Adaptive Duty-Cycling Algorithms for Efficient Energy Harvesting in Wireless Sensor Networks

Dan Ștefan Tudose
Computer Science Department
University Politehnica Bucharest
Bucharest, Romania
dan.tudose@cs.pub.ro

Alexandru Marin, Marius Geantă
Computer Science Department
University Politehnica Bucharest
Bucharest, Romania
marius.geanta@cti.pub.ro, alex.marin@cti.pub.ro

Abstract—Wireless sensor nodes which harvest energy from the environment have become an alternative to battery powered nodes. Requirements for efficient use of the extracted energy led to development of algorithms that manage the node functions depending on the amount of collected energy. This article summarizes findings in researching power management and duty-cycling algorithms for solar energy harvesting wireless sensor nodes. A novel solution of adaptively setting the duty-cycle of a wireless sensor node in order to maximize its monitoring lifetime is introduced. The developed algorithms are particularly suited to energy harvesting wireless sensor networks situated in locations where energy is scarce or where harvested power exhibits ample diurnal or seasonal variation. The results described in this article shows that the proposed wireless sensor network architecture can represent a viable solution for monitoring indoor environments characterized by low illumination. The setup was tested and validated under various lighting conditions, using the adaptive techniques described in the paper.

Keywords— wireless sensor networks; energy harvesting; power management; duty-cycling; supercapacitors; solar energy

I. Introduction

Wireless sensor networks (WSN) are traditionally powered using batteries. Although this method is acceptable for some applications, it is difficult to ensure maintenance in scenarios where nodes are placed in remote locations and the effort to replace their batteries becomes considerable. A better way of allowing nodes to function in diverse places is to extract energy from the environment. There are different sources of energy such as solar, wind, mechanical, electrostatic, that can be harvested through technologies such as photovoltaic cells, wind turbines or piezoelectric systems. The main advantage of harvesting energy is the extended functional time of the node. It can virtually operate indefinitely, the user benefiting of increased reliability and simultaneously lower costs due to eliminating the need of battery replacement.

However, there are some issues to be considered when approaching energy harvesting solutions in WSN. First, the availability of the energy to be harvested has a great impact on the design. The minimum level is usually zero, when no energy can be collected. Because of this, a method to efficiently store the energy when it can actually be collected is needed. The choice is between: batteries, capacitors or super capacitors. Secondly, the unpredictability of energy levels has to be accounted for in the power management scheme of the sensor node. Although for solar there is a known cycle of day and night, the amount of energy can differ from one day to another, sunny and cloudy days, and this can affect the long term node operation. Different strategies were proposed to handle the varying input energy levels, including prediction models such as the one proposed by Kansal et al. [1], or adaptive control systems [2].

This article focuses on harvesting solar energy using small footprint photovoltaic panels and super capacitors as storage devices. The contributions include design and implementation of adaptive duty-cycling algorithms for WSN power management, which are described in section 3. The algorithms' performance is compared and evaluated using Sparrow v3 nodes powered by super capacitors of different capacities, inside indoor environments with various illumination conditions. The objective of this research is to demonstrate that the hardware and firmware setup is a working solution for continuously monitoring indoor environment conditions with energy harvesting nodes.

II. Related Work

In recent years several work pursued the design of efficient algorithms in energy harvesting sensor nodes. These schemes followed three major objectives: first, manage the available energy level in a sustainable manner such that the nodes never run out of resources; second, efficiently use the harvested energy in order to extract maximum utility from the node functions; third, adapt to the unpredictable nature of the harvesting environment and modify the energy utilization according to perceived changes.

The article written by Kansal et al. is one of the first works to study power management in energy harvesting WSN. It introduced Energy Neutral Operation (ENO) for energy harvesting nodes. This states that the energy consumption will be kept in balance with the gained energy over a defined time period, such that the node operates continuously.

Another contribution of the article is the proposed power management algorithm, based on duty-cycling. Duty-cycle computation alternates sleep times, during which the node is in low power state but still collecting energy from the environment, with operating times where the node executes its designed functions. It is based on the premise that the utility function is increasing with the time the node stays active and consumes energy, and the largest power consumer is the radio transceiver.

The proposed algorithm is splitting time in N equal slots per day. Inputs are the current harvested energy levels along with predicted harvesting values. Prediction on the amount of solar energy to be harvested is done using an Exponentially Weighted Moving-Average (EWMA) filter applied on the previously collected data. The duty-cycle is computed to compensate for the difference between the actual values and the ones predicted by the model.

Vigorito et al. proposes a model-free approach to the duty-cycling problem by exploiting adaptive control theory techniques. A control algorithm is applied to the dynamic system, the harvesting node, with the objective of keeping the voltage within an interval centered on a target value. The equation to be solved is minimizing the quadratic cost function $|output - target|^2$ while keeping the ENO valid. After calculating the duty-cycle, a smoothing function is applied on the obtained value in order to lower the variance. It is considered that lowering the variance can be a requirement of certain WSN applications.

The main contribution of this work consists in eliminating the need for a prediction model, like the one used by Kansal et al. and offering an algorithm with very low computational requirements and very high generality, not dependent on certain energy sources and collected data.

In the work published by Cammarano et al. [3], a new prediction model named Pro-Energy is proposed, claiming an improvement of 60% over previous models such as EWMA or WCMA [4]. The disadvantage of EWMA was that the weight of the previous day data in estimating the energy intake for the current day was too large, leading to prediction errors when sunny and cloudy days were alternating.

WCMA is addressing the issues in EWMA, however it is predicting only 1 time slot ahead. The Pro-Energy model’s idea is to derive predictions for a longer series of time slots based on stored profiles of harvested energy values. An array of gained energy values in the past N time slots is maintained and compared for similarity with one of the other D stored profiles of N data time slots. The comparison is done using the mean absolute error (MAE) over a span of K last values. The chosen prediction is linear combination of the value from the profile that gives the minimum error with the value obtained in the last time slot. For better accuracy, profiles are combined in a weighted profile. Pro-Energy achieves predictions 75% more accurate than EWMA when the forecast period is short (30 minutes) and is closing to 50% improvements on medium-term like two to three hours.

Hsu et al. [5], introduces a system model view of an energy harvesting node and a theoretical framework to calculate the optimal power management in such a node, based also on ENO. The implementation is developed starting from a low complexity solution that uses EWMA and is later adapted to address the shortcomings mentioned earlier. The evaluation consists in a comparison of the three methods: optimal, simple and adaptive, concluding that the adaptive is close to the optimal calculated values of solar energy utilization.

Next the solution for duty-cycling wireless sensor nodes is presented. Two novel algorithms that calculate the sleep times based on current and previous energy levels are introduced.

III. Adaptive Duty-Cycling Algorithms

The requirements considered when establishing the design were:

- Reliability: respect the ENO principle and do not spend more energy than what is collected, leading to perpetual operation of the harvesting node.
- Generality: the algorithm can be applied in any harvesting environment regardless of the energy source.
- Efficiency: execution times of the algorithms are constant, having a minimal impact on the node computation resources (processor and memory) as well as on the energy consumption.

The design of the adaptive algorithms is also founded on the fact that the largest energy consumer source in the node is the radio transceiver during data transmission. As a consequence, a decrease in the transmission rate during a time period leads to less energy being spent. This way energy consumption varies with the time the node stays in low power sleep mode where it does not use the transceiver.

The transmission rate is increased or decreased by modifying the sleep periods based on the amount of energy harvested by the node. The main approach is based on a greedy technique: increasing the transmission rate only when an increase in voltage is detected, meaning energy was collected and can be utilized. Otherwise, the energy consumption stays the same, or decreases when voltage decrease is detected. Therefore, it can be stated that in a given time period T consumed energy is less than the energy harvested from the environment, respecting the ENO principle. This condition is verified in the evaluation for a time period T of one day.

In the experimental setup harvested solar energy using photovoltaic panels is employed on nodes that monitor temperature, light and humidity. A simple topology consisting of the target node that directly reports the collected values to the base station is employed. The inputs to the algorithm are: V, the current voltage, PreviousV, voltage from anterior reading, VMAX, maximum operating voltage, VMIN, minimum voltage under which the node will shut down its function.

For the first algorithm, the time period of one day is split in N minutes time slots and read the current voltage at the end of each time slot. This way it can be easily determined if the system lost or gained energy during the last slot, with the current sleep rate. The output S is the period under which the node will function in low power mode, which is referred to as sleep time. After this period, the node wakes up, executes its function and transmits data to the gateway.

As a result, the node will collect data at a rate of S during the time slot, the rest of the time remaining in sleep and harvesting energy. At the end of each time slot the node executes the duty-cycling algorithm to determine the sleep time S for the next slot. Fig. 1 shows the principle through two consecutive time slots with different sleep time values.

What should be underlined is that, although the voltage may vary many times in a time slot, in any direction, the decision is not taken until the time slot finishes. The second algorithm will use a different approach.

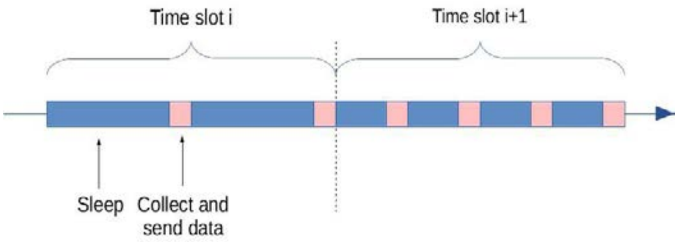

Fig. 1. Duty-cycling principle

The algorithm is based on three threshold values for the voltage level: first TMAX, voltage threshold over which the sleep time at the minimum SMIN is set, second TMIN, voltage threshold under which the sleep time at the maximum SMAX is set and third CRITICAL = VMIN + E, the threshold under which no data transmission through the radio transceiver is done.

Between TMIN and TMAX the algorithm adapts the sleep period proportional with the difference ΔV between current and previous voltage level. ΔV represents the amount of energy gain if positive, or loss if negative. The following sleep rate adaptation is applied:

$$S_{t+1} = S_t - \Delta V * INC \qquad (1)$$

,where INC is the increase value for the sleep period. This value adapts according to ΔS, the difference between current and previous sleep rate. When energy is gained using a sleep rate lower than the previous, INC is also increased by β. This leads to a larger decrease of the sleep rate and increases the energy consumption.

Similarly when energy is lost and the sleep rate was already increased, INC is increased by a constant amount β. If consecutive time slots have voltage increase or decrease together with sleep decrease/increase the sleep rate will grow or shorten quicker than if the degradation is kept constant. The algorithm is defined in pseudo-code below.

Algorithm 1 Adaptive Algorithm 1

```
procedure SleepAllocation
    S ← initial sleep time
    INC ← initial sleep increase factor
    for each time slot do
        ΔV ← V − PreviousV
        ΔS ← S − PreviousS
        if ΔV > 0 and ΔS < 0 or ΔV < 0 and ΔS > 0 then
            INC ← INC ∗ β
        else
            INC ← initial sleep increase factor
        S ← S − ΔV ∗ INC
        if V < CRITICAL then
            Disable sending data
        if V <= TMIN then
            S ← SMAX
        if V >= TMAX then
            S ← SMIN
    end for
end procedure
```

In the second algorithm, the day period is not split in multiple time slots and a decision is taken whenever a slot is finished. The sleep update procedure is called every time the node is waken up and is dependent on the value of the current sleep. TMIN, TMAX and CRITICAL thresholds are still used, but the manner in which the sleep period converges towards SMIN or SMAX is not as steep as in the previous algorithm.

Furthermore, the incremental approach of sleep updating is replaced with a new, exponential one. The sleep rate is adapted in the manner described below in (2), with sleep increase or decrease being proportional with the current sleep time value:

$$S_{t+1} = S_t - \Delta * S / \alpha \qquad (2)$$

In the above equation, Δ represents the direction of the voltage update: positive, i.e. 1, if the voltage increases compared to the previous reading, and negative, i.e. -1, if the voltage decreases. Furthermore, α is a configurable parameter, determining how much the sleep period is modified compared to its current value.

Another difference between the first and the second algorithms is that the sleep adaptation function is not called only if the voltage value is between TMIN and TMAX. The sleep period continues to be adapted even when the voltage is out of [TMIN, TMAX] range, until the sleep value reaches one of the SMIN or SMAX values. As already stated, when the voltage goes below TMIN or above TMAX, the sleep period does not get updated immediately to SMIN or SMAX. Rather than doing this, the sleep rate adaptation slowly drives the sleep period in the direction of those values.

If the voltage drops below TMIN and the sleep period is still under SMAX, the sleep rate adaptation function is further called each time a voltage decrease is detected, having as a result an increase of the sleep period. The new sleep period is compared with SMAX, taking its value when it exceeds it. However, if a voltage increase is detected while the voltage value is below TMIN, no sleep period update is performed. The next sleep update will occur the first time when the voltage value becomes greater or equal than TMIN.

Similarly, if the voltage increases above TMAX and the sleep period value is greater than SMIN, the sleep rate adaptation function is still called each time a voltage increase is detected, having as a result a decrease of the sleep period. The new sleep period is compared with SMIN, taking its value when it becomes smaller than it. If a voltage decrease is detected while the voltage value is above TMAX, no sleep period update is performed. The next sleep update will occur the first time when the voltage value drops below TMAX or is equal to it.

In order to better translate all the ideas above into pseudo-code, three new functions are introduced: BelowRange(V), InRange(V) and AboveRange(V). Each of them returns true or false, comparing the voltage it receives as argument with TMIN and TMAX. To be more specific, BelowRange(V) returns true when V < TMIN and false otherwise. InRange(V) returns true when TMIN ≤ V ≤ TMAX and false otherwise. AboveRange(V) returns true when TMAX < V and false otherwise.

The complete algorithm is defined in the pseudo-code below.

Algorithm 2 Adaptive Algorithm 2

```
procedure SleepAllocation
    S ← initial
    for each sleep time slot do
        ΔV ← V − PreviousV
        if ΔV > 0 then
            if InRange(V ) or AboveRange(V ) then
                S ← S − S/α
        if ΔV < 0 then
            if InRange(V ) or BelowRange(V ) then
                S ← S + S/α
        if V < CRITICAL then
            Disable sending data
        if S < SMIN then
            S ← SMIN
        if S > SMAX then
            S ← SMAX
    end for
end procedure
```

In the next sections the experimental test setup is presented. The results yielded by each algorithm are evaluated on the Sparrow v3 sensor nodes on which they were implemented.

IV. Experimental Setup

The system integrates, roughly, four parts. First of all, the wireless sensor node, Sparrow v3. Second, a solar panel, used for solar energy harvesting. The third component is a DC to DC converter circuit, which increases and regulates the low voltage it receives from the solar panel. Last, but not least, the regulated voltage is used for charging a super capacitor

The current paper is based on the work done by Marin et al. [6], and the same wireless sensor node architecture, Sparrow v3, was described in detail in [7].

Sparrow v3 is a WSN equipped with three types of sensors: temperature, relative humidity and ambient light. Its microcontroller is an ATmega128RFA1 [8], which incorporates a 2.4GHz radio transceiver, compatible with IEEE standard 802.15.4. ATmega128RFA1 is an 8-bit architecture microcontroller with low power consumption starting from 250nA while in deep sleep mode. Detailed power consumption levels are illustrated in Fig. 2.

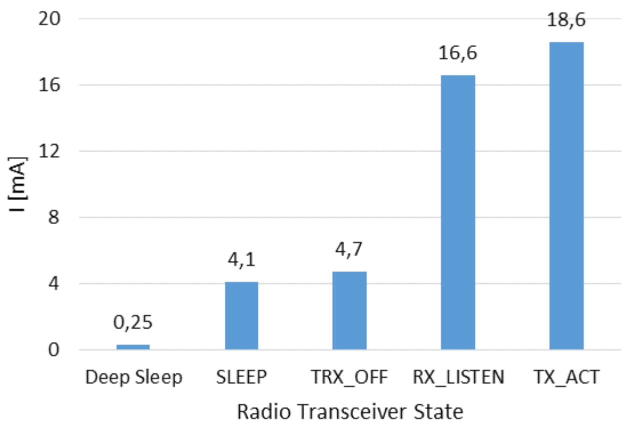

Fig. 2. Atmega128RFA1 power consumption states

As already mentioned, the energy storage devices are super capacitors which were chosen instead of using batteries for multiple reasons, the most important being its increased cycle life. A super capacitor can survive to up to 500,000 recharge cycles, without having its capacity severely reduced on the road. Batteries can last only a few hundred recharge cycles, but other issues are also encountered while their age increases, such as chemical reactions that may reduce or even break the battery functionality. For the current project two types of super capacitors were used: 3F/5V, and a larger one of 15F/4.6V.

The solar panels used for the experimental setup were much larger than the IXOLAR^{TM}SolarBITs previously used [9]. Also, the voltage and current outputs of the panels in this article exceed greatly the outputs of IXOLAR^{TM}SolarBITs. The latter ones occupied a total surface of 1.54 cm^2 and had an output of 0.51V, 39mA (XOB17-12x1) and 1.53V, 11.7mA (XOB17-04x3). These panels proved to be a good choice if the wireless sensor node was kept outdoors, harvesting enough energy for its needs in those settings.

However, these panels were not suitable for an indoor scenario in which the system is capable of operating uninterruptedly in lower light conditions. To achieve this goal larger panels were needed, capable of harvesting enough solar energy in this new situation. One of these two new solar panels has a surface of 61.75 cm^2, the voltage and current output being of 2V and 100mA, respectively. The other one has a surface of 67.5 cm^2 and an output of 18V and 15mA. From now on, these solar panels will be referred to as Panel A and Panel B.

The DC to DC converter circuit plays a very important role. First of all, it increases the voltage generated by the solar panels, voltage that is most of the time much smaller than that of the super capacitor. Without this voltage increase, solar energy cannot be harvested unless the solar panel generates a voltage greater than that of the energy storage device. Second of all, when having a low leakage current in shut-down mode, the DC to DC converter circuit plays the role of a diode, preventing the super capacitor from discharging.

Two different DC to DC converters have been used in the experiments described in this article. The one coupled with Panel A is based on BQ25504 [10]. The second one, coupled with Panel B, is based on LTC3129 [11]. From now, the first panel - DCDC pair will be referred to as Setup A and to the second pair as Setup B.

An important issue that needs to be mentioned is that the DC to DC converter must be chosen in accordance to solar panel voltage and current output. Panel B could not be coupled with the DC to DC converter based on BQ25504, due to the fact that specifications show that an 18V input would exceed its operational limits.

In previous published work, a DC to DC converter circuit based on LTC3105 [12] was used. Similar as before, to the pair XOB17-12x1 - LTC3105 will be referred to as Setup O1, XOB17-04x3 - LTC3105 pair being Setup O2.

The increase in charge rate was observed for all previously mentioned setups when subjected to the same constant illumination conditions. Each setup was exposed to a halogen lamp with the same luminous intensity. For each setup, the increase in voltage from a minimum value VMIN ≈ 2V to a maximum of VMAX ≈ 3.4V was recorded. The storage element used in these tests was the same 15F/4.6V super capacitor and all experimental setups run the same application: sleep for one minute, wake up to read data from sensors, compose a package of 13 bytes which is sent to the gateway, then resume sleep.

Besides monitoring the charging rate, another goal of this application was to consume as less energy as possible, in order to mitigate the impact of power consumption on the charging rate, making the results of the experiment more accurate. For this purpose the package size was shrank to 13 bytes, while using 127 bytes for other measurements.

In Fig. 3 the charging rates of Setup A and Setup B are compared. It can be observed that the charging rate is much better for the first setup, which can charge the energy storage devices two times faster than the second setup. It is shown later in the article how this difference affects the sleep adaptation algorithm of nodes using Setup A and Setup B.

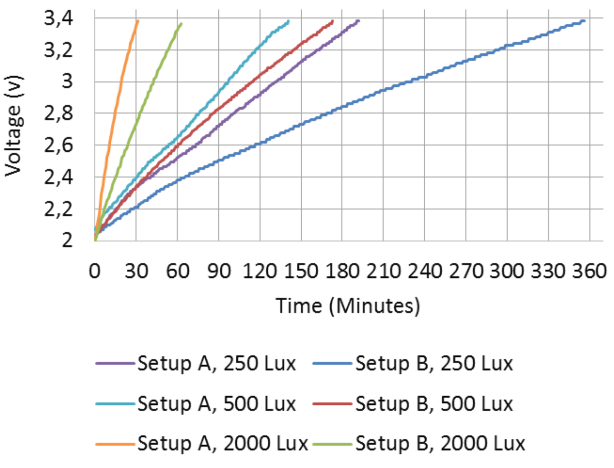

Fig. 3. Charging rate comparison between Setup A and B

Another important aspect which needs to be mentioned is that both setups behave well in low lightning situations, the super capacitor being fully charged in three to six hours even when the illuminance is as low as 250Lux.

These results could not have been obtained with any of Setup O1 or Setup O2 in similar conditions, making them unsuitable for indoor placement of wireless sensor node solution.

Fig. 4 shows the difference between the current charging rate of Setup A and Setup B, on one hand, and the charging rate observed when using Setup O1 and Setup O2, on the other hand.

It can be observed that for the same illumination conditions, with the newest setups the charging rates grow with up to 40 times compared to the first approach. This observation further underlines the efficiency of the current setups for indoor placement, when compared to the old approach. The goal of harvesting as much energy as possible in low lighting conditions is hence met.

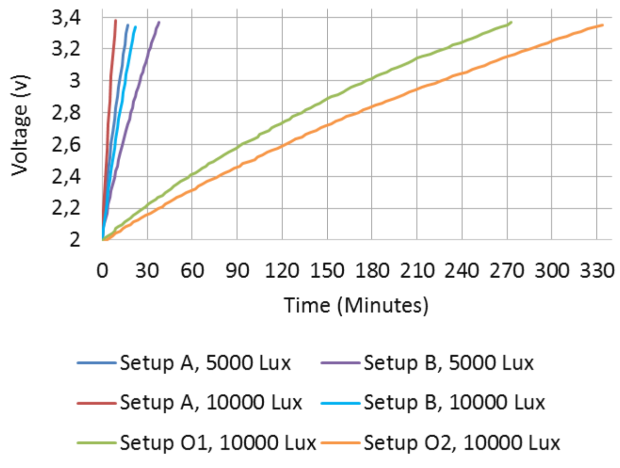

Fig. 4. Charging rate comparison between Setup A, B, O1 and O2

In the next section the software development related to the work discussed in this paper is presented. Furthermore, the structure of the data frames sent by the monitoring nodes to the gateway is shown.

V. Software Implementation

As previously stated, ATmega128RFA1 incorporates a radio transceiver which is compatible with IEEE standard 802.15.4. This is the standard over which ZigBee, a popular high-level communication protocol, is implemented. It would have seem a good idea to use ZigBee for the project, since this is possible and there is a lot of support available. The chosen approach was to implement a communication protocol which is highly optimized for low power operation and long duty-cycling, in order to control power consumption as well as possible.

The format of the messages sent by the monitoring nodes to the gateway is described in Fig. 5. The preamble of the package retains details about its size. This information is mandatory for the transceiver because it needs to know how much data there is to be sent. Of course, since the size is represented on only 8 bits, it means that ATmega128RFA1 can't send packages larger than 256 bytes. However, this is more than sufficient, as the 802.15.4 standard specifies a maximum package size of only 127 bytes. The next two bytes are reserved for the node's address while next 32 bits retain a message sequence number. The nodes are capable of creating large-scale networks with many years of indoor monitoring, so these fields need to be sized accordingly. Following, there are four fields of one byte each, for storing sensor data: temperature, relative humidity, light and voltage. In the end, a two bytes CRC is padded, for integrity checking on the gateway. The CRC verification is made automatically, in hardware, a bit from a specific register being set accordingly if the package CRC field is different from the received message's CRC.

size	address	seq_nr	temp	humid	light	volt.	crc
8bits	16bits	32bits	8bits	8bits	8bits	8bits	16bits

Fig. 5. Message format

Larger 127-byte messages are also employed. Their format is very similar to the one shown in Fig. 5, the only difference being that dummy data is added between the voltage and CRC frames. The reason for this artificial increase in packet size is to test the system under different loads in order to be able to better see the impact of the proposed adaptive algorithms.

VI. TESTING ENVIRONMENT CHARACTERIZATION

The performance of the algorithms and setups described in the previous two sections was evaluated by running experiments in various illumination conditions. Figure 6 highlights the illumination conditions during a clear August day and the voltage variation for the Setup A and Setup B.

The nodes charged until the light intensity reached the peak of approximately 500Lux at noon, the voltage level reaching its maximum for the day at approximately 3V. After that, the node began to drop energy in the evening, after sunset. Both nodes used a super capacitor of 3F/5V in this experiment. The charging rates below are not equivalent to the ones observed in the experiment detailed in Section 4, due to three reasons.

The first one is that smaller capacitors are used here, so the voltage would vary differently in the same lighting conditions as above (the charging rate would be much higher). Secondly, the longer 127 bytes messages are used, having as a result a much higher impact on power consumption for a sent package. Also, the sleep period is not fixed to one minute, as in the previous scenario.

This voltage monitoring was performed with the nodes running an adaptive algorithm which modified the sleep time when a voltage increase was detected. The period for which the nodes were sleeping was less than a minute for most of the time when the voltage maintained its increasing rate.

Compared to the results of the experiment in Section 4, where Setup A behaved overall better than Setup B, the results shown below may seem contradictory, since it may be thought that Setup B had better harvesting results.

The reason for which this happens is because Setup B sends fewer messages than Setup A (its sleeping interval is always longer than the one of Setup A). In order not to complicate the chart, the sleep period was not added in the representation. Charts with sleep time variation compared with voltage variation during a period of multiple days will be shown in the next paragraphs. From this moment the illumination conditions presented will be referred to as Environment A.

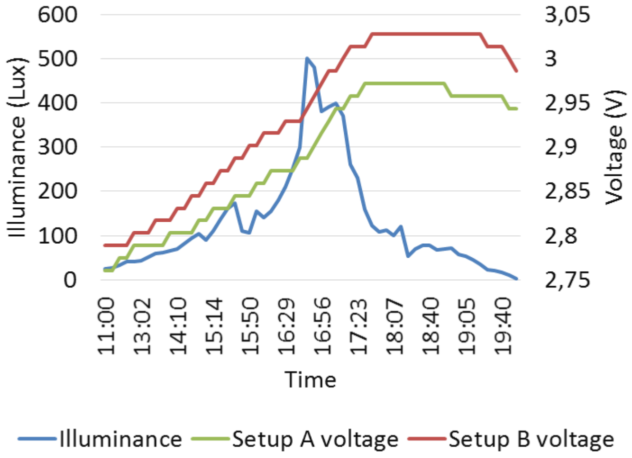

Fig. 6. Environment A illumination conditions and voltage variation

The experiment was repeated in a different environment, with lower maximum light intensity, but with higher average intensity. As it can be observed from Fig. 7, for a similar day of August, with maximum illumination between 300 and 350Lux the nodes charged and began to drop energy only in the evening. Setup A charged in this second experiment a larger capacitor, of 15F/4.6V, this being the reason for which its voltage variation looks so different than the one of Setup B, which uses a capacitor of 3F/5V.

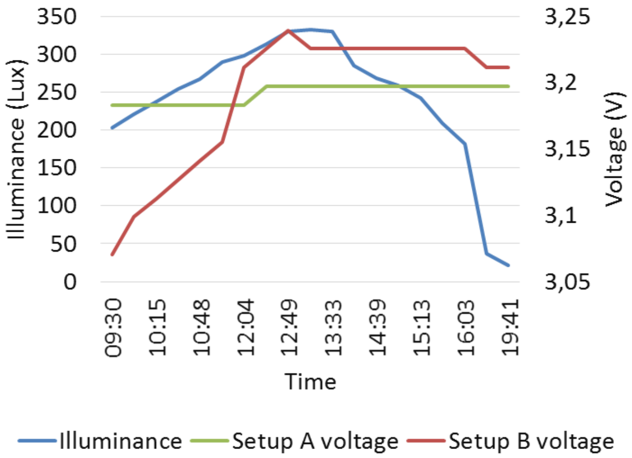

Fig. 7. Environment B illuminance conditions and voltage variation

Both systems harvest more energy in this second illumination conditions compared to Environment A. It can be seen that in the conditions presented in Fig. 6, the illumination goes over 200Lux for a period of only one hour, while here the same thing happens for most of the day, as shown in Fig. 7. From this moment, the illumination conditions presented in the figure below will be referred to as Environment B.

In the next section the adaptive algorithms' behaviour in the two previously described environments is presented.

VII. Experimental Evaluation

Algorithm 1 first results are presented in Fig. 8. The blue dotted line shows how the sleep time adapts with the voltage level changes caused by the day-night cycle. For the first experiment the node was operating continuously during a five day period, without any dead times.

The thresholds were set at the following values: CRITICAL=1.9V, TMIN =2.11V, TMAX =3.23V. The initial sleep increase factor INC is set at 10s and β is 2. Minimum sleep time SMIN is 1s and maximum sleep time SMAX is 300s or 5 minutes. The sleep time had a maximum of 140 seconds, so it never reached SMAX.

On voltage increase sleep time reaches SMIN in the afternoon corresponding with the highest illumination values shown in Fig. 6. The voltage for both Setup A and Setup B, shown in Fig. 9, oscillates between 2V and 2.5V. Voltage readings from each day at same hour were compared and the conclusion was that the ENO principle is respected, the node does not lose energy on long term.

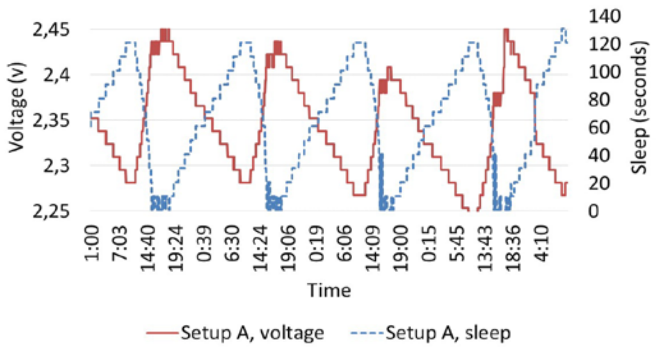

Fig. 8. Algorithm 1. Environment A. Setup A

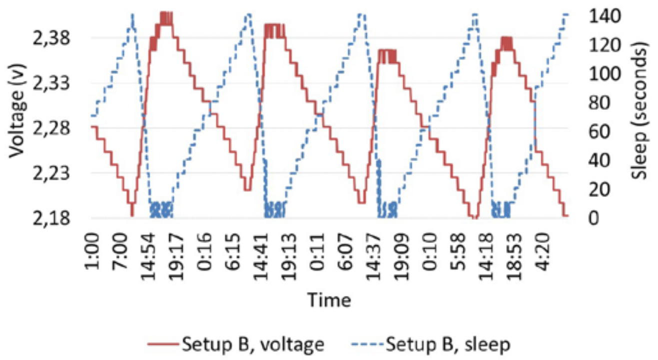

Fig. 9. Algorithm 1. Environment A. Setup B

In Environment B the voltage levels shown in Fig. 10 show how the algorithm behaves for a setup with the large capacitor of 15F. Due to the higher capacity, for this node the voltage variation is slower and sleep time as well does not vary too much, remaining in the 30 to 50s range. Fig. 11 details the voltage-sleep variation for Setup B. Because the average light intensity is higher in Environment B the sleep period is consistently at 1s during the day. At the same time, voltage remains at the maximum value, in this case of approximately 3.3V, higher than TMAX for longer periods during the day. The node utility is increased without loss of energy. Again, for the latter experiment the ENO condition is respected.

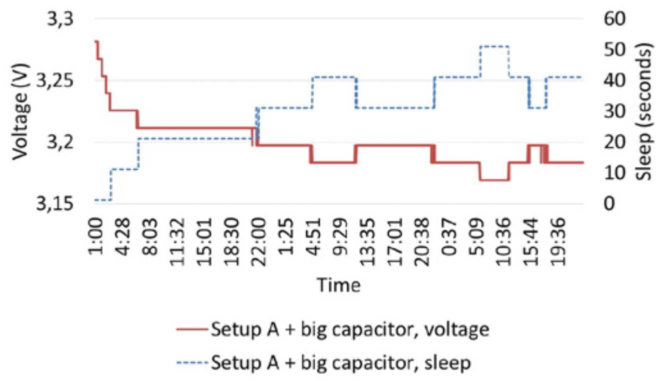

Fig. 10. Algorithm 1. Environment B. Setup A. Big super capacitor.

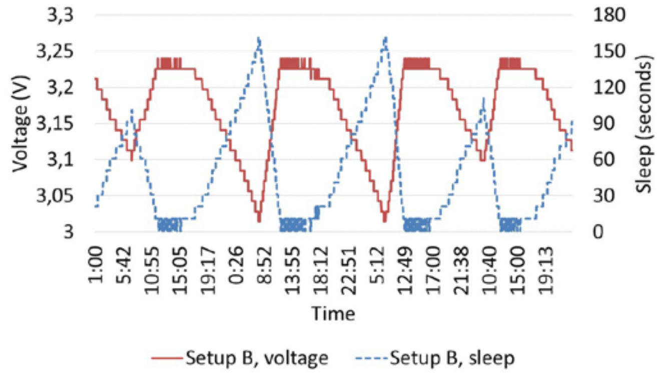

Fig. 11. Algorithm 1. Environment B. Setup B

Algorithm 2 tests were first conducted in Environment A, with an α value of 1/2. Other thresholds were set as following: SMIN=10 seconds, SMAX=1800 seconds, TMIN=2.8V, TMAX=3.0V and CRITICAL=1.9V. An α value of 1/2 means that each time a voltage variation was detected, the sleep time was modified accordingly, increased or decreased, with a difference of half compared to its current value. Also, TMIN and TMAX values suggest that desired voltage levels need to be kept in the [2.8V, 3.0V] range.

The results with the current threshold values were not so good, as it can be seen in Fig. 12 and Fig. 13. The measurements taken during a period of five days show a clear descendent trend for the voltage. Another observation may be that both nodes report the same voltage values in the last day of the experiment, even though Setup B began its monitoring with an advantage of 0.2V compared to Setup A. Furthermore, it can be seen that in the case of Setup B the sleep time reaches the value of SMAX two times, while Setup A stays far from it. These differences are happening due to the much higher energy harvesting for Setup A. The node with Setup B tries to keep its voltage value in the range requested by the thresholds, increasing its sleep time to maximum after multiple days of continuous voltage decrease compared to the previous day. The same trend can be observed for Setup A as well, but what makes the separation here is the smaller difference from the initial voltage.

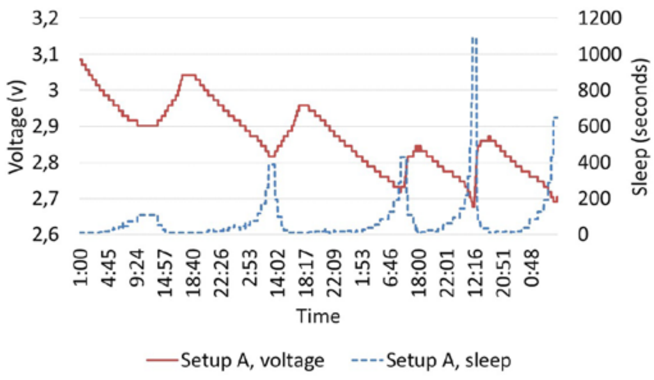

Fig. 12. Algorithm 2. Environment A. Setup A, α = 1/2

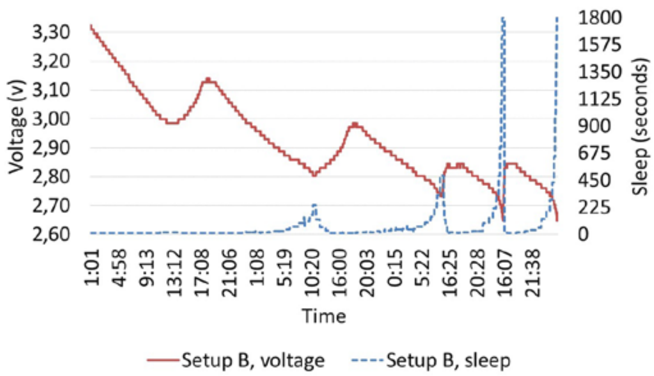

Fig. 13. Algorithm 2. Environment A. Setup B, α = ½

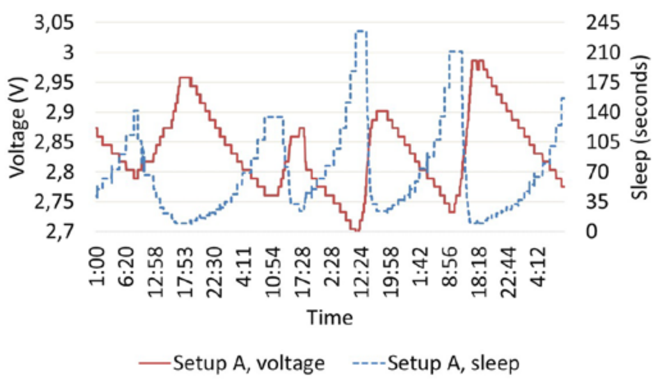

Fig. 14. Algorithm 2. Environment A. Setup A α = 1/4

The node sleep time moves too fast from its maximum SMAX value to its minimum SMIN value, hence making the node to send too many packages as soon as some harvesting begins. It is true, on the other hand, that the sleep time goes quickly from SMIN to SMAX as well, but considering the obtained data, where the nodes' sleeping values are very low for most of the day, going to minimum as soon as some energy harvesting begins, a potential improvement in lowering the value of α can be seen. This will have as a result a slower transition from SMIN to SMAX and back. It is true that the sleep time will rarely to never reach SMAX again in this conditions (in the testing environment conditions), but the same thing will happen when referring to SMIN, as well.

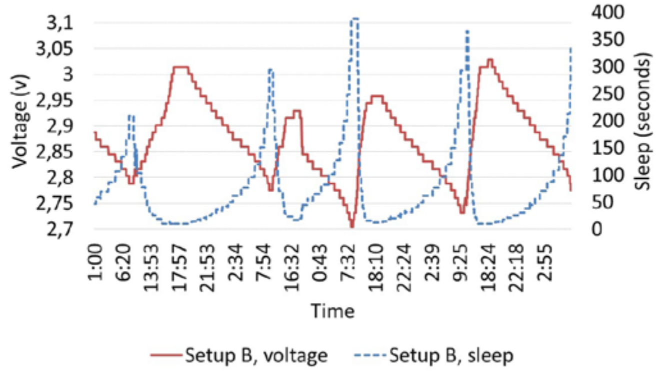

Fig. 15. Algorithm 2. Environment A. Setup B, α = ¼

Considering the previous results the second algorithm needs improvement. The threshold α was modified from 1/2 to 1/4, as in order to make the sleep adaptation less steep than the previous one. The obtained results were much better, for both setups, and can be seen in Fig. 14 and Fig. 15. It can be observed that the voltage values have a better trend, the sleep time varying, at the same time, in a less steep manner. Both setups behaved well, as the voltage value was kept in the requested range. One could observe, again, that Setup B reaches sleep times of greater value than Setup A, the differentiator being the solar panel capabilities in terms of voltage and current.

VIII. Conclusions

Development of harvesting solar energy technologies in WSN motivated this work. Solutions of efficiently using the harvested energy to power a Sparrow v3 sensor node in a monitoring application were successfully developed and tested. The proposed algorithms make use of duty-cycling to alternate between deep sleep mode and active mode in the node.

The article shows how performance of the described algorithms can be improved, their impact on the overall node lifetime, allowing the network to continue functioning for several consecutive days in fluctuating illumination conditions. Furthermore, the authors try to find the best solution from a hardware point of view, making the Sparrow Wireless Sensor Network suitable for uninterruptedly indoor monitoring.

In terms of future work, improvements could be made from both software and hardware point of view. First of all, in the current work, only basic adaptive algorithms showed an actual improvement in network parameters. Better algorithms could be found, adding as a small drawback an increase in complexity. However, these algorithms could conceivably run on an improved hardware version of the architecture, which could employ state-of-the-art and energy efficient components (e.g. microcontroller, sensors). A second idea would be to make the node as smaller as possible, keeping its full functionality and continuous operation. Thirdly, multiple types of sensors could be added, in order to monitor more environmental parameters.

REFERENCES

[1] Kansal, A., Hsu, J., Zahedi, S., and Srivastava, M.B. (2007). Power management in energy harvesting sensor networks. *ACM Trans. Embed. Comput. Syst., 6(4).*

[2] Vigorito, C., Ganesan, D., and Barto, A. (2007). Adaptive control of duty cycling in energy-harvesting wireless sensor networks. *In Sensor, Mesh and Ad Hoc Commu- nications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on, 21–30*.

[3] Cammarano, A., Petrioli, C., and Spenza, D. (2012). Pro- energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks. *In Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on, 75–83.*

[4] Piorno, J., Bergonzini, C., Atienza, D., and Rosing, T. (2009). Prediction and management in energy harvested wireless sensor nodes. *In Wireless Communication, Ve- hicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on, 6–10*.

[5] Hsu, J., Zahedi, S., Kansal, A., Srivastava, M., and Raghu- nathan, V. (2006). Adaptive duty cycling for energy har- vesting systems. *In Low Power Electronics and Design, 2006. ISLPED'06. Proceedings of the 2006 International Symposium on, 180–185.*

[6] Marin, A.G. and Tudose, D. (2015). Energy independent wireless sensor network design. *In Control Systems and Computer Science (CSCS), 2015 20th International Conference on, 267–272.* doi:10.1109/CSCS.2015.94.

[7] Voinescu, A., Tudose, D., and Dragomir, D. (2013). A lightweight, versatile gateway platform for wireless sen- sor networks. *In Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition, 1–4*. doi:10.1109/RoEduNet.2013.6714202.

[8] Datasheet, 2014. *ATmega128RFA1 datasheet.* [Online] Available at: http://www.atmel.com/Images/doc8266.pdf [Acessed 30 October 2015].

[9] Datasheet, 2009. *IXOLARTMHigh Efficiency SolarBIT datasheet.* [Online] Available at: http://www.ti.com/lit/ug/tidu383/tidu383.pdf [Accessed 30 October 2015].

[10] Datasheet, 2015. *BQ25504 datasheet.* [Online] Available at: http://www.ti.com/lit/ds/symlink/bq25504.pdf [Accessed July 12th 2016].

[11] Datasheet, 2013. *LTC3129 datasheet*. [Online] Available at: http://cds.linear.com/docs/en/datasheet/3129fc.pdf [Accessed July 12th 2016].

[12] Datasheet, 2010. *LTC3105 datasheet*. [Online] Available at: http://cds.linear.com/docs/en/datasheet/3105fa.pdf [Accessed July 12th 2016].