be idle. A further device connected to a PC was used to collect some significant data from all WSN nodes (e.g., the synchronization intervals). The synchronization algorithm reported in [11] is adaptive. This means that the time interval between two subsequent synchronizations changes at run–time depending on the wanted target accuracy. In particular, for a given value of ε_{max}, the synchronization interval is expanded [i.e., multiplied by $(1 + a)$, with $a \geq 0$] if the estimated probability of having in–tolerance nodes is larger than a preset threshold Q. Conversely, the interval between two synchronizations is shortened [i.e., multiplied by $(1 - b)$, with $0 \leq b < 1$] when the percentage of nodes with in–tolerance clocks is smaller than Q. As a consequence, the synchronization interval values tend to exhibit a lognormal distribution [11]. The mean of this distribution can be regarded as an estimator of the synchronization period T_{s_Q} for which

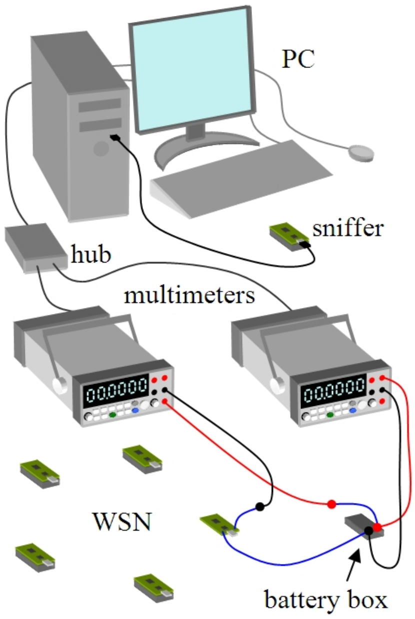

Fig. 4. Experimental setup. Five Xbow TelosB nodes are programmed to run both a periodic humidity monitoring task and the synchronization protocol described in [11]. The applied voltage and current drain of a generic node are measured simultaneously by means of two DMMs Agilent 34411A which are connected to a PC collecting both measurement results and sensor information by means of a further node (i.e. a sniffer).

the probability of having nodes with time offsets in the range $[-\varepsilon_{max}, \varepsilon_{max}]$ oscillates around Q.

Starting from these considerations, the experimental activities were carried out in two phases. In the first one, after setting $\varepsilon_{max} = 1$ ms, $Q = 100$ %, $a = 0.02$ and $b = 0.33$, the synchronization procedure was repeatedly executed for several minutes in order to collect a significant amount of interval values. Accordingly, the mean of the lognormal distribution was computed and the maximum relative frequency offset ν_{max} of the considered set of nodes was estimated from $\varepsilon_{max}/T_{s_Q}$. In the case considered ν_{max} is smaller than expected, i.e. 22 ppm.

In the second phase, at first the adaptive behavior of the synchronization algorithm was disabled by setting $a = b = 0$. In this way, the synchronization intervals remain constantly equal to the initial value. Then, all nodes were programmed to collect a humidity sample every $T_m = 0.6$ s and to run the synchronization procedure for different periods in the range $[1, 120]$ s. The voltage supply of a single WSN node as well as its current drain were measured using two digital multimeters (DMMs) Agilent 34411A (one used as a voltmeter and the other as an amperometer) connected to the PC, as shown in Fig. 4. The integration time of both DMMs was set to 100 power line cycles (i.e., 2 s) in order to exploit the best instrument resolution and to maximize normal mode noise rejection. The duration of each experiment was about $10T_s$ seconds. Since the battery voltage is approximately constant, each average power consumption value can be estimated as the product of the average current drain and the average supply voltage, i.e. $\overline{P} = \bar{I}\bar{V}$. All measurement results are plotted in Fig. 5 (solid line). Notice that the shape of the curve is in good agreement with the simulation results shown in Fig. 2, although the values are smaller. The standard uncertainty associated with $\overline{P}$ was calculated using a procedure similar to that described in [14], i.e.:

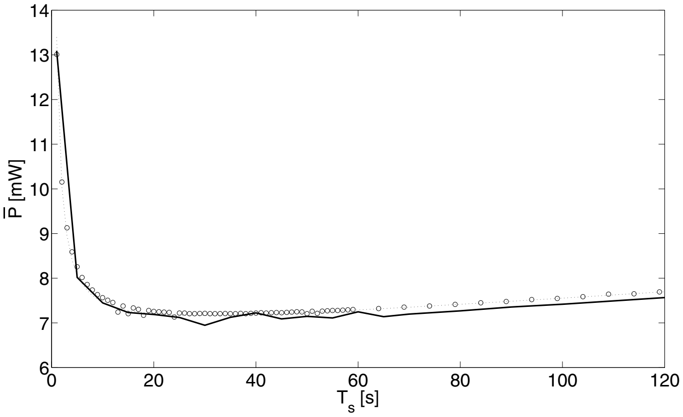

Fig. 5. Average power consumption as a function of the synchronization interval T_s for $\nu_{max} = 22$ ppm (namely the estimated maximum relative frequency skew in the considered WSN). In the picture, the solid line refers to the measurement values collected using the setup shown in Fig. 4, the dashed line results from (1), and the bubble points correspond to simulations. The optimal synchronization period is $T_s^* = 29.5$ s.

$$u(\overline{P}) = \sqrt{\bar{I}^2 u^2(\bar{V}) + \bar{V}^2 u^2(\bar{I})} \tag{6}$$

where $u^2(\bar{V})$ and $u^2(\bar{I})$ are the combined standard uncertainties related to current and voltage measurements, respectively. Both $u^2(\bar{V})$ and $u^2(\bar{I})$ include random and systematic contributions, as stated in [14]. The former ones can be estimated using the typical *type A* evaluation procedure and are particularly significant in the case of short measurement times (e.g., for small value of T_s). Conversely, the systematic contributions are independent of measurement duration and can be estimated from instrument specifications by following a *type B* evaluation procedure [15]. In our experiments, the maximum standard uncertainty value resulting from (6) is 0.3 mW, which is adequate because it is about one order of magnitude smaller than the measured values.

In order to perform a final check about the correctness of the proposed model we also compared the collected measurement results with the theoretical curve given by (5) and with the corresponding simulations. The values of the parameters in (1) were estimated with the help of a digital oscilloscope Agilent DSO7032A. In particular, such values are $d_{on} \approx 2.5$ ms, $d_m \approx 76$ ms, $d_s \approx 126$ ms, $p_{on} \approx 7.5$ mW, $p_r \approx p_m \approx 52$ mW, $p_0 \approx 2.8$ μW and $p_s \approx 53$ mW. Both the theoretical curve (dotted line) and the simulation results (bubble points) shown in Fig. 5 are in perfect accordance with the experiments, thus confirming the validity of the model.

V. Conclusion

When a WSN is used in periodic monitoring applications characterized by a small duty cycle, a quite obvious strategy for saving power is to switch off the radio module of all nodes during idle time intervals. However, this approach may cause major connection problems if nodes are not properly synchronized. Indeed, some nodes might try to communicate when others are still sleeping. In this work we proved that the overall average power dissipated by a node running a periodic monitoring task and a periodic synchronization procedure, can be suitably minimized while meeting the timing accuracy requirements needed for reliable inter–node communications. An in–depth analysis of the problem mentioned above has led to the definition of a simple analytical expression [i.e., equation (5)] returning the value of the optimal synchronization period. Several simulation and experimental results confirm the correctness of the proposed model. Even if (5) depends on several parameters which are strictly related to the chosen hardware platform, most of them are approximately constant and can be measured quite easily or derived from the specifications of the WSN nodes. A critical parameter that somehow limits the practical applicability of (5) is the maximum relative frequency offset ν_{max}. In fact, not only (5) is quite sensitive to ν_{max}, but the frequency skew may differ considerably from node to node. Furthermore, it may also change in time due to variable operating conditions (e.g., temperature fluctuations). In order to address this issue, next–generation synchronization algorithms should be able to estimate the value of ν_{max} at run–time and to adjust the duration of the optimal synchronization period accordingly.

Acknowledgment

The results presented in this paper are part of the FP–7 EU project *Control of Heterogeneous Automation Systems* (CHAT), EC contract IST-2008-224428. Authors would like to thank the local coordinator of the project, Prof. Luigi Palopoli, for supporting this work.

References

[1] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[2] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of 4th International Symposium on Information Processing in Sensor Networks*, 2005, pp. 457–462.

[3] R. Torah, M. Tudor, K. Patel, I. Garcia, and S. Beeby, "Autonomous low power microsystem powered by vibration energy harvesting," in *Proceedings of 6th Annual IEEE Conference on Sensors*, Atlanta, Georgia, USA, Oct. 28-31, 2007.

[4] H. Lhermet, C. Condemine, M. Plissonnier, R. Salot, P. Audebert, and M. Rosset, "Efficient power management circuit: From thermal energy harvesting to above-ic microbattery energy storage," *IEEE Journal on Solid-State Circuits*, vol. 43, no. 1, pp. 246–255, Jan. 2008.

[5] T. Paing, J. Shin, R. Zane, and Z. Popovic, "Resistor emulation approach to low-power rf energy harvesting," *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 1494–1501, May 2008.

[6] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *SECON'06: Sensor and Ad Hoc Communications and Networks*, 2006, pp. 286–295.

[7] S. Croce, F. Marcelloni, and M. Vecchio, "Reducing power consumption in wireless sensor networks using a novel approach to data aggregation," *The Computer Journal*, vol. 51, pp. 227–239, 2007.

[8] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan, "Adaptive duty cycling for energy harvesting systems," in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design (ISPLED'06)*, Tegernsee, Germany, 2006, pp. 180–185.

[9] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, vol. 36, 2002, pp. 147–163.

[10] C. M. Chao and Y. C. Chang, "A power-efficient timing synchronization protocol for wireless sensor networks," *Journal of Information Science and Engineering*, vol. 23, no. 4, pp. 985–997, 2007.

[11] A. Ageev, D.Macii, and A.Flammini, "Towards an adaptive synchronization policy for wireles sensor networks," in *Proceedings of the International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, Ann Arbor, Michigan, USA, Sep. 2008, pp. 115–120.

[12] A. Wang, S.-H. Cho, C. Sodini, and A. Chandrakasan, "Energy-efficient modulation and mac for asymmetric microsensor systems," in *International Symposium on Low-Power Electronics and Design (ISPLED)*, Huntington Beach, CA, USA, Aug. 2001, pp. 106–111.

[13] A. Ageev, D. Macii, and D. Petri, "Synchronization uncertainty contributions in wireless sensor networks," in *International Instrumentation and Measurement Technology Conference (I2MTC)*, Victoria, BC, Canada, May 2008, pp. 1986–1991.

[14] D. Macii and D. Petri, "An effective power consumption measurement procedure for bluetooth wireless modules," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 4, pp. 1355–1364, 2007.

[15] *ISO/IEC Guide 98:1995, Guide to the Expression of Uncertainty in Measurement*, Geneva, Switzerland, 1995.