Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 272849, 33 pages
http://dx.doi.org/10.1155/2013/272849

Research Article

An Adaptive Energy-Management Framework for Sensor Nodes with Constrained Energy Scavenging Profiles

Agnelo R. Silva,[1] Mingyan Liu,[2] and Mahta Moghaddam[1]

[1] *EE-Electrophysics, University of Southern California, Los Angeles, CA, USA*
[2] *EECS Department, University of Michigan, Ann Arbor, MI, USA*

Correspondence should be addressed to Agnelo R. Silva; agnelors@gmail.com

Received 6 June 2013; Accepted 19 August 2013

Academic Editor: Davide Brunelli

Modern energy harvesting systems for WSNs involve power scavenging sources, rechargeable batteries, and supercapacitors. Typical energy-management systems calculate/predict the remaining energy stored in a node, and associated actions are dispatched involving the networking protocols. However, long-term characteristics of the mentioned hardware components are typically neglected preventing the achievement of very long maintenance-free lifetimes (e.g., >5 years) for the nodes. In this work, a systematic analysis of this problem is provided, and an open energy-management framework is proposed which promotes (a) the nontraditional combination of primary cells, supercapacitors, and harvesting systems, (b) the concept of a distributed system inside a node, and (c) the adoption of the dual duty-cycle (DDC) operation for the WSNs. The DDC's core component is a cross-layer protocol implemented as an application-layer overlay which maintains the operation of the network under very high energy efficiency. Its trade-off is the reduction of the network throughput. Therefore, the DDC system has mechanisms that dynamically switch the WSN operational mode according to application's needs. Detailed guidelines are provided in order to allow the implementation of the solution on existing WSN platforms. The energy efficiency of the low duty-cycle mode of the solution is demonstrated by simulated and empirical results.

1. Introduction

Despite the possible existence of power scavenging sources for wireless sensor networks (WSN) nodes, this fact does not necessarily imply a longer lifetime or a high reliability level for such nodes. For instance, when a photovoltaic cell is used, the energy harvesting process is typically not continuous or stable. Moreover, when rechargeable batteries are part of the energy harvesting system, the lifetime of the node is ultimately dictated by the age or by the number of charge cycles of those batteries, among other factors. Even adopting very well-controlled charging procedures, typical secondary cells for WSN nodes have a lifetime smaller than 3 years. A potential solution to achieve a 5- to 10-year maintenance-free solution is the adoption of a battery-free design, as proposed in [1], where supercapacitors are used as temporary energy reservoirs. However, the challenge of such solution is to sustain a certain level of reliability when the capability of the power source is insufficient for the node operation.

The above issue is aggravated when the duty-cycle of the node is not solely governed by the main application. This is the case when the node must also actively collaborate in the network. Moreover, the effort to incorporate energy consumption metrics into existing physical and higher-layer networking protocols is still a challenge, and typically such provision is not implemented in commercial WSN solutions. As a result, it is very difficult to achieve realistic long lifetime for the nodes in conjunction with relatively high reliability levels. In this work, an adaptive and flexible framework for sensor nodes with constrained energy scavenging profiles is presented. This energy-management framework has hardware and software components, and a significant emphasis on integration is given in this work in order to facilitate the partial or integral adoption of the proposed framework on existing WSN platforms.

The motivation for this work is associated with the goal of having a *reliable* WSN solution with a lifetime between 5 and 10 years. That is, during this period of time, no human intervention is expected due to power depletion of a node. It will be shown that, to achieve this goal, the complexity level of the solution at the design-time is relatively high. Also, the initial cost of a node is expected to be at least 30% higher than a traditional off-the-shelf node. However, the practical functionality and the total ownership cost (TCO) of the system at long term can be very attractive. Note that the advocated vision in this work diverges from the traditional concept that a WSN comprises hundreds of very low-cost nodes each one individually with a relatively high probability of failure. On the contrary, the focus of this work is on the achievement of a very high-quality and controlled solution.

The paper begins with the presentation of the *energy effort tripod* and *energy control loop* concepts in Section 2. It is highlighted that effective energy savings for WSN nodes potentially depends on a balanced solution in terms of hardware, network, and application demands. In Section 3, the foundations of the proposed framework are discussed: (a) the optional (but recommended) use of primary cells associated to harvesting systems, (b) the advantages of a distributed system inside a node, and (c) the adoption of the dual duty-cycle operation (DDC) for WSN nodes. The core part of the proposed framework is a cross-layer network protocol which is presented in Section 4. This protocol is implemented as an application-layer overlay on top of existing WSN solutions. Such overlay mechanism can be dynamically activated and deactivated in order to allow the network to achieve the best performance while satisfying existing energy constraints. Many of the components of the proposed framework are in fact part of a long-term and ongoing project involving one of the largest outdoors WSN deployments still in operation [2–4]. The field results of this project in conjunction with simulated outcomes are reported in Section 5. A discussion related to the integration of the framework with other ongoing WSN research efforts is provided in Section 6, and the paper is concluded in Section 7.

2. Energy Management in WSNs

In this section, typical pitfalls and challenges in the design of energy systems for WSNs are discussed. Next, the important *energy effort tripod* and *energy control loop* concepts are introduced.

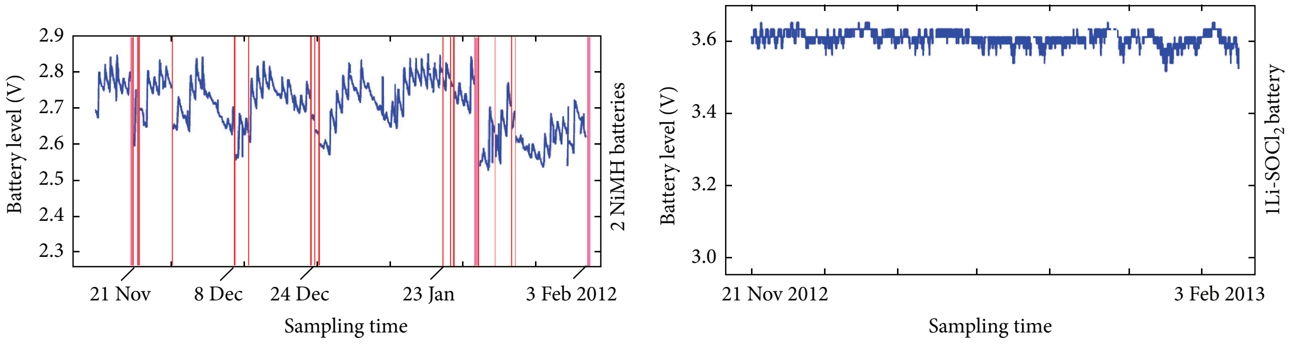
2.1. Design Challenges and Pitfalls.

Many well-designed projects fail due to small details and incorrect (but generally accepted) assumptions. Therefore, before presenting the proposed framework, it is important to highlight some aspects associated with the current state-of-the-art technology on energy harvesting systems for WSNs.

WSN Design besides Long Lifetime and Reliability. One critical pitfall associated to the energy aspect of WSN designs is to perpetuate the original vision of a WSN with hundreds to thousands of very cheap nodes [5] where a high rate of node failures is actually expected. Although such vision can still correspond to the needs of some applications, a quick investigation at the current WSN deployments around the world reveals a different trend for WSNs. For instance, few existing long-term networks actually have more than 50 nodes. More impressive is the ongoing success of the infrastructure-based WSN solutions (star or tree topologies), such as the ones based on IEEE 802.15.4/ZigBee [6, 7]. A higher node reliability is typically crucial when WSNs move from ad hoc to infrastructured architectures. Accordingly, to this current trend, a significant emphasis in this work is given to the reliability of the nodes. In this context, the term *reliability* is associated with the goal of having nodes that rarely become unavailable due to a *noncontrolled* power depletion.

Energy Scavenging Does Not Imply a Perpetual Lifetime. Another pitfall associated to the design of WSNs is to consider that the adoption of an energy scavenging system is automatically associated to an endless node lifetime. Besides the need to consider the life expectancy of the sensing components, such as a humidity sensor or a soil moisture probe, typical power systems can rarely achieve a 5-year lifetime due to a plurality of reasons discussed in this section. Therefore, the first step toward a successful low-cost WSN solution (in terms of functionality + reliability + long lifetime) is the investigation of the expected lifetime of each of the components of an node. In general, the reported premature death cause of WSN nodes is the energy subsystem, in particular the batteries. Primary (nonrechargeable) batteries typically have a very short lifetime in WSNs [8, 9]. However, it is also important to have in mind that secondary (rechargeable) batteries potentially have a lifetime smaller than 2-3 years.

Possibility of Adopting a Nonrealistic Energy Model. A significant number of WSN papers present three regular omissions or inaccuracies regarding to the way the energy model is proposed or adopted. First, the transients, such as due to the activation/deactivation of a radio transceiver, are typically neglected as pointed out in [10]. Second, many values used as input parameters for the models are directly imported from the datasheets of the components without further consideration of the effects of integrating these components together. For instance, based on its datasheet, a radio transceiver module has a nominal sleeping current of 10 μA. However, it is observed that, once it is attached to a MCU, leak currents are detected which are many folds higher than the nominal sleeping current. Similarly, a voltage regulator can be included in the design of a WSN node, in particular when energy harvesters are also employed. However, many reported energy models do not include the energy cost of a possible voltage regulator. As a result, an energy model that disregards the existence of voltage conditioners can be drastically distorted (non-linearly) when it is adopted in a node. The severity of this statement can be illustrated by the following real-case. When both MCU and radio modules are sleeping, usually it is not possible to put the voltage regulator in sleeping mode (typically *shutdown* mode in the context of regulators). Therefore, rather than an expected amount in μW as the power consumption for the node (as expressed in many WSN energy models), the node can potentially have an effective sleeping consumption on the order of mW. Therefore, as one reduces the application duty-cycle, the adopted energy model reveals



(a)

(b)

Figure 1: Effect of subzero temperatures on secondary (a) and primary (b) cells (Ann Arbor, MI, USA) [9]. The recharging process of the secondary cells is impacted by low temperatures causing node failures (lines in the figure). Primary cells are more resilient to extreme temperatures.

its non-linear distortion once the node is still consuming a significant amount of energy even in sleep mode.

The third pitfall commonly observed in the WSN literature is associated with the expected lifetime of batteries. In general, it is assumed that 100% (or a close value) of the initial nominal energy will be actually available for the operation of the node. Some papers even justify this assumption by highlighting the fact that both MCU and radio support low voltage levels, such as 1.5 V. In practice, it is very hard to achieve values even close to 80% of the nominal energy capacity of the cell. Factors that invalidate the mentioned assumption are the self-discharging current, the aging of the battery, temperature, discharging regime, charging regime (for secondary cells), and so forth. As a rule-of-thumb, when the battery reaches its terminal state, a significant amount of energy (e.g., >25%) still remains inside the cell. However, only very low discharging currents are typically possible from that moment on. Note that even if the load affords a low voltage level, the bottom line is actually the constraint of having the load only draining very tiny currents. This is hardly the case in particular for radio modules in WSN nodes. Therefore, if the hardware/software solution embedded on that node does have any provision to use this *remaining* energy at the cell, the adopted energy model must only consider conservative values for the actual initial energy stored at the battery. In many cases, such conservative value is less than half of the nominal energy of the battery.

As observed, realistic energy models for WSNs are inherently complex, but they can be simplified if conservative values are adopted. Moreover, every time a new WSN platform is designed, a significant number of experiments involving the final hardware, different kinds of batteries, and realistic discharging regimes must be considered before an energy model can be proposed for that node and also the network. On the other hand, it is interesting to observe that energy models for batteryless solutions are typically reported as properly matching the application needs [11]. In general, it is the case because detailed empirical investigation is realized to justify the energy model in a critical scenario involving a small amount of available energy at the energy reservoir (e.g., a supercapacitor).

Lifetime of Rechargeable Batteries. Typical secondary cells used in WSNs require special attention because besides their inherent *shelf* lifetime (e.g., <3 years) there are other factors that can drastically reduce their lifetime. For instance, the maximum number of nominal charge cycles is usually smaller than 1,000 considering the kind of cells typically reported for WSN nodes. Therefore, without a careful control of how and when the charge cycles are performed, the lifetime of such cells can be realistically smaller than 1 year. Moreover, temperature is a critical factor in particular for secondary cells. In general, extreme temperature can drastically degrade the cell's performance. For instance, in [9], it is reported that at sub-zero temperatures, many solar-powered nodes stopped the charging process followed by long periods of network inactivity, as shown in Figure 1. A solar panel completely covered with snow is one of the potential reasons for the mentioned issue. However, for this particular case study, the inability of the secondary cell to be charged at subzero temperatures was the main reason behind the functional failures. It is also observed in Figure 1 that nodes powered by a primary cell (Lithium Thionyl Chloride, in this example) are not affected by extreme temperatures.

Solid-State Batteries and WSNs. Solid-state batteries are a recent technological advance that can impact the design of future WSN solutions. These secondary cells are claimed to have a lifetime between 5 and 10 years and a maximum number of charge cycles between 5,000 and 10,000 [12]. Based on these preliminary values, it is possible to envision a reliable WSN solution with a very long lifetime based on such batteries. Nonetheless, these cells have three significant drawbacks: high cost, low-energy density, and low-power density. While the former aspect can be only a matter of time as a function of the industrial scale, the remaining aspects reinforce the need of a careful designed energy-management system if such cells are expected to be used in WSN nodes.

Lifetime of Low-Cost Outdoor Energy Harvesters. The life expectancy of low-cost energy harvesters for outdoors is typically not informed by the manufacturers. For instance, to date, manufacturers of micro wind turbines do not provide such information. Similarly, although relatively big solar panels (e.g., >30 cm × 30 cm) are typically robust and have a realistic lifetime of more than 3 years, it is not the case in relation to small solar panels used in WSN nodes. We performed outdoor tests for more than 2 years with different types and models of small solar panels. Unfortunately, the results were very disappointing: the majority of the panels presented a significant performance deterioration in less than 1 year. The majority of them changed their glossy surface by a white porous surface where dust easily accumulates. In one of the sites, which experiences high temperatures (e.g., >40°C), more than 10% of the panels cracked. To the date, we did not find off-the-shelf small solar panels with the typical robust encapsulation found at bigger panels. Another critical aspect in relation to small solar panels left unattended outdoors is the dirt left by birds. In our outdoor deployments involving sites in three USA states, we observed the same phenomenon: a small solar panel mounted on top of a pole is a typical place where birds choose to temporarily rest. Without a proper protection against the birds, such solar panels potentially require periodic cleanness. The bottom line is the importance of evaluating the robustness of the components of a harvesting system before assuming a perpetual lifetime for a node.

2.2. The Concepts of Energy Effort Tripod and Energy Control Loop. In the previous section, some aspects related to the energy subsystem of a WSN node are highlighted, and it was shown that the achievement of a maintenance-free solution for periods of more than 5 years is not a trivial task. At that analysis, the network and application aspects are not considered. However, the adoption of an energy-management solution requires the integration of energy effort in terms of hardware, the network, and the application. Accordingly, the focus of this section is to discuss how these three aspects are properly integrated in an energy-management framework. We will conclude that the knowledge related to the energy state of the nodes is paramount.

The term *framework* is defined as a broad outline of interrelated items, not a detailed step-by-step set of strict guidelines. The advantage of this design approach is manly the gains in terms of flexibility: one is exposed with some underline concepts and ideas and can easily adapt them to his problem or environment, in this case, a certain WSN platform. Accordingly, the basic concepts available at the WSN energy-management literature are summarized by two pictures presented in this section.

The first concept, the *energy effort tripod*, is illustrated in Figure 2. Assuming a certain limited energy level for a node, an efficient way to achieve functionality and reliability for a very long period of time is to balance efforts in terms of hardware, network algorithms, and application demands. For instance, if advances in the hardware/software of a node allow the reduction of the sleeping energy consumption of a node by one order of magnitude, such effort is potentially voided if the effective duty-cycle of the node (due to the network, the application, or both) is still very high. Similarly, a significant reduction of the network overhead can have little energy impact compared to a very high and frequent application demand (e.g., multimedia data traffic). Therefore, the starting point is to limit the demand of the main application and to define possible acceptable levels of Quality of Service (QoS) for the nodes, a group of nodes, and the network. As expected, service metrics are required in energy-management systems. Such metrics involve data latency, volume of data traffic, frequency of data bursts (e.g., scheduling in data-driven applications), data loss acceptance/levels, and so forth.

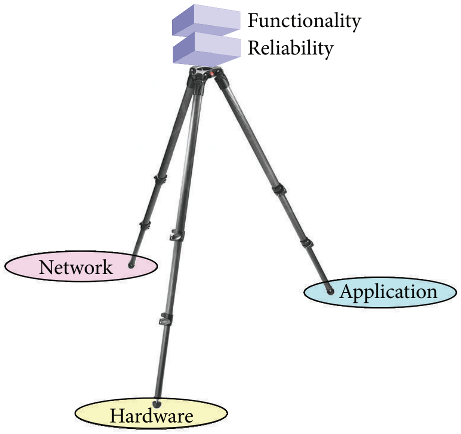


Figure 2: Energy effort tripod concept: coordinated efforts involving hardware, network algorithms, and application demands lead to an energy-balanced and efficient WSN solution.

Once the application demands are clearly defined and realistically constrained considering the energy systems and power sources available for the nodes, the next step is to evaluate how the network and the hardware of the nodes can be improved or, in other words, *balanced* with the application demands. In general, the adoption of a very flexible WSN solution (not tailored to a certain category of application) is also associated with energy-hungry network protocols. For instance, consider a WSN application that every 5 minutes monitors the infrastructure of a bridge. One strategic question to be considered in this case is related to how the network protocols can be optimized considering a static topology and also a fixed monitoring schedule.

Similarly, a very energy-efficient hardware module may not be a balanced solution. It is the case when the energy-saving mechanisms provided in the hardware can actually impact the functionality and reliability of the network and ultimately the main application. For instance, consider the case of a batteryless node which adopts a combination of solar panel and supercapacitors. During the daylight, the hardware of the nodes is functional and energy-efficient and the intensive collaboration in the network is not impacted. However, during the night, the behavior of the nodes can drastically change: a batteryless node may not be capable of performing regular transmissions multiple times per second, even under a very low duty-cycle (e.g., <1%) regime. Nonetheless, consider the fact that the majority of the current WSN network protocols operate assuming that the node wakes up multiple times per second. Therefore, it is clear that the power system design and the adopted network solutions are not properly balanced in this example.

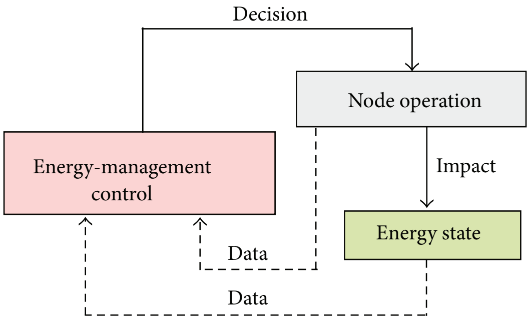


Figure 3: Energy control loop concept: the operation of a WSN node is regulated by its energy state. The decisions are triggered by an energy-management module that can be implemented internally in the node, at the network level, in a centralized data server, or by a combination of these options.

In some cases, the previous illustrated batteryless solution for WSN nodes imposes a certain level of data latency which is incompatible with the application requirements, and again the balance is not achieved. The main point behind this discussion is not about advocating in favor or not in relation to a certain technology. It is actually related to what can be adjusted in the WSN design in order to obtain a proper balanced solution in terms of energy, functionality, and reliability. In many cases, an optimized solution is complex because it is only achieved by combining enhancements in the hardware, in the network, and also in the application (i.e., relaxing the required QoS metrics). For the latter aspect, it is clear that *control* is necessary, and, in fact, this is the role of energy-management modules, as illustrated by the next conceptual figure.

The second concept to be discussed is called *energy control loop*, as illustrated in Figure 3. Such control can be performed at the node level, in a portion of the network, by means of a central data server, or by a combination of these options. As shown by Figure 3, the operation of the node, such as the activation of an energy harvester, the activation of the radio module, or the way the node behaves in the network, is governed by the decisions of an energy-management module. As expected, such actions impact the energy state of the node, such as the remaining energy available for the node. Therefore, a proper design goal is to have an energy-management module that receives feedback related to the operation of the node and also energy-related data. Note that the dashed lines used in the figure are an indication that such feedbacks are optional. Specifically, it is possible that the energy-management module makes inferences about the energy state of a node without receiving explicit feedback data from the node. Next, we will see how energy-management control efforts can be realized at node, network, and central system levels.

Consider a scenario where part of the energy-management processing is performed inside the node, as illustrated in Figure 4(a) which is related to the activation/deactivation of an energy harvester. Such control can be realized purely in the node level without requiring any network activity. In this case, the energy state is related to the measurements of voltage levels and output currents of the power source. The maximum power point tracking (MPPT) technique is one example of such kind of energy control which allows the energy harvester to achieve its maximum efficiency.

The energy-management can also be realized by means of energy-aware networking protocols, as illustrated in Figure 4(b). Consider an example of a collaborative protocol that dynamically assigns certain roles (e.g., cluster head, router, data aggregation/fusion master node, etc.) to the best qualified nodes according to their remaining energy [13]. Observe in Figure 4(b) that two feedback data flows occur: one related to the application data and basic network functionalities and the other associated specifically with energy metrics. In some cases, the final decision related to the temporary role of a node at the network can be done at the node itself without further network activity. In other cases, the decision is performed by a specialized node that has a more holistic view of the network.

In many cases, the energy decisions performed at the node and network levels can be insufficient for the achievement of the expected QoS metrics. At the other extreme, there are cases where the quantity of the sensing data is far beyond the necessary level of information (e.g., too high sampling rate) and there is room for energy optimization. For instance, consider the case when it is enough from the viewpoint of the main application that only one-third of the nodes in a network simultaneously monitor a specific event. Typically, only the data server can take such decision because it is closely associated to the historical data flow from multiple sensor nodes and also with the data quality metrics stored exclusively at the data server. Accordingly, as shown in Figure 4(c), the main application at the central Data Server can issue commands associated to the scheduled activity for each node at the network [2, 14]. Observe that the feedback line in Figure 4(c) is omitted which seems to be counter-intuitive considering the fact that the current discussion is about the control *loop* concept. However, in many implementations of energy-management systems, the energy state of the nodes can be simply inferred based on the network activity of the nodes and explicit control feedback is not necessary.

Considering that this work is the presentation of a *framework*, not all possible forms of energy-management implementations are represented by the previous illustrations. Nonetheless, it is secure to state based on the investigation of related work that the majority of the very energy-efficient systems are actually a combination of efforts at the node, network, and central levels. In general, a complex design and higher implementation costs are expected. On the other hand, such approach is typically accompanied with the advantages of having a balanced energy solution as highlighted by the energy effort tripod concept. In the next section, the two discussed concepts will be translated in practical design guidelines for WSNs.

3. Energy-Adaptive Framework for WSN Architectures

In this section, the generic discussion in Section 2 evolves in more practical and flexible guidelines that will compose the

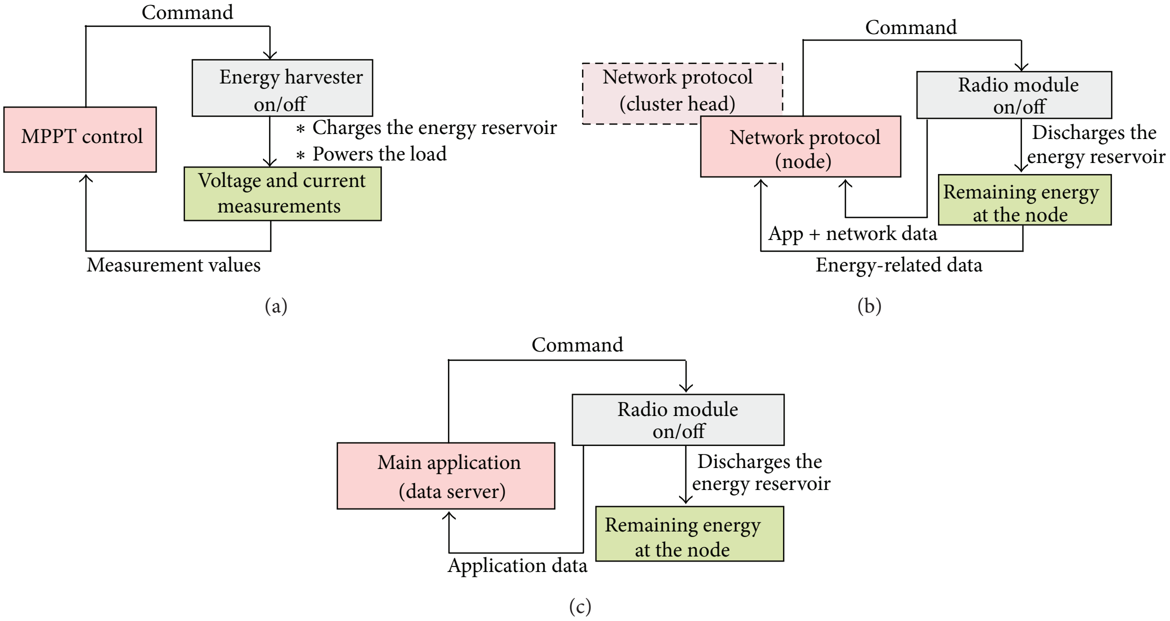

Command
Energy harvester on/off
MPPT control
* Charges the energy reservoir
* Powers the load
Voltage and current measurements
Measurement values
(a)
Command
Network protocol (cluster head)
Radio module on/off
Network protocol (node)
Discharges the energy reservoir
Remaining energy at the node
App + network data
Energy-related data
(b)
Command
Radio module on/off
Main application (data server)
Discharges the energy reservoir
Remaining energy at the node
Application data
(c)

Figure 4: Examples of energy-management efforts. (a) At the node level: optimized control of an energy harvester, (b) at the network level: the remaining energy of a node is used as criteria for its selection in network activities, and (c) at a centralized level: based on the sensing data received from the nodes, the data server defines what nodes will sense according to location/time scheduling.

proposed energy-adaptive framework involving WSN nodes with constrained energy scavenging profiles. It is important to highlight that this framework is not being proposed necessarily to substitute existing ones but to optimize such efforts in order to achieve very long maintenance-free periods of time for the nodes in conjunction with high levels of functionality and reliability. This section, which is almost one-third of this work, starts with a presentation of the three foundations of the framework. Next, a discussion about the characteristics of the energy-efficient low duty-cycle (LDC) operational mode are provided in conjunction with potential target platforms for the framework. Finally, high-level implementation guidelines are provided.

3.1. Foundation 1: The Strategic Use of Primary Cells.

Primary cells have high energy densities, typically 3 times in comparison with rechargeable batteries [9]. That is, for the same physical volume, primary cells provide higher energy capacity. Another advantage of a primary cell is the possibility of using almost 90% of its nominal energy capacity by means of proper techniques [9], in contrast with rechargeable batteries, as already discussed in Section 2.1. Moreover, primary cells are very resilient to extreme temperature outdoors. Finally, the typical shelf-aging of primary cells is around 10 years compared to 3 years of secondary cells. Nonetheless, for the best of our knowledge, this is the first time that primary cells are highlighted as an important basis for a WSN energy-management system that involves energy harvesters, in particular if the goal is to achieve a maintenance-free period of more than 5 years. Also, the inclusion of such cells increases the costs and the size of the WSN node. Therefore, it is very important to understand the context and assumptions behind this proposal, and such analysis is divided into four topics.

Optional Use of Primary Cells. The use of primary cells is closely related to the goal of long life solution while achieving high levels of reliability and functionality for the solution. However, there are cases that such provision is not necessary, and a batteryless solution fully satisfies the application requirements. For instance, consider sensor nodes that harvest energy from mechanical vibrations at the engines installed in an industrial plant. Potentially, the energy budget of these nodes can be sufficient for a realistic zero-energy, batteryless, and WSN implementation based solely on the mentioned harvester and supercapacitors. In this case, assuming that the energy harvesters also have a long lifetime, there is no need to implement almost the entire framework proposed in this work because in this case the nodes do not have a critical constrained energy profile. Similarly, the relative small costs of regular maintenance of nodes installed at indoors can justify the avoidance of energy-management mechanisms in particular for small networks. Therefore, the proposed use of primary cells is clearly optional and depends on the characteristics of each WSN application and its environment.

Primary Cells Have Low-Power Density. In practice, the goal of using 90% or more of the energy capacity of primary cells is seldom achieved in WSNs. Based on a careful study regarding this topic [9], potentially only 10% to 30% of the nominal energy capacity of a primary cell can be realistically used in typical WSN nodes. The main reason behind this fact is identified in the same work: these cells cannot sustain their nominal energy capacity if frequent peak currents (e.g., >15 mA) occur. In fact, typical WSN nodes have peak currents higher than >100 mA although the nominal transmission mode current is much smaller than this value. Therefore, the already mentioned 3-time energy capacity advantage of primary cells in comparison with secondary cells (assuming the same volume) is canceled by the peak current effect. To illustrate the point, assume another solution based on rechargeable batteries and the potential utilization of only 50% of their nominal energy. In this case, this energy reservoir can still have two times the effective energy capacity compared to the mentioned primary cell. Considering this analysis, it is not a surprise to read frequent reports about the need of exchanging primary cells in WSNs in regular periods of few months or even weeks [8]. Therefore, it is a common sense in the WSN community that primary cells must be avoided. On the other hand, in this work, we advocate a balanced hybrid power system where primary cells have their strategic role defined. However, as expected, such recommendation of using primary cells in the proposed framework only holds if the peak current effect can be mitigated as proposed in [9].

Achieving a Reliable Energy Harvesting System. The use of primary cells is introduced in the framework as a way to guarantee certain levels of QoS considering the occurrence of situations where the energy harvesting resources are not enough to sustain the functionality of the node, part of the network, or even the overall network. Ideally, an energy-management system would not have a battery or any other energy-related component that requires regular maintenance. But once batteries are used, the remaining energy at these reservoirs must be controlled. Without any energy control, the primary cells also become an uncertainty factor at the system which mitigates its importance to achieve a higher reliability level. The bottom line is that primary cells are only being proposed in this context if it is possible to measure/predict their current capacity (at node level). Moreover, primary cells are highlighted due to their high energy capacity and relative low cost. However, other energy reservoirs can also be adopted as *backup* modules. For instance, in [15], low self-discharge rechargeable batteries are used for self-powered water/gas metering nodes. Also, in [16], the recent fuel cell technology is proposed with the backup role in a hybrid energy system of a WSN node.

Case Study. During a period of more than 2 years, we adopted a standard WSN solution based on ZigBee technology, solar panels, and rechargeable batteries in two outdoor sites with extreme high and low temperatures [2]. The reliability of this solution was clearly impacted due to the performance of the energy subsystem. Later, we designed a WSN node called Ripple-2A with significant enhancements in terms of network performance. However, this node maintained the traditional design for outdoors WSN nodes (solar panel and rechargeable batteries), but it was implemented by means of different hardware components. Although Ripple-2A has achieved the design goals in terms of efficiency with an overall network overhead smaller than 1%, the weakest part of the solution, in terms of reliability and long-term lifetime, was again the energy subsystem. As already discussed in Section 2.1, the main issues were related to relative small lifetime for the rechargeable batteries (around 1 year), their performance under extreme temperatures, and the fragility of small solar panels available at the market.

The next step was the nontraditional design of a WSN node which could be powered by nonrechargeable batteries. Several months of research were dedicated for the realization of long-term outdoor experiments that could validate this approach. The new node design is called Ripple-2D, leaving room for two additional kinds of nodes: Ripple-2B (solar panel + supercapacitors) and Ripple-2C (solar panel + supercapacitors + non-rechargeable batteries). The latter one is currently under development, and it follows many of the guidelines proposed in this work. The Ripple-2D solved the pulse current effect issue by slow-charging supercapacitors which in turn power the radio module. In short, the current drained from the battery never goes beyond 15 mA, and the pulse effect is voided. As expected, the delay introduced by this charging step can impact the functionality of existing network protocols. Therefore, we also designed a new network solution to address this challenge. Many of the ideas behind this design are part of the framework proposed here. Based on the results from simulated and empirical (accelerated and very-long term experiments) results, the effective capacity of the battery is found to be higher than 90% in relation to their nominal value [9]. The designed lifetime of the Ripple-2D under this project is almost 2 years (conservative value). Currently, the majority of the nodes are close to reach the target lifetime and they are in continuous and reliable operation [4], proving that this solution is indeed significantly superior compared to original Ripple-1 and Ripple-2A architectures based on rechargeable batteries.

These results provided the foundation for the realistic achievement of a very long lifetime for the nodes in conjunction with a high level of reliability. Moreover, the overall network solution also provides a very deterministic way to forecast the lifetime of each individual node [4], as will be discussed in detail in Section 4. At the data server's side, it is also possible to dynamically extend the lifetime of the nodes by means of spatiotemporal activation of a subset of the nodes [2]. In this way, the initial life expectancy of 2 years can still be significantly extended. At node and network sides, recent advances in compressive sensing (CS) can also be added to the solution in order to extend the lifetime of the network without significant sacrifice in terms of data quality, as will be discussed in Section 6. In short, this case study demonstrates that the energy-management efforts can be realized at node, network, and data server levels. The next generation of WSN node for this project, Ripple-2C (currently under development), combines a solar panel with supercapacitors and non-rechargeable batteries, exactly as recommended in this section for energy-constrained scenarios. The ultimate goal is to extend the lifetime for realistic values beyond 5 years in order to properly support the main project behind this effort [2, 3].

3.2. Foundation 2: Distributed System inside a Sensor Node. At the proposed framework, a node can switch between its regular operation and a very low duty-cycle mode. From now on, we will call a WSN node that supports such dual duty-cycle (DDC) mode of operation a *DDC node*. The implications

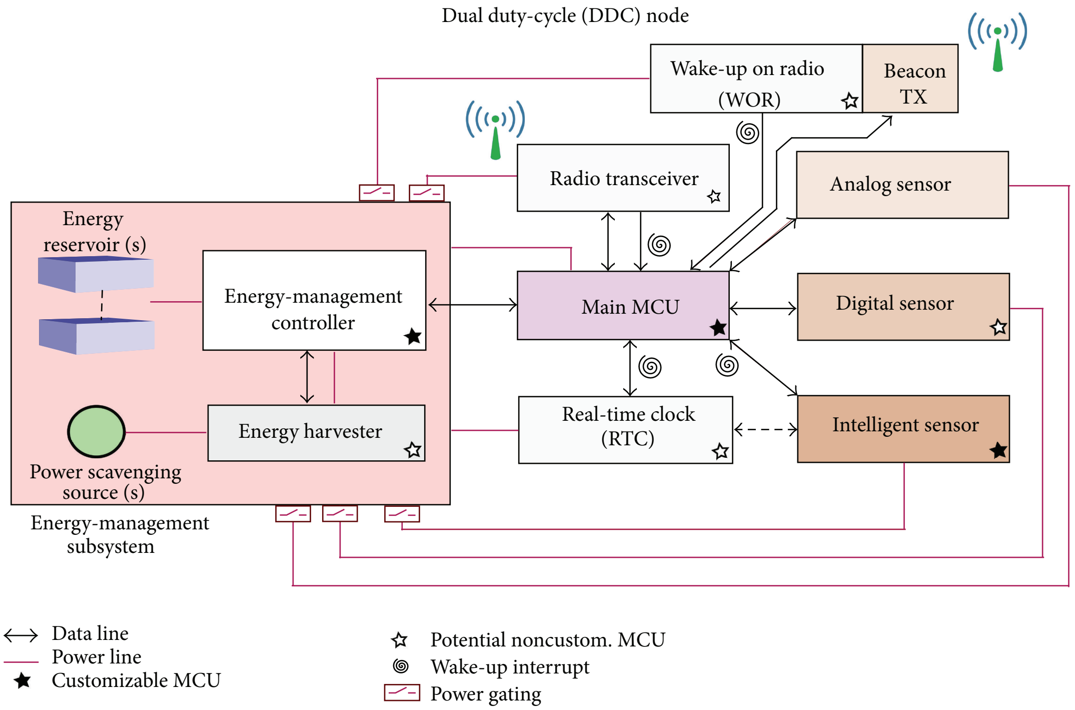


FIGURE 5: The dual duty-cycle (DDC) node has multiple microcontrollers (MCUs) and intelligent devices. That is, it encompasses a distributed system inside itself. The main goal is to achieve energy savings at unprecedented levels compared to traditional nodes. The existence of multiple power lines (lines without arrows in the figure) rather than a single power line is associated with the use of the power gating technique [9] and different voltage conditioning schemes for the internal modules.

associated with the dual mode of operation at the design of a node are definitely not trivial, in particular when energy harvesters are employed. Significant hardware and software additions at the DDC node are required and the reasons why such changes are necessary are better understood when the overall framework is explained. At this section, a brief presentation of the modifications in a DDC node is listed. The fundamental change is related to the evolution from an architecture model centralized in a single microcontroller (MCU) into a solution with multiple MCUs, each one with a distinct role. In other worlds, a DDC node is essentially a distributed system inside a node, as shown in Figure 5.

The following real-word case is used to illustrate one of the justifications for the addition of complexity and costs in a DDC node. Consider a typical scenario where a solar panel is being used to charge a secondary battery. The design goal is to use the maximum amount of the energy stored in the battery. Typically, a DC-DC converter is necessary considering the potential variation of the voltage level on the battery while it is being discharged. While the node is active, the mentioned DC-DC converter can potentially achieve very high levels of efficiency, such as >95%. Therefore, its use is clearly justified considering the mentioned goal. When the node enters sleep mode, the load current drastically drops, as expected, reaching values on the order of μA. However, because the DC-DC converter must be continuously active, the overall power consumption of the node is effectively dominated by the consumption of this converter. Unfortunately, its power consumption can be 2 or 3 orders of magnitude higher than the sleeping consumption of the MCU and adjacent modules because the DC-DC converter typically has very low efficiency when subjected to light loads. Therefore, in particular for low duty-cycle regime, the design of the node can be modified in order to void the voltage converter while the MCU is in sleep mode. Such simple goal adds significant complexity to the design.

Because the proposed framework is founded on a mechanism that drastically changes the operational duty-cycle of the DDC node, the energy subsystem of the node must also be very efficient while operating in sleep or similar mode. Therefore, an increase of complexity is expected in the design of such node. Accordingly, the next design step is to attempt to separate the power lines and avoid the use of voltage regulators in modules that are constantly powered on. Moreover, it is necessary to discover the power needs of the different modules of the DDC node. For instance, the dynamic voltage level of the main MCU may not be the same as the radio module or as the real-time clock (RTC). Such complex scenario is better understood by means of the Figure 5. For example, note that the concept of having a single shared power line for all the modules gives place to multiple and power-controlled lines. While some aspects of this figure are discussed next, additional details related to the software side of a DDC node are given in Section 4. It is important to remember that the recommended guidelines in this work can be partially followed. Therefore, the realization of a full DDC node may not be the goal of a WSN designer considering the characteristics of his application and specific energy aspects. Nonetheless, some of the concepts underlying the framework and its DDC node can be borrowed and integrated on an existing WSN platform, as discussed next.

Autonomous Energy-Management Controller. The control of the power-gating process (activation/deactivation of the internal modules [9]) of the node is performed by this MCU. Moreover, the activation of voltage conditioner(s) and supercapacitors charger(s), the selection of the main energy reservoir, and the power reset of the main MCU (watchdog-timer function) are tasks provided by this module.

Autonomous Energy Harvester. While operating in very low duty-cycles, a sleeping main MCU can potentially miss important energy harvesting events. Therefore, an autonomous energy harvester module can be an ideal solution provided its active power is very small. The energy harvester and energy-management controller roles can be integrated in the same MCU.

Main MCU. this module runs the software associated with the low duty-cycle mode in a DDC node which is mainly implemented as an application-layer overlay. Typically, the legacy platform that is being ported into the DDC node is actually called *Radio Transceiver* module. For instance, consider a DDC node that uses a TelosB module (TinyOS 2.x and IEEE 802.15.4-based communication). In this scenario, it is important to remember that the TelosB is not the Main MCU, but the radio transceiver module of the node, one of the blocks in Figure 5.

Radio Transceiver. It refers to any WSN node that is being ported at the DDC node. In the case of a fully customized DDC node, the radio transceiver can be any radio that provides at least point-to-point communication. As expected, the majority of the existing WSN nodes fall into this category of radio devices. If a primary cell is used, in order to avoid the pulse effect it is recommended that the radio module be powered via supercapacitors [9]. The main drawback of this approach is the significant data latency associated with the time necessary to charge the supercapacitors. In general, when a WSN radio transceiver is directly connected to non-rechargeable batteries, the lifetime of these cells is strongly reduced.

Real-Time Clock (RTC). In low duty-cycle mode, the RTC is used to wake-up the main MCU according to a certain scheduling. The power consumption of the RTC must be significantly smaller compared to the sleeping power consumption of the main MCU. Typically, the RTC device is powered by a non-rechargeable battery in a DDC node.

Wake-Up On Radio (WOR). The WOR is an optional module that allows a DDC to quickly swap between low duty-cycle mode to regular mode. In order to be adopted in a DDC node, the WOR module must consume a very tiny power (e.g., $<5\,\mu$W). In [17], a nanopower WOR is reported making it a potential candidate for WOR in DDC nodes.

Analog Sensor. Typically, this low-cost kind of sensor requires a stabilized power supply. Also, as a passive device, it is not capable to wake up the main in the event of changes at the environment where it is deployed.

Digital Sensor. typically, this kind of sensor has its own non-customizable MCU, an internal voltage regulator, and a serial interface such as I^2C or SPI for communication with the MCU. In general, it is not capable to wake up the main MCU in case of detection of events.

Intelligent Sensor. This is the newest generation of sensors that has the capability to wake-up the MCU based on the analysis of an external event. Although one can customize its own MCU to achieve the above goal, recent available technology goes one step ahead and provides a way to perform continuous monitoring of an event at the cost of few μW. As expected, this scheme allows the significant reduction of the duty-cycle of the main MCU in event-driven applications. Very recently, the manufacturer Atmel announced the Sleep-Walking technology [18] which is basically the integration of the Intelligent Sensor with the main MCU in this context. Although not shown in Figure 5, when a WSN node is ported in a DDC node, the three mentioned kinds of sensors (analog, digital, and intelligent) can be attached at the main MCU, at the radio transceiver (which is the regular WSN node), or at both. These options are associated with the function of these sensors at the framework and also at the main application.

Multiple Power Lines. With this provision, each individual module can be powered on/off independently by means of the power-gating technique [9]. Moreover, only the devices that require voltage conditioning are connected to these special power lines. In fact, there is an underlying effort to avoid, if possible, the use of voltage regulators [9] in any module that is constantly powered, such as the main MCU and the RTC. Nonetheless, the multiple power lines effort must be compared with the simpler solution of putting a module in standby mode, if this function is available. The baseline for power consumption must be the sleeping power consumption of the main MCU. Moreover, it is important to remember that power gates also present a leakage current and not all electronic switches can be used for this role. As a reference for comparisons, the typical power leakage (loss) due to the power-gating is smaller than $0.3\,\mu$W.

Recent technological advances point into the direction of pico- and nano-MCUs embedded in a significant number of electronic devices, many of them considered *analog devices* for decades. This is the case for power scavenging sources, sensors, battery chargers, and even bulb lamps. Therefore, the second foundation of the proposed framework, the adoption of a distributed system inside a node, is actually well aligned with the industry trends. Nonetheless, based on the energy effort tripod concept discussed in Section 2.2, besides the hardware, also the network and the application characteristics must be considered in order to achieve a truly functional, reliable, and long-term maintenance-free WSN solution. So far, this work gave a significant emphasis on the energy subsystem and hardware aspects. From now on, the focus will be mainly oriented toward energy-management software mechanisms. Such software modules assume the existence of

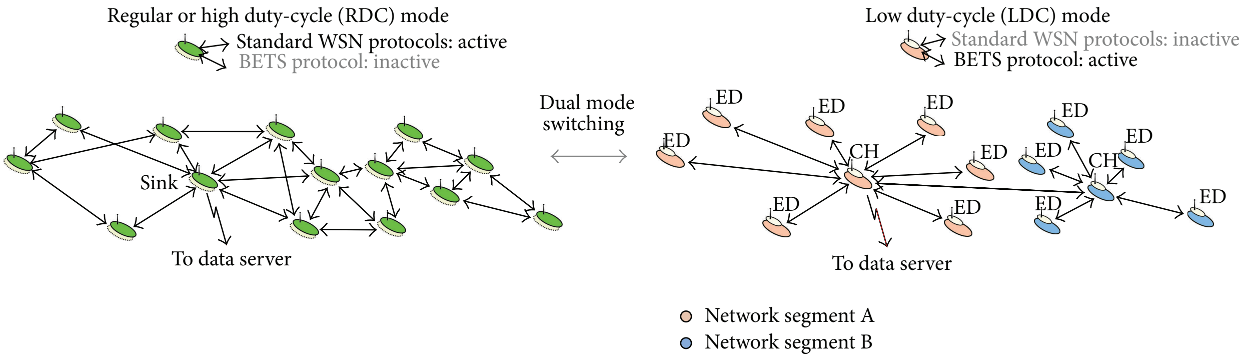


Figure 6: Dual duty-cycle (DDC) operation: the network switches between LDC and RDC modes. In RDC mode, the network maintains its original characteristics. In LDC mode, the BETS protocol becomes active, a planned network segmentation occurs, and the network achieves its maximum energy efficiency.

the hardware for DDC node with the features presented in this section. Important questions associated to the need of having dual duty-cycle modes and how a DDC node actually operates will be answered in the next section.

3.3. Foundation 3: Dual Duty-Cycle (DDC) Operation. A truly energy-balanced and efficient WSN solution subjected to constrained/irregular energy resources depends on coordinated optimization efforts in terms of hardware, network algorithms, and flexible application demands. Such coordination is the main focus of this section, and a discussion on how a DDC system can achieve the mentioned goals is provided. A WSN node that follows the guidelines provided in this section, the so-called DDC node, can be designed entirely from scratch or, alternatively, can be the result of the integration of an existing WSN with additional hardware and software modules. Details of how to implement a DDC node by porting an existing WSN platform are provided later in Section 3.5. Similarly to the hardware guidelines so far provided, one can decide to implement some of the underlying concepts in this section in his WSN design without fully implementing a DDC operational system.

Intuitively, the expression *dual duty-cycle operation* transmits the idea of having a system that operates in low and high (or regular) duty-cycle regimes. Therefore, a natural question that raises is why not designing toward uniquely a low duty-cycle operation, such as <1%, if it is more energy-efficient? The answer is related to two of the legs of the energy-effort tripod concept in Figure 2: the network and application characteristics must be also be considered for a balanced solution. Starting with the application constraints, in some WSN systems a high network throughput is required, such as in a surveillance system involving video and audio traffic. As expected, there are periods of time when the nodes are subjected to a very high duty-cycle operation. However, it is also possible that not all nodes are involved simultaneously in a high data traffic. Moreover, such intensive usage of the network typically occurs in bursts, such as when an event of interest is detected. Therefore, many WSN applications already present some form of dual mode of operation. However, not all WSN solutions optimally *exploit* this fact in order to achieve maximum energy savings.

The above example involving a surveillance system is one of the target scenarios for the proposed framework and the idea is relatively simple, while in regular mode, the existing WSN solution is fully used *as it is*, that is, without significant changes due to the proposed framework. Therefore, it is expected maximum network performance (as provided by the current WSN solution) with potential sacrifice on the energy performance. However, when the application does not require such high network throughput, the DDC nodes can be commanded to switch from regular duty-cycle (RDC) mode to low duty-cycle (LDC) mode. The overall process is illustrated in Figure 6 where three additional aspects of the LDC mode are also shown: the 2-tier architecture, the network segmentation, and the protocol called BETS. These aspects will be discussed later in the next sections.

It is important to highlight that nodes in LDC mode are not necessarily inactive or continuously sleeping. Instead, such nodes are actually following a predefined low duty-cycle maintenance scheduling. A node in LDC mode does not need to wake up multiple times per second as is the usual case in traditional WSN protocols. In LDC mode, the nodes only need to regularly wake up after long and continuous sleeping periods (e.g., 15 minutes) for sending measurements or for just updating their status in a central data server. However, while in the middle of its deep sleeping period (hibernation), a node in LDC mode can be forced to return to its regular or RDC mode of operation. For instance, in an event-driven application, if an event of interest is detected by one of more nodes, the network must quickly resume its maximum performance operation. Note that, the Intelligent Sensor previously discussed is an important component to turn this scenario feasible. Another example is related to a WSN application for commercial buildings where the nodes employ indoors photovoltaic (PV) panels. In this case, the network swaps from LDC to RDC mode as soon one person starts his activities in his office or at the building. When the human presence is no more detected (e.g., during the night), the network returns to its LDC mode in order to save energy. In this way, many of the functionalities of this WSN application are still provided during nights, weekends, and holidays.

For some datacollection monitoring applications, the RDC/LDC switching may not be necessary, and the node can permanently stay in LDC mode as in the previous mentioned case study [2]. This category of WSN applications does not require significant amount of data traffic and is delay-tolerant (DTN) [19, 20]. For instance, a soil moisture monitoring system and many other environmental monitoring applications typically have an application duty-cycle smaller than 1% and only require measurements every 15 minutes or more. Therefore, DDC nodes in permanent LDC mode can potentially satisfy the needs of the application. Nonetheless, one can argue that a traditional WSN solution can also satisfy this application and still be an answer for the mentioned surveillance application. Considering this argument, doubts can raise in relation to the real need of adding complexity and costs to a WSN by implementing DDC nodes. The answer lies at the extreme (and realistic) energy savings achieved when a node operates *in LDC* mode. In fact, the simulated and empirical results in Section 5 showing the network lifetime extension by multiple folds are related to the LDC mode. Therefore, the bigger is the period of time that a node stays in LDC mode, and the higher is the energy efficiency of the solution. While the network is operating in RDC mode, its energy performance is essentially the same as the original WSN platform. The term *essentially* is used here because the energy overhead due to the additional hardware necessary for implementing the DDC node is assumed to be very tiny, and this specific aspect will be discussed later in Section 3.5.

In short, the dual duty-cycle switching mechanism is provided to obtain the best of both worlds (energy and network performances).

(i) In regular (RDC) operation, the WSN node has essentially the same performance as it would have if the framework was not applied, but with the cost of potential not optimum energy efficiency: network performance ⇑, energy performance ⇓.

(ii) In low duty-cycle (LDC) operation, the WSN is drastically reformulated in a process similar to virtually removing all existing nodes and deploying a new network but maintaining the same physical layer (radio transceiver) of the nodes. The LDC mode has excellent energy performance achieved with some level of network performance sacrifice: network performance ⇓, energy performance ⇑.

3.4. Characteristics of the LDC Mode. In this section, it is explained why the energy saving mechanisms of a DDC node in LDC mode hardly can be achieved by simply using off-the-shelf WSN nodes. Nonetheless, the achieved very high energy efficiency has its price in terms of significant penalties on data latency and throughput. Although such drawbacks are actually expected for a node operating in LDC node (otherwise, it would be operating in RDC mode), there are other constraints that can impact the adoption of a DDC system for all WSN scenarios. Accordingly, a discussion on the limitations of the DDC system is provided, such as the lack of support for node mobility. Also, characteristics of potential target applications that permanently operate in LDC mode (LDC-only applications) are also provided.

3.4.1. Motivation. To better understand the energy savings associated with the LDC mode, two discussions are provided, one that introduces the preliminary ideas and other that gives additional quantitative intuition. It is important to highlight that the following discussion is crucial for a comprehensive understanding of the goals behind this work.

In a typical data collection application, the nodes regularly wake up, sense their environment, and transmit the sensing data to a central point. In many cases, the amount of the data is relatively small and the sampling rate is also pretty small resulting in a very low duty-cycle operation. However, this conclusion is exclusively based on the *viewpoint of the application*. Unfortunately, the overhead of the networking protocols can be sufficiently high and dominate the energy consumption of the nodes. On the other hand, if the application's duty cycle is relatively high, the overhead of the network is typically negligible. Therefore, in order to effectively compare two solutions, such as two different network protocols, it is necessary to understand the effective network overhead caused by each one of the evaluated protocols. Besides the typical overhead added by the networking protocol in the form over additional bits or bytes at the *message payload*, it is also necessary to understand the impact of that solution in the timeline. Specifically, it is well known that, even without any application activity, many WSNs sustain some sort of *network infrastructure* traffic to maintain the nodes synchronized, to detect the state of the nodes, and so on. Therefore, even under a very low application duty-cycle, the effective duty-cycle can still be relatively high. Naturally, because the radio is typically the most power-hungry module in a WSN node, the term *effective duty-cycle* is related to the use of the radio transceiver module in Figure 5. The important question to be answered at this point of the discussion is related to the magnitude of such overhead.

In order to review this problem under a quantitative viewpoint, let us consider an antimold WSN-based solution for a commercial building where the sensor nodes are strategically installed inside the walls. Due to the fact that the nodes will be deployed in areas of difficult access and, in many cases, without energy scavenging opportunities, non-rechargeable batteries are required. Due to economical reasons, the exchange of such batteries must only occur after 5 years. Sensing measurements must occur every 20 min, but cycles of up to 60 min are still acceptable. The power profile of the nodes, typical WSN nodes, is shown in Table 2. The sensor node comprises a processor (MCU), a sensing module, and a radio transceiver. It periodically wakes up, performs some processing (1 s), takes measurements (5 s), sends/receives data to/from the base station (3 s), performs more processing (1 s), and finally sleeps again. Assume that the communication performance of the nodes and their reliability are very high and a fixed topology for the nodes is defined. In this scenario, different stack of WSN protocols are tested and any additional measured network overhead is assumed to be exclusively due to the characteristics of that protocols stack. Also, different sampling rates are used. Some protocols require that the nodes be active for more time in comparison with others. Moreover, some protocols are associated with a significant number of redundant data paths between two nodes. All

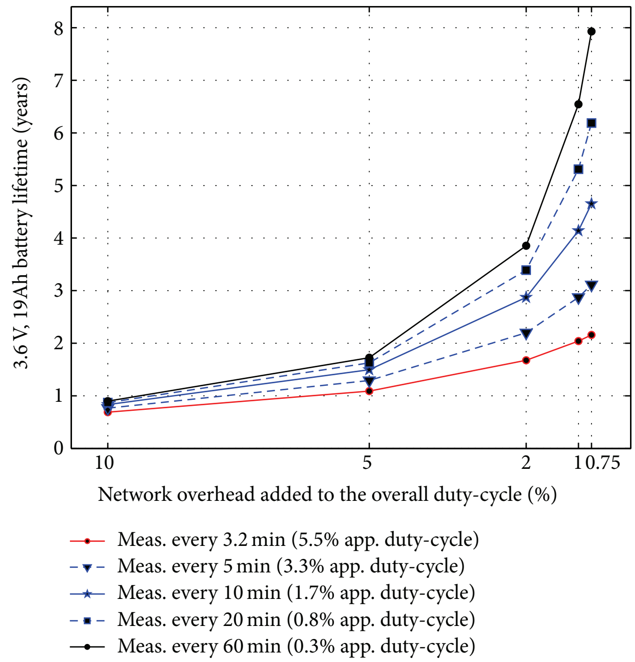


Figure 7: Lifetime of a WSN node (see Table 2) for different network overheads and application duty-cycles.

these differences impact the overhead of these protocols. As a result, after careful measurements, it is found that the effective duty-cycle of sensor node increases by 0.75%, 1%, 2%, 5%, and 10% according to the choice of the protocol stack. Assuming that the initial energy of a node is 245 KJ, the goal is to run simulations that relate the lifetime of the node with the choice of the protocol stack and application duty-cycle. It is assumed that no communication errors occur and that 100% of the initial stored energy is effectively used by the node (ideal hardware) because we want to focus only on the network overhead effects on an ideal scenario.

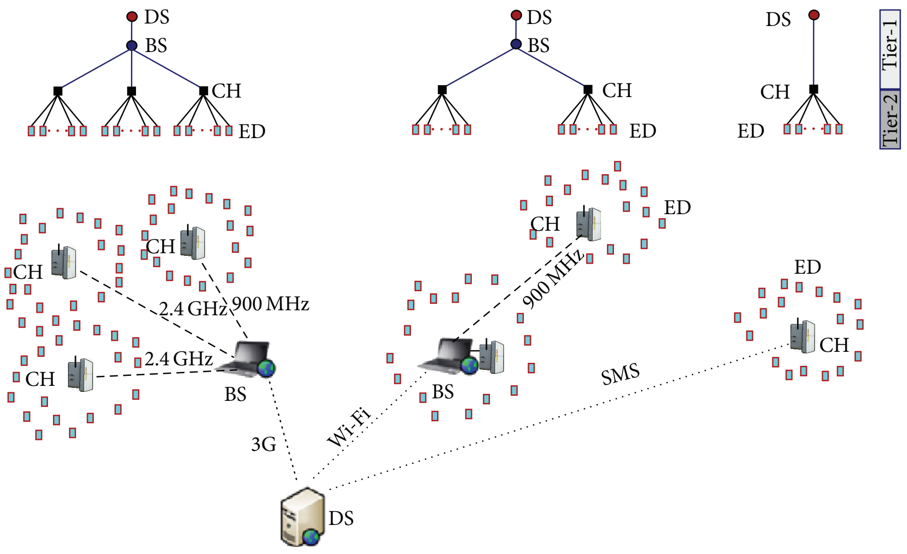
The results of these simulations are shown in Figure 7. As expected, when the network overhead increases, the lifetime of a node decreases. However, such impact is particularly stronger for low duty-cycles applications. For instance, if the network overhead increases from 0.75% to 5%, the life expectancy of a node decreases around 75% and 50% for measurement cycles of 60 and 3.2 min, respectively. But caution is required in this analysis: although the difference between 75% and 50% does not seem to be very drastic, it is not actually the case. These percentages are relative to different lifetime goals and when one translates these values into years, it is found that the lifetime of the nodes decreases by around 6 years when the network overhead increases from 0.75% to 5% and 60 min schedule is used. For the same increase in terms of network overhead, the lifetime of the nodes decreases by 1 year if the 3.2 min schedule is used. The conclusion is clear: a relative slight decrease at the performance of the stack of networking protocol dramatically impacts the lifetime of node for low duty-cycles applications.

Therefore, in our ongoing example, if we opt for 60 min schedule in order to save energy in terms of active power, the choice of the lighter stack of protocols is the only way to make the solution feasible. Moreover, if we want to improve the data quality of the solution by increasing the sampling rate to cycles of 20 min, the only way to achieve the required lifetime >5 years is to have a network overhead not higher than 1%. Unfortunately, the typical network overhead in WSNs is much higher than this value. For instance, these are some reported radio duty-cycles of MAC protocols [21]: LPL 10%, T-MAC 2.5%, S-MAC 23%, and B-MAC 11.4%. Note that the overhead due to other protocols, such as network and transport, are not included in these values, and we still assumed an error-free network. Therefore, the final answer in relation to the illustrative project is that it is potentially unfeasible if we simply adopt traditional WSN technology. In the context of our framework, the goal is the development of solutions for this and many others scenarios, while in LDC mode, the DDC node must experience a very small effective network overhead (e.g., <1%).

As a rule-of-thumb, when a WSN protocol provides a higher level of flexibility and functionality, it is also expected a higher network overhead due to this protocol. Therefore, in the formulation of the operation of a node in LDC mode, it is important to consider specific application needs and *avoid* unnecessary network features. Note that in the previous example an ad hoc deployment, mobility, multi-hopping, and even collaboration are not required features. Considering that the topology is static, it is possible to organize the network into star-based segments and, at the central point of each segment, a special node (i.e., cluster heads) can collect the data from the sensors and transfer data to/from a base station or data server. With such ideas in mind, a 2-tier asynchronous network comes naturally as a feasible architecture, as shown in Figure 8. Observe that, in terms of topology, this is the original and still dominant way to organize computers in corporative networks. Moreover, the majority of the IEEE 802.15.4/ZigBee networks also follow such scheme, sometimes augmented with low-height trees topologies [6, 7, 22]. In short, in order to achieve overall network overheads smaller than 1% (in terms of duty-cycle), the DDC system operating in LDC mode must adopt a very simple and efficient network topology and protocol(s). How this goal is achieved and also the drawbacks of this approach are considered next.

3.4.2. Network Topologies Constraints. The DDC system operating in LDC mode is based on a cross-layer protocol called best-effort time-slot Allocation (BETS) protocol proposed in [4]. It is implemented as an application-level overlay because this is the simplest way to have a DDC node switching between RDC and LDC modes without changing the software layers implemented at the WSN platform that hass been ported. However, if one designs a DDC node from scratch, the main functionalities of the BETS protocol can be potentially implemented at the data link layer. The details of BETS are discussed in Section 4. For now, it is enough to understand its general operation and requirements.

The BETS design is guided by simplicity behind the concept called *selfish node* presented in [4]. A regular sensor node acts *selfishly* in the sense that no message relaying is actually



DS: data server
BS: base station
CH: cluster head (also a sensor node)
ED: end device (regular sensor node)
BETS protocol ED-CH communication

Figure 8: Different topologies for the nodes in LDC mode [4].

performed by the node. As the name implies, this idea comes in the opposite direction in relation to ad hoc networks and many current WSNs with emphasis on *cooperation*. The sensor node, which is called end device (ED) simply wakes up, takes measurements, and sends the data to a specific collecting or cluster head (CH) node. After successfully sending its data to CH, the ED receives an acknowledgment from CH with the precise time for the next cycle and goes to sleep. In order to avoid the need of message relaying among ED nodes, the network is segmented, and each segment has a star-like topology with a predefined maximum number of EDs. The CH can communicate with all EDs in that segment, and its role is similar to the typical *sink* in the WSN literature. The method of the communication between CH and a central point is a completely open aspect in this framework. It can be implemented by means of any wireless technology. It is also possible to create a network of CHs involving long-range links among them and selecting one of them as a base station (BS) which would be in charge of sending the data packets to a data server (DS). Such very flexible architecture is possible because BETS is an asynchronous protocol in relation to the message delivery between ED and DS. Once the ED's data reaches the CH node, the transaction is concluded from the viewpoint of the ED node. When and how CH compacts the ED messages, if this is the case, to send to DS is also an open implementation aspect.

The network architecture supporting BETS (called Ripple-2 in [4]) is also hybrid: any wireless communication technology can be used provided that a simple point-to-point link can be implemented. For instance, instead of using an off-the-shelf WSN node as the radio transceiver for the DDC node (see Figure 5), one can adopt 900 MHz point-to-point radio modules without any networking layer besides the physical layer. Moreover, distinct network segments can adopt different wireless link technologies. Similarly, the technology used for the ED-CH link can be different from the CH-BS, CH-DS, or BS-DS links. Therefore, the BETS protocol is a proper choice for the LDC node: it is very flexible, open to integration, and intuitively it seems to have a very tiny overhead due to its simplicity. Some of the possible topologies for the nodes in LDC mode are shown in Figure 8. Note that the maximum height of the network tree is 2 which highlights the fact that, with time, many enhancements can be easily added to the solution. Based on the analysis of this figure, some limitations or constraints of the proposed solution can also be identified.

Constraint 1: No Native Support for Mobility. Node mobility is not supported by this solution because a fixed and well-planed topology is assumed *a priori* for each network segment. When the ED node wakes up, it simply takes measurements and transmits the data. There is no provision for *network setup* phase or any kind of search for the location of the CH node: the ED node simply sends the data and expects that CH receives and acknowledges that message. Therefore, a mobile node could not fully implement the selfish node behavior which was previously described. Therefore, in a native DDC system with LDC and LDC modes, the mobile nodes can only be part of the portion of the network that is operating in RDC (regular) mode.

Constraint 2: Not All Network Topologies Are Supported. Some physical network topologies can impede the DDC to operate in LDC mode. For instance, consider a line-fashion deployment, such as in a sequence of sensor nodes in a bridge.

In this case, it is very hard to implement logical star topologies unless the communication range of a node encompasses a significant number of nodes at both directions of the line which is usually not the case. Consequently, another protocol must replace BETS in order to include some level of *light* collaboration among EDs in LDC mode.

Constraint 3: CH-DS/BS Links Can Pose Challenges. Typically, all WSN nodes in a network use the same wireless technology, such as IEEE 802.15.4 PHY (e.g., TelosB-based WSN). When the network (or portion of it) switches from RDC to LDC mode, the *new* formed network obeys a predefined scheme with one or more star-based segments. The assigned CH nodes now need to send the data to DS by means of the same low-power, short-range wireless links because their radio transceiver is still the same. Therefore, the collaboration among CHs is potentially necessary in this scenario. The proposed framework does not offer guidelines for the CH/BS or CH/DS communication. Nonetheless, one potential solution in this case is the adoption of two radio modules at CH nodes. In many scenarios, the second radio (CH-BS/DS link) can be a Wi-Fi adapter if the place also has a WLAN infrastructure. Again, in terms of energy efficiency, it is necessary to investigate the energy costs of this approach compared to traditional WSN solutions.

3.4.3. Target Applications for LDC-Only Mode. The energy efficiency of the LDC mode is achieved with the cost of network performance penalties in addition to some topology constraints. In fact, we must have in mind that the full adoption of the proposed framework is realistically not possible in many cases. On the other hand, there are scenarios where the RDC/LDC switching is not even necessary, and the network is always in LDC mode. This is the case when the WSN application is a low duty-cycle data-collection, it is delay-tolerant, and it does not have to support mobile nodes.

In summary, three cases in relation to the application of the DDC system proposed by this framework can occur: (a) for many regular WSN applications, the DDC system must operate in dual mode (RDC/LDC), (b) for many low duty-cycle data-collection applications, the DDC system only needs to operate in LDC mode, and (c) for some applications, the DDC system cannot be employed but some guidelines of this framework can still be adopted to enhance the energy performance of the solution.

Because the LDC operational mode is closely associated with very small network overhead, the goal of maintaining the DDC system uniquely in LDC mode is actually the best option in terms of energy-management optimization. However, just because the main application is low duty-cycle, it does not imply that dual mode operation can be removed. There are some requirements that prevent the adoption of a LDC-only mode and the RDC/LDC switching must be preserved. One requirement is related to the characteristic of the application in a LDC-only network: it must follow schedules, such as of a data-collection one. Also, the application must be delay-tolerant and the network cannot have issues in order to follow a 2-tier and non-collaborative architecture. A summary of the characteristics of LDC-only applications is listed in Table 3. A detailed discussion is provided next.

Data latency in LDC-only mode can be a problem for some applications. The complete message data latency of an ED↣DS message can be on the order of seconds or even more mainly due to the asynchronous behavior of the network (LDC mode). The latency for the other way (DS↣ED) is even worse and can be in the order of minutes. Fortunately, typically for a data-collection application, the ED↣DS message direction is the one of interest. Even in this case, the reasons for a significant data delay are as follows.

(i) Once a measurement-cycle starts, the ED node must wait for its time to transmit the data.

(ii) Once the data from this ED node arrives at CH, it is necessary to wait until CH concludes the data collection from all nodes at that cycle in order to have the transmission to DS.

(iii) At the end of each cycle, CH must activate its CH/BS or CH/DS link to perform the data transfer. Such activation delay can be significant. For instance, to activate a SMS based (text), the SMS modem can take almost 1 min to conclude the link activation and message transmission.

(iv) If multiple CHs send data to a shared device (BS), the delay due to the coordination among CHs and also due to the activation of the BS/DS link can also be significant.

A second aspect to be evaluated before deciding by the LDC-only network operation is related to the energy requirements for the CH nodes. The energy consumption of CH follows, at the best theoretical case, a linear relation with the number of EDs in that network segment. For instance, if a CH is in charge of 20 nodes, its lifetime is expected to be around 1/20 the lifetime of a regular (ED) node, assuming that all nodes start with the same initial energy capacity. Therefore, the lifetime of CHs can be strongly impacted if the WSN design does not carefully consider the energy challenges in CH nodes. The increase of the number of segments (thus, also the number of CH nodes) can be a technique that alleviates the energy workload on each assigned CH node. The dynamic sharing of the CH role among nodes in the segment, such as in a round-robin fashion, can also be a solution. Finally, the adoption of a hybrid power source system for CHs based on energy scavenging and non-rechargeable batteries can extend the lifetime of CHs. The overall characteristics of a DDC system operating in LDC mode are summarized in Figure 9. Note that two of the highlighted challenges in this design approach are the need of some form of power hibernation for both ED and CH nodes and also an efficient time synchronization technique for the ED/CH communication. Both aspects are considered in our previous works: the former one, involving hardware techniques, is considered in detail in [9], and the latter one is presented in a short-format in [4]. The detailed discussion of the BETS protocol, the core component of a DDC system, is discussed in detail in Section 4, where its algorithms are presented. The performance evaluation of BETS is provided in Section 5.

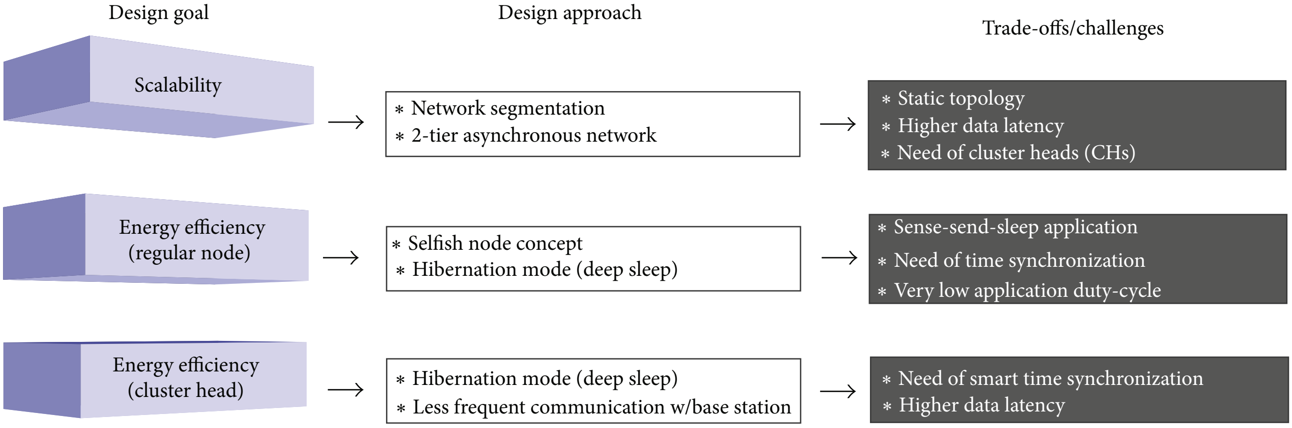

Design goal
Design approach
Trade-offs/challenges
Scalability
∗ Network segmentation
∗ 2-tier asynchronous network
∗ Static topology
∗ Higher data latency
∗ Need of cluster heads (CHs)
Energy efficiency (regular node)
∗ Selfish node concept
∗ Hibernation mode (deep sleep)
∗ Sense-send-sleep application
∗ Need of time synchronization
∗ Very low application duty-cycle
Energy efficiency (cluster head)
∗ Hibernation mode (deep sleep)
∗ Less frequent communication w/base station
∗ Need of smart time synchronization
∗ Higher data latency

Figure 9: The characteristics of the network while in LDC mode: the BETS protocol is designed to provide high energy efficiency for both ED and CH nodes.

3.5. DDC System: Implementation Guidelines. The main goal of this section is to clarify what are the steps necessary for the partial or full implementation of the proposed energy-management framework. Also, it is a good place to summarize the exposed concepts and acronyms specifically introduced in this work, as shown in Table 1. In relation to the full implementation of the framework, there are at least 3 important scenarios to be analyzed.

Case 1: Node in LDC-Only Mode and Using Ordinary Radios. A low duty-cycle data collection application is being used. This application is delay-tolerant and does not have mobility support. In this case, the DDC system never switches to RDC mode and this is the simplest scenario to implement. The ordinary radios provide a simple point-to-point communication link and, in general, there is no need of further configurations for these radio modules.

Case 2: Node in LDC-Only Mode and Using Legacy WSN Nodes. A low duty-cycle data collection application is being used, and the system never switches to RDC mode. It is necessary to configure the legacy WSN node to provide a simple point-to-point communication with the minimum possible networking functionalities. Any feature or service beyond the physical and medium access control (MAC) layers is unnecessary and, even worse, can potentially increase the network overhead of the solution. Therefore, such features must be permanently deactivated. For instance, a target legacy WSN node can have hidden dynamic topology control and time synchronization features that are not necessary for this scenario and must be deactivated. Good ways to discover if this is the case are (a) to observe how quickly two WSN modules exchange messages just after they are initialized and (b) to use a radio frequency (RF) *sniffer* device to monitor the content of the packets. For the latter case, low-cost monitoring tools for the 2.4 GHz ISM band are available, such as the device in [23] which is used in our ongoing project. If it is not possible to completely deactivate unnecessary features in legacy WSN nodes, maybe the better option for this specific scenario is to exchange the legacy nodes to ordinary radio modules.

Case 3: Node in Dual Mode and Using Legacy WSN Nodes. This is the general case where the WSN application is more strict in relation to network QoS metrics. While in LDC mode, it is necessary to configure the legacy WSN node to provide a simple point-to-point communication with the minimum networking functionalities, as discussed above. However, when the node returns to RDC mode, such networking features at the legacy WSN nodes must be activated again. Therefore, the integration effort in this scenario involves the addition of an application-layer module at the legacy WSN module to dynamically activate/deactivate high-level networking functionalities of the device. If the unnecessary features in legacy WSN nodes cannot be deactivated while the node is operating in LDC mode, the optimum energy efficiency provided by the BETS protocol cannot be achieved.

A typical example of a device with the role of DDC's radio transceiver (Figure 5) under Case 1 is XBee-Pro 802.15.4 Series 1 (Digi Inc.). For Cases 2 and 3, the open-source TelosB mote and the commercial XBee-PRO ZB ZigBee (Digi Inc.) are good examples considering their popularity, documentation, market availability, and ease of integration. Note that all these devices operate at 2.4 GHz band and are IEEE 802.15.4 PHY-compliant, which is the general trend for current WSNs.

The following are implementation guidelines, labeled as *Gx*, which can be applied in all 3 mentioned cases, unless explicitly indicated.

(*G1*) To implement a DDC node, as illustrated in Figure 5, it is necessary to have the main MCU separated from the radio transceiver in all 3 mentioned cases. The BETS protocol implementation software resides at the main MCU.

(*G2*) When an energy scavenger is employed, the Energy Management Controller is separated from the main MCU. In this case, it runs the software module associated with local energy-management decisions. One of these decisions is the selection of the energy reservoir used to power the Radio Transceiver or any other power-hungry module. Moreover, the charging of

Table 1: Main terms and acronyms introduced in the context of this work.

Term	Explanation
Energy effort tripod	Fundamental framework's concept: an energy-balanced WSN solution involves hardware, network, and application.
Energy control loop	Fundamental framework's concept: the operation of a WSN node must be regulated by its energy state.
Selfish node concept	Ultralight energy consumption of a node: wake up, sense, send, and sleep. The expected network overhead is <1%.
Dual duty-cycle (DDC)	2 modes of WSN operation: (a) current WSN solution (RDC) and (b) constrained but highly energy-efficient mode (LDC).
Regular duty-cycle (RDC)	Provides the same network performance as the existing solution with potential energy penalties.
Low duty-cycle (LDC)	Maintenance mode with excellent energy performance. It can also be used for low duty-cycle data-collection applications.
DDC switching	Framework provision which allows that the application's needs (or an event) automatically switch the DDC mode.
DDC system	Part of the proposed energy-management framework which involves hardware and software additions to the WSN.
DDC node	Distributed system (multiple MCUs) inside a single device. The main MCU runs the BETS protocol.
Power gating	Technique extensively used in a DDC node which activates/deactivates the hardware modules of a node (not sleep mode).
BETS protocol	Best-effort time-slot allocation protocol is the core DDC module that controls the behavior of the network in LDC mode.
Energy reservoir	In a DDC system, the energy reservoirs are typically supercapacitors, primary batteries, or a combination of them.
Wake-up on radio (WOR)	Micro- or nanopower technology that allows the DDC node to switch to RDC mode when an event of interest occurs.
Beacon TX	Radio transmitter used to trigger remote WORs. Typically, it is separated from the regular radio transceiver module.
Intelligent sensor	It is not the WSN node but refers to a sensor probe which has the capability to wake up the main MCU via interrupt.
End device (ED)	The regular sensor node in the BETS architecture while in operating in LDC mode.
Cluster head (CH)	The data-collector node in the BETS architecture while in LDC mode. ED nodes only communicate with a CH node.

Table 2: Power profile used in the simulations.

	MCU	Sensors	Radio
Active	5 mW	30 mW (5 s)	70 mW (3 s)
Sleeping	0.01 mW	0.1 mW	0.1 mW

Table 3: Target applications for LDC-only mode.

Design space	Option
Goal	Sense only
Time	Periodic data collection
Deployment	Planned a priori
Topology	Static
Data Rate	Low
Delay tolerance	Must support

supercapacitors is directly controlled by this module, as well as the power gating of many modules, such as a sensing probe. When an energy-harvester is not used, the functions of the energy-management controller can be absorbed by the Main MCU.

(*G3*) For Case 3, the main MCU is sleeping, while the node is in RDC mode and the radio transceiver (i.e., a legacy WSN node) is active. However, sometimes, it is necessary to wake up the main MCU. It occurs because in order to power on/off any module, such as a digital sensor connected to the legacy WSN node, there is an energy-management hierarchy to be followed: radio transceiver ⇒ main MCU ⇒ energy-management controller ⇒ power-gating device. In this case, the legacy WSN node can wake up the main MCU by means of an interruption line, as shown in Figure 5.

(*G4*) In general, for Cases 1 and 2, the sensor devices are physically connected to the Main MCU. For Case 3, they can be either connected to the Main MCU (recommended) or to the Radio Transceiver.

(*G5*) The *power hibernation* refers to a continuous and long period of sleeping time defined in terms of minutes or hours [9]. While hibernating, it is possible to shut down the majority of the modules of the node which leads to significant energy savings. The drawback of this approach is the resulting delay to have the node ready to resume its tasks.

(*G6*) The operational schedule of the node in LDC mode is defined by a centralized energy-management module running at the data server and this information is sent to the nodes via BETS protocol. For Cases 1 and 2, that is, LDC-only mode, such scheduling information is typically defined in terms of minutes, such as cycles of 15 or 20 minutes for some environmental applications [3]. Not all nodes at the segment need to follow the same schedule. In fact, heterogeneous scheduling is one of the provisions of a central data server based energy-management to evenly balance the energy resources in the network. For Case 3, the schedule of the nodes in LDC mode is potentially defined by the radio transceiver, which is the legacy WSN node. In this case, the value of the LDR cycle is mainly associated with the expected network QoS metrics and the characteristics of the application.

(*G7*) The wake-up on radio (WOR) module is connected to the main MCU. Its function is to wake up the node when it is hibernating in LDC mode and an events of interest is detected. Without such provision, a node in LDC mode could not quickly resume to the RDC mode in case of a critical event. For instance, consider a surveillance system under Case 3 and assume that the nodes are hibernating while in LDC mode. One intelligent sensor attached to a specific sensor node detects the presence of an intruder and it promptly awakes the main MCU of the node where it is installed. However, the major challenge is to wake up all EDs of the same segment and eventually all the network in order to switch back to RDC mode and provide the expected functionalities of the surveillance application. Therefore, the key-answer to address this challenge is the WOR module: once the first node is awaken by its intelligent sensor, it transmits a special *beacon* by means of the *beacon TX* module shown in Figure 5. This beacon triggers the WOR module of the CH node of that segment. In turn, the CH wakes up all EDs of the segment using the same technique. An ultra-low-power WOR is a recent and sophisticated technology [16], and its use is recommended for event-driven scenarios.

(*G8*) For Cases 2 and 3, one preliminary and critical test to be performed is related to the feasibility of using the legacy WSN node as the radio transceiver module of the DDC node. It is necessary to evaluate a point-to-point communication with two nodes by means of their *serial ports* because this is the typical way that the main MCU communicates with the radio transceiver. For instance, for TelosB, there are the pins UART0RX and UART0TX at the TelosB's expansion connector that allows such test. Similarly, the XBee modules provide the serial ports TXD and RXD for the same purpose. For the integration of any WSN node that supports pure ad hoc communication (i.e., without involving any form of network hierarchy), the process is relatively straightforward. However, legacy WSN nodes that are natively based on infrastructured topologies, such as XBee-PRO ZB ZigBee, require special attention. For this specific example, a node with the CH role in LDC mode can have the ZigBee Coordinator (or Router) profile defined for itself, and the nodes with the ED role can have the ZigBee end device profile.

(*G9*) The BETS protocol assumes that the network segmentation is already in place when it is in operation. It means that for Case 3, an application-layer software running at the nodes in RDC mode must logically configure the network segment(s). One ED node can only be logically attached to a single segment, and each ED node has assigned a logical address which is used by BETS to identify each ED. In this case, two nodes can have the same logical address provided they belong to distinct network segments. However, if a node of segment A is also in the communication range of the CH at the segment B, potential errors can occur. The simplest solution is to assign a distinct RF channel for each segment, in particular if they are physically close to each other. However, if the frequency-hopping spread spectrum (FHSS) technology is being employed by the nodes, further investigation is necessary. Moreover, for Case 3, it is possible that the network segment of the node in RDC mode does not correspond to the BETS segment (LDC mode), as in the case shown in Figure 6. That is, an existing network hierarchy in RDC mode may not be the same CH-ED hierarchy in LDC mode. It is not the case for legacy WSN nodes that have ad hoc operation. However, this problem can potentially occur when ZigBee-based nodes are used. In this case, the switching between operational duty-cycle modes must be preceded by a dynamic setup of the WSN modules. For the mentioned ZigBee example, potentially the BETS segmentation will be achieved by configuring channel frequency, node profile, PAN id, or a mix of them.

4. Cross-Layer Protocol for Very Low Duty-Cycle (LDC) Mode

From now on, this work assumes that the DDC system is operating in LDC mode. Therefore, the network has already been logically segmented in multiple sets, each one with a CH node and its associated ED nodes. In this section, we focus our attention to one of these network segments and discuss how exceptional energy-performance is possible with a DDC node running the best-effort time slot Allocation (BETS) protocol in LDC mode. BETS is an example of a cross-layer protocol that is compliant with the goals of the proposed framework. However, one can design a similar cross-layer protocol that can also be efficiently used by the nodes in LDC

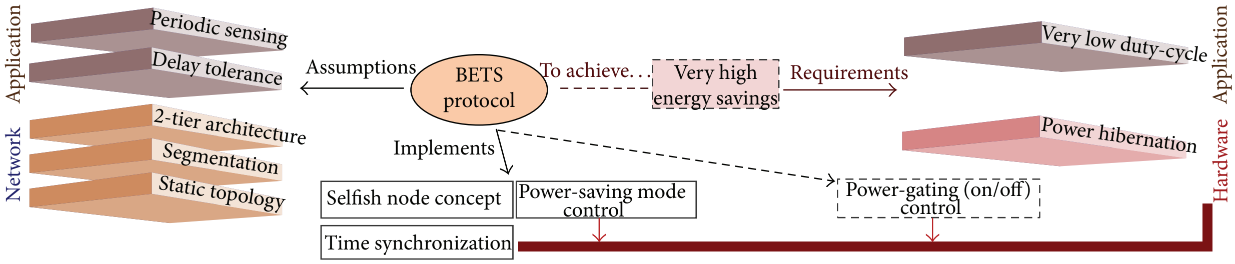


Figure 10: The cross-layer nature of the BETS protocol: a candidate of choice for the LDC mode.

mode. The section starts with an overview of related work in this context, followed by an overview of BETS and its own terminology. Next, the design goals of BETS are presented, and the functional details are discussed by means of its algorithms.

4.1. Related Work. The virtual elimination of collaboration among wireless nodes is not a novelty. The IEEE 802.15.4 standard [6] was introduced as a low-rate, short-range communication solution for Wireless Personal Area Networks (WPANs). One of the network topologies defined in this standard is a star topology where a PAN coordinator is in charge of the communication with the remaining devices. Similarly, the Bluetooth technology is based on a star topology with a master node as the central point [24]. Note that such arrangements are similar to the relation CH-EDs in the BETS solution. In fact, the design of the network architecture associated with BETS, Ripple-2, was influenced by these standards and their outstanding success.

IEEE 802.15.4 only defines the specifications in relation to the physical and MAC layers. However, upper ISO/OSI layers can be optionally used to allow ad hoc deployments, multihopping, trees, and mesh networks. One example of such augmentation is the ZigBee standard [7] which defines the network, application, and security layers. Initially adopted as a WPAN solution, the functionalities of ZigBee become pretty similar to the ones in traditional WSNs. As expected, many WSN deployments based on ZigBee devices have been reported [2, 22, 24, 25].

In this context, BETS can be seen as an effort to add scalability and extreme energy efficiency to IEEE 802.15.4 (star topology mode) solutions without incurring in the higher overhead and complexity of the ZigBee standard. Also, the overhead similar to the one associated with the PAN association procedure [6] in the 802.15.4 standard does not exist in the BETS solution. Moreover, BETS is also designed to support any underlying point-to-point physical layer, not only 802.15.4 PHY. In fact, the ED-CH link implementation is not even limited to a radio operation. Finally, different from a PAN coordinator (or ZigBee router, or ZigBee coordinator), the CH node in the BETS solution can *sleep* and, even better, it can hibernate. This fact drastically reduces the energy requirements of the data-collector device which still is a challenge for typical 802.15.4-based solutions.

A simplified version of WSN based on multiple stars is presented in [26] for forest monitoring, but the details related to the underlying networking protocols are not provided. A hierarchical architecture for delay-tolerant networks is presented in [27], and a customized MAC protocol called LiteTDMA is employed. Hardware specialization of WSN nodes, in particular with the introduction of the power-gating technique, is proposed in [28] and it is extended in [9]. Slot-based MAC implementations have been proposed, such as TRAMA [29], PMAC [30], Z-MAC [31], and H-MAC [32]. Although BETS is not a MAC protocol, its core functionalities in relation to the time-synchronization among nodes of the same network segment have some similarities with the mentioned MAC protocols.

To the best of our knowledge, this work (as an extension to [4]) is the first to propose a non-collaborative model for WSNs by means of the implementation of the selfish node concept presented in Section 3.4. As already discussed in the previous sections, such low-energy model is exclusively adopted while the DDC node is in LDC mode. Even for some applications, the nodes can permanently operate in such LDC mode (where BETS lies).

4.2. Protocol Overview. BETS is a novel cross-layer protocol implemented as an application-level overlay. It is designed for low data rate, low duty-cycle, and sense-and-send WSNs. When the DDC node operates in LDC mode, BETS is the protocol of choice. BETS operates at the MAC and upper-level networking layers. If an existing MAC protocol already exists in the target sensor platform, the MAC functionalities of that platform can be disabled or simply ignored if not causing significant overhead, as discussed in Section 3.5. In other words, the ultimate actions necessary to achieve a fair, contention-free, and reliable communication channel are taken by BETS. As shown in Figure 10, BETS assumes that a periodic sensing application is in place which matches the way the network behaves while in LDC mode. The protocol has a provision to capture the scheduling data sent by the main application (data server) to the nodes and defines the proper allocation of the wireless channel in the time domain. In this sense, the term *schedule* refers to the same object for both application and network discussions.

Also shown in Figure 10, the energy efficiency of the protocol is mainly achieved by sacrificing the network performance in terms of data latency. Therefore, the main application must afford such higher delay which can vary from seconds to hours according to the final implementation. No routing-related functions are actually provided by BETS, and

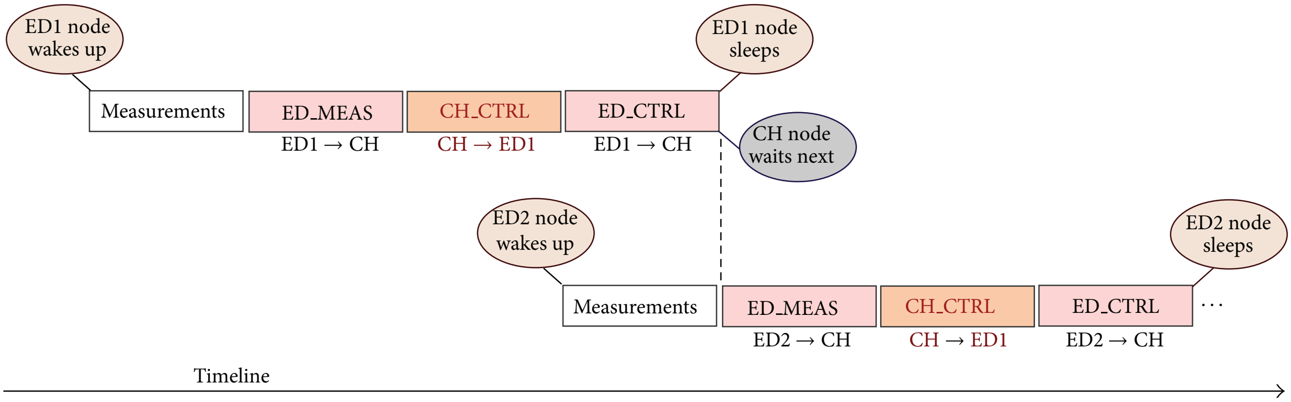


Figure 11: BETS functionality (ED side): an implementation of the *selfish node* concept.

it is assumed that the network is divided into multiple star-like segments, following an asynchronous 2-tier architecture approach. Another assumption shown in Figure 10, static topology, calls our attention to the fact that BETS protocol cannot be easily modified to support mobile nodes. There are optional assumptions to be considered in the BETS context. As shown at the right side of Figure 10, if the ultimate design goal is to achieve very high energy efficiency (i.e., more than 1 order of magnitude compared to state-of-the-art solutions), the combination of very low application duty-cycles (i.e, <1%) and power hibernation techniques [9, 28] is a required step. As already discussed, a DDC node has such capabilities.

The adoption of star-like segments potentially reduces the complexity of a network design. However, it is important to verify if the center point (access-point, controller, or cluster-head) does not become the real bottleneck of the solution. For instance, the energy issues related to the CH node in WSNs have been studied for a long time, and a CH role-rotation scheme has been proposed [13]. In the architectural context where BETS is implemented, such role rotation is not easy to be implemented. Therefore, in order to achieve energy efficiency for both ED and CH nodes, a nontrivial solution is required and this is the main challenge of the BETS design. In fact, besides the role rotation, we did not find in the literature an approach to efficiently reduce energy consumption of CH-like nodes in a static topology. In BETS, we have the goal of having the energy profile of the CH node linearly following the average energy profiles of the ED nodes in the segment. Note that, besides a higher energy capacity (if CH role rotation is not adopted) and strategic communication coverage (which depends on the location of the node), the CH role does not require special hardware/processing capabilities, and any node can potentially have this role assigned to itself.

From the ED's viewpoint, the fundamentals goals of BETS are the implementation of the *selfish node* concept and an efficient way to avoid that two or more EDs try to use the communication channel simultaneously. Because the current MAC protocols do not fully implement the selfish node concept, BETS must have MAC-related functions in order to fulfill the mentioned goals. The messages exchanged by ED and CH nodes are shown in Figure 11. From the viewpoint of the selfish node, the process occurs as follows.

(1) ED node wakes up and takes measurements.

(2) Without any channel negotiation, ED sends the ED_MEAS message to the CH node (unicast). This message basically contains the measurements and few control data, such as its energy state and communication metrics. In some scenarios, instead of individual sensing measurements, the ED_MEAS message can contain aspects related to the status of the node, compressed data derived from a historical sequence of measurements, and so forth.

(3) Without any channel negotiation, CH sends back a CH_CTRL message to the ED node (channel broadcast, logical unicast). This message contains the schedule for the next cycle related to that node.

(4) Without any channel negotiation, ED sends a ED_CTRL message to the CH node (unicast). This message acknowledges the reception of the schedule.

(5) ED node configures its wake-up circuitry accordingly to the received schedule and sleeps.

As expected, no collaboration among nodes exists, and the implementation of the protocol becomes significantly simple at the ED side. In fact, such simplicity clearly indicates the proper realization of the selfish node concept. One can argue that even the CH_CTRL and ED_CTRL messages can be eliminated for a full implementation of a selfish node concept. However, the reliability of BETS requires that some sort of minimum communication-quality control exists, as will be explained later in this section. In fact, the small overhead associated with these messages becomes irrelevant in low duty-cycle applications.

From the CH's viewpoint, the implementation of the BETS protocol is not so straightforward as in the ED case. Besides the proper support for EDs, it is important that CH sleeps in optimum cycles. Different schedules for the nodes of the same segment can potentially cause energy inefficiencies for the CH node. Even a global schedule may not provide

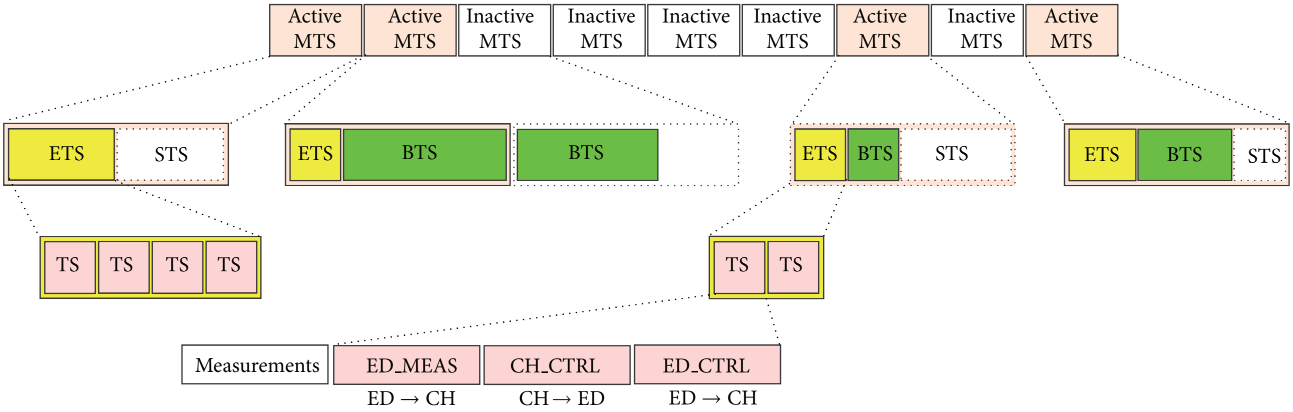


Figure 12: An example of regular (no errors) operation of BETS from the CH's perspective. At inactive MTSs, all nodes (EDs and CH) are sleeping. However, the CH node can still use an inactive MTS for the CH-BS (or CH-data server) communication.

the best energy performance for the CH node. For instance, assuming only 5 min application cycles, one ED can follow the sequence 0–5–10–15. . . in the line of time, while another ED that was initialized later follows the 3–8–13–18. . . sequence. In this case, both EDs have the same schedule, but the CH node cannot take a longer sleep although such goal is clearly possible to be achieved in this case. Therefore, BETS must provide a way to accommodate schedules in the best efficient way possible, both for ED and for CH nodes. In fact, the first reason for the term *Best-Effort* in the BETS acronym is related to this aspect. To solve the mentioned issue, which hereafter is called *dispersion*, BETS adjusts the first cycle of the second node which was turned on at moment $t = 3$. Therefore, this node will have this wake-up sequence under BETS: 3–5–10–15. . . Note that the first programmed cycle, and only this one, is adjusted in order to group all EDs with the same schedule. It is worth to highlight that such adjustment is fundamental for saving energy at the CH side, while such mechanism does not affect the EDs. In other words, BETS is a dispersion-free protocol.

The second reason for the *Best-Effort* term is related to the reliability of the solution. In order to achieve very high end-to-end reliability metrics (in this case, for the ED-CH link), multiple handshake messages may be necessary. However, for every active node, BETS provides a time-slot with a small and fixed length. Bigger and/or dynamic slot lengths can be adopted to increase the communication reliability, but there are energy penalties to be considered. In our simulations to be shown in Section 5, different communication channel error rates are analyzed under BETS in order to evaluate energy and reliability metrics. Also, in one of our current BETS implementations, the average ED-CH distance is around 210 m and for that case, the total data loss was smaller than 1.8% during multiple weeks. The slot length was parametrized to be large enough to allow just a single additional round of ED_MEAS – CH_CTRL – ED_CTRL messages if necessary. Doing so, the solution became more reliable but less energy-efficient, and clearly the data latency of the solution increases. The bottom line in this discussion is that the provision of large time-slots (multiple messages in sequence) in order to increase the reliability may not be really necessary. Nonetheless, due to the BETS flexibility, the slot length can be modified.

4.3. Definitions and Terminology. Before proceeding with a detailed explanation of BETS, some terms shown in Figure 12 must be properly introduced or better defined. In addition, some contextual aspects are discussed in order to ease the adoption of BETS for the LDC mode of DDC systems.

Logical Network Segment (or Simply, Segment). A fundamental assumption for BETS is that the network is divided into logical clusters or segments. This division is realized considering the physical topology of the network. Accordingly, it is expected that the CH node be located at the center of a virtual circle where all nodes inside that circle are able to communicate with that CH (unit disk graph approach). Realistically, the communication range of the ED nodes will significantly vary due to many reasons. Moreover, the location of the nodes must be primarily governed by the application needs. Therefore, it is very hard to achieve an ideal division of the network into circles that do not overlap. Due to this fact, additional communication techniques must be employed to enforce that a node solely communicates with a single CH node even if more than one CH can be reached. For instance, by using different channels/frequencies or even by using software filters, it is possible to deal with the overlapping circles issue. This enforced concept of segmentation divides the overall network into logical network segments: ED nodes of the one (logical) network segment can only communicate with the CH node of that segment and vice versa.

CH's Children. All ED nodes of the same network segment associated with a certain CH node are children of that node.

Registered Children. When CH communicates with DS, the latter can potentially send explicit information about the number of children EDs and their respective schedules. In this case, the ED nodes are considered *registered* children, and the CH can properly calculate how much time to spend waiting for the contact of an ED node based on the number of registered children. Such information is easily available in planned deployments with a static topology and can also be modified to reflect possible node failures. The number of registered children is not required for the functionality of BETS, but it increases the energy efficiency of the CH node because it can hibernate as soon as possible.

Major Time Slot (MTS). Once CH is initialized (boot), the line of time is divided into fixed periods of time called MTSs, each one with the length *mtsLength*, a software variable expressed in units of seconds. In our implementation and also in the simulations, the value for mtsLength is 300 s (5 min). In order to achieve a collision-free solution and maximum energy efficiency, it is assumed that the application schedules are also given in mtsLength units.

Active and Inactive MTS. The CH node does not have to be necessarily active all the time, that is, for every MTS. As expected, during some MTSs, the CH node is sleeping because all its children are also sleeping, and such MTSs are called *inactive MTSs*. When CH is ready to hear an ED node, the respective MTS is called an *active MTS*. The active MTS (AM) is divided into three sequential parts with variable lengths: ETS, BTS, and STS, as explained next.

ED Time Slot (ETS). It refers to the initial part of an active MTS (AM) which is used specifically for communication with children EDs. The dynamic length of ETS corresponds at least to the sum of the assigned time-slots for the active children at that MTS. One of the BETS algorithms uses the *registered children* parameter in order to determine the optimum length of ETS for each AM. Its ultimate goal is to allow CH to sleep as soon as possible.

BS Time Slot (BTS). It refers to the second part of an active MTS which is used for the CH-BS (also CH-DS) communication. The length of a BTS varies as a function of the amount of data to be sent to the BS node, CH-BS link throughput, and possible errors at this link. To save energy, ED data from distinct active MTSs can be aggregated. Doing so, the BTS length is 0 for the majority of MTSs and it is maximum for few MTSs. Although the CH-BS data transfer can be divided into multiple MTSs, a BTS transaction cannot conflict with an ETS of an active MTS. However, by also using inactive MTS for CH-BS/DS communication, the BTS transaction can last longer, as shown in Figure 12. Finally, it is possible to change the CH-BS transfer scheduling according to the energy performance metrics at the CH node. Due to the asynchronous nature of the second network layer (CH-BS), the communication with the BS must not impact the BETS performance for the ED nodes.

Sleeping Time Slot (STS). It refers to the third part of an active MTS which is actually not being used. During this period of time, the CH is inactive and potentially sleeping.

Time Slot (TS). This the time slot allocated to each individual ED. The TS has a fixed and unique length *tsLength* for each segment, a software variable expressed in units of seconds. Such parameter basically corresponds to the time necessary for a single ED_MEAS – CH_CTRL – ED_CTRL transaction, as illustrated in Figure 12. However, in practice, tsLength is a little bigger and it is influenced by many factors, such as the characteristics of the ED's radio transceiver and wireless channel, the use of a power-gating technique, and number of possible retransmissions. In our implementation and also in the simulations, tsLength is 8 s (1 retransmission is supported). When the retransmission is not necessary, which is usually the case, the corresponding reserved time period at the end of a TS window provides a gap between TSs and potentially mitigates drift clock and channel contention issues. Moreover, multipath effect is significantly reduced when very large gaps are employed. As a result, it is possible to extend the communication range of the nodes to values very close to their maximum ones mentioned in their radio module datasheets [4].

Homogeneous Scheduling. It refers to the scenario where all ED nodes of the same network segment have exactly the same sleeping schedule. That is, they wake up/sleep at the same AMs.

Heterogeneous Scheduling. It refers to the scenario where at least two ED nodes of the same network segment have different schedules. A network with multiple segments can have homogeneous and heterogeneous scheduling schemes at the same time.

Dispersion. In heterogeneous scheduling, the dispersion is defined as an anomaly characterized by having some ED nodes using improper MTSs, causing a negative impact on the energy efficiency of the CH node. In other words, the goal of having the maximum number of inactive MTSs is not achieved although it is possible. As already discussed, BETS is designed to be dispersion-free.

Emergency Mode (EM). In normal BETS operation, all EDs in a segment are properly synchronized with the CH node. That is, they correctly follow their assigned time slots. However, when a node is (a) deployed for the first time, (b) restarted, or (c) does not receive CH_CTRL after sending ED_CTRL message, it does not have any time-slot assignment. In this case, the node follows a different algorithm (EM) in order to communicate with the CH node.

Convergence. When all ED nodes in a segment are in regular operation (not in EM), that network segment is said to be *convergent*. In a nonconvergent segment, one or more node can try to contact CH while it is sleeping. Also, the node in EM can interfere with the current assigned TSs, and that segment is no more contention-free (temporarily). While in nonconvergent state, the network has significant energy penalties. Therefore, a design goal for BETS is to have segments that quickly converge.

Node Qualification. Because BETS is a loose protocol in terms of wireless channel error control mechanisms, it is important to establish ways to prevent/mitigate errors at the ED-CH link. The node qualification is a procedure used to address this challenge. During more than 2 years, we have been deploying nodes in different outdoors sites, and the following *node qualification* guidelines are based on our field work experience. Before the final deployment of an ED node, it is recommended to verify the conditions of the wireless channel at the location where an ED node is expected to be deployed, in particular for the ones in critical areas (long distance, ED-CH line-of-sight issues, trees, topology, etc.). First, the average noise floor level is measured (NF). Second, it is measured the received signal strength (RSS) as provided by the ED node in relation to messages sent by CH. The RSS value must be significantly higher than NF (e.g., 5 dB or more). For instance, if the RSS = −87 dBm and NF = −93 dBm, this location is potentially close to the *boundaries* of the network segment, but it is acceptable. In this limit-case, it is also recommended to run communication tests to evaluate the channel conditions for a final decision. If the node cannot pass the qualification test, a new network segment must be used, the CH must change its position, or new antenna schemes must be adopted, and so forth. The bottom line is that the energy performance of BETS is strongly affected by the wireless channel errors as will be discussed in Section 5.

4.4. Design Goals. The main goals of the BETS protocol are summarized as follows.

(1) To be functional in relation to

   (a) providing a way to send fixed or dynamic schedules for ED nodes that are originated at the main application running at the BS node or above (data server);

   (b) collecting sensing and basic control data (network-related errors and energy-related metrics) from ED nodes and send such data to the main application.

(2) To implement the *selfish node* concept and maintain fairness for the wireless channel access.

(3) To be energy efficient (ED side) by imposing a network overhead not higher than 0.75% even if the probability of errors at the communication channel (ED-CH link) is as high as 5%. This overhead limit is motivated by the analysis of the scenario illustrated in Figure 7 specifically related to very low duty-cycle sense-and-send applications.

(4) To be energy efficient (CH side) by allowing CH to have optimum sleeping cycles even in case of heterogeneous scheduling.

(5) To isolate network problems between segments.

(6) To isolate problems related to ED-CH and CH-BS/DS communication links.

(7) To support network management tasks as follows: (a) verify the reliability of individual ED nodes and the ED-CH communication links and (b) isolate erratic ED nodes.

(8) To mitigate the wireless channel contention among the ED nodes of the same segment. BETS must be insensitive to the existence or not of a MAC protocol running above the physical layer.

(9) To provide support for optional use of power hibernation schemes in order to achieve very high energy savings.

Observe that no specific attention is given to network performance metrics, such as maximum transmit delay or throughput. This fact anticipates the most important trade-off of BETS: the network performance is expected to be sacrificed in order to obtain impressive energy achievements in conjunction with excellent scalability and reasonable reliability. The main reason for this trade-off is the isolation between ED-CH and CH-BS/DS data flows (asynchronous approach). Therefore, the possibility to switch from LDC (BETS) to RDC mode is a necessary provision for critical WSN applications.

4.5. BETS: Normal Operation. In order to realize the design goals number 2 (selfish node concept) and number 8 (contention-free), a TDMA approach is used to avoid contention among nodes and allow a fair usage of the wireless channel. By reserving a time slot (TS) for each ED node that will use the same future active MTS (AM), the wireless channel becomes potentially contention-free. When an ED wakes up in its assigned TS, it immediately starts sensing and, without any delay, sends the data to CH. Observe that this sense-and-send approach provides the best energy efficiency possible because no care is given by the ED node related to the possibility of a busy channel or the availability of CH. Clearly, it corresponds to the selfish node concept. In fact, this procedure is autonomously repeated by the ED node from time to time, even if the CH node is not operating. In normal operation, once ED_MEAS is received, it is followed by the CH_CTRL message sent by CH. The CH_CTRL message has two purposes. First, this message acknowledges the proper reception of ED_MEAS. Second, CH_CTRL contains configuration data to ED, such as the next time that the node must be active. Once ED receives CH_CTRL message, it sends back the ED_CTRL message. This message also has two goals. Besides serving as an acknowledgment for CH_CTRL, the ED node uses this message to send its control (log) data: battery status, power shortages, communication errors, and so forth. At the BS side (or above), the main application can recognize erratic patterns associated with a specific node. Therefore, by assigning a very long schedule specifically for that node (e.g., hours or days), this one can be virtually isolated from the network.

The ED_MEAS + CH_CTRL + ED_CTRL *transaction* forms the core of a handshake-based procedure in BETS. However, in contrast with the traditional usage of acknowledgments, missing of one of the messages does not necessarily trigger a retransmission. In our implementation of BETS, we provide a second transaction round at the same TS if the first one fails. This explanation helps to clarify why the default tsLength used in our simulations (and real implementations) is 8 s while the associated messages only sum up to 3 s, according to the Table 2. Besides the retry timing, tsLength also encompasses the potential timeouts associated with collisions and other communication errors. Observe that the mentioned message redundancy provision is not a formal specification of BETS, but tsLength can be increased even more to support multiple retries. In this way, a critical network segment in terms of channel communication errors can have a higher tsLength to increase the likelihood of successful transaction.

In our real-world BETS implementation, all the 3 mentioned messages are sent twice with a small delay between the messages. Empirically, we figured out that such provision highly mitigates the possibility of a missing message in particular for outdoors. This second form of redundancy provides a way to increase the reliability of the communication without having to introduce timeouts or additional complexity at the protocol. Again, such effort is a design possibility, but not a specification of BETS. Nonetheless, this discussion is important to highlight the strength of BETS in terms of its adaptability for different network scenarios.

The acknowledgments provided by CH_CTRL and ED_CTRL messages are primarily used as a network management tool. In other words, it is possible to identify energy and communication problems related to a certain node or a group of nodes. Also, the communication quality is evaluated in both directions (i.e., ED-CH and CH-ED). This later aspect is very important because CH and ED nodes may have different antennas. For instance, an omni-directional one for CH and a directional one for EDs. Alternatively, a higher antenna gain (typically a bigger antenna) for CH and a regular one for EDs. The bottom line is that, in all cases, the goal number 7 (network management) is fully satisfied in BETS. That is, it is possible to detect and correct reliability issues at the network already deployed. On the other hand, the 3 mentioned messages only satisfy the goals number 2 and number 8 when the segment is operating in normal conditions; that is, it is convergent and all nodes are well synchronized. Erratic scenarios under BETS are considered next.

4.6. BETS: Dealing with Erratic Scenarios. So far, the energy-efficiency, fairness, and collision-free channel characteristics of BETS are achieved provided that all nodes properly have and follow their TS assignments (convergence). In this section, the erratic scenarios are considered and the BETS convergence process is discussed.

The first erratic scenario is related to the well-known clock drift issue. Even when all EDs are properly synchronized, the minimal differences among their internal clocks will eventually cause overlapping between TSs. BETS solves this problem by *continuously* providing a schedule adjustment for each ED node. Such adjustment occurs every time an ED node receives a CH_CTRL message. Therefore, in general, the clock drift hardly impacts the solution. However, the combination of very long schedules (e.g., >5 h) and low-quality clock modules must be avoided in order to mitigate the risk of clock drift issues. If such schedules are really expected, two guidelines can be employed. Firstly, higher quality clock systems with temperature compensation can be used at CH and ED nodes. Alternatively, a higher value for tsLength can be used in order to enhance the effective gap between consecutive time slots. Note that this parameter is a key one to properly adapt BETS to the existing topology/network scenario.

The remaining erratic scenarios are basically associated with the same final result, and they can be analyzed in a single scenario under BETS. Specifically, no matter if the ED-CH transaction fails (collision or other communication error), CH/ED node restarts, or an ED node is recently deployed, in all cases the network temporarily is non-convergent. When an ED node timeouts the reception of an expected CH_CTRL message, it automatically enters in *emergency mode* (EM). A related solution typically used in MAC protocols is the combination of channel overhearing with random back-offs. In BETS, only the second part of this technique (random back-offs) is used. The reason for this approach is related to the support of hibernation mode at EDs, as stated in the goal number 9 and explained next.

The fixed-length nature of a TS (deterministic approach) highly promotes the adoption of supercapacitors at the ED nodes as part of the hibernation solution. However, because the power used to overhear a wireless channel is very high, it would be necessary to charge supercapacitors with an amount of energy multiple times the expected one for a single TS transaction, thus resulting in drastic energy inefficiencies. Because this design aspect is closely related to the hardware characteristics, we empirically evaluated different power management solutions for typical WSN nodes, and we concluded that, under the context of BETS, instead of the overhearing scheme, a more energy-efficient solution would be using the back-off technique alone. The EM procedure, with an optional support for supercapacitors, is provided by two algorithms (Algorithms 1 and 2), one for the ED node and the other for the CH node.

In order to provide support for different hardware platforms, some of the parameters in Algorithms 1 and 2 are clearly hardware-dependent. Algorithm 1 is executed once ED node enters in EM, and it is reexecuted for each unsuccessful tryout while in that mode. To generate *randomTime1* and *randomTime2*, a discrete uniform distribution function is used. Assuming that the Node Qualification procedure described in Section 4.3 was considered, all ED nodes are expected to have similar behavior in relation to the EH-CH link performance. Therefore, the mentioned uniform function can potentially be adopted in our model.

Algorithm 2 is executed once CH node starts a new active MTS, and it is reexecuted each time a new ED_MEAS message is received. Note that the basic idea behind the Algorithm 2 is to make the CH node extending the current ETS slot in order to wait more time for the missing/non-convergent children nodes. At the upper part of the algorithm, it is considered the case where no information about the number of the nodes are provided and a higher time extension is provided (not the best energy-efficient approach). At the bottom part of the algorithm, we consider the expected case where the CH knows how many nodes are expected to belong to its segment. In this case, two extra ETS slots (in terms of tsLength time) are included in the time extension for ETS considering the fact that when one node tries to communicate with CH, it can also collide with another one, properly assigned ED, and both are can go EM state. Once one of the non-convergent nodes

```
Require: maxContinuousEM: max. time in EM mode before rebooting
Require: currTime: time elapsed since the beginning of the active MTS
Require: randomTime1: (random value between 5 to 11) $*$ tsLength
Require: randomTime2: (random value between 300 to 720) $*$ tsLength
Require: SCchargingTime: time to charge SCs in EM mode (default: 0)
Require: tryouts: incremented for each unsuccessful CH transaction
Ensure: A segment with up $N = 0.8 * (mtsLength/tsLength)$ EDs converges
  $tryouts \leftarrow 0$
  if $currTime > maxContinuousEM$ then
    Reboot node
  else
    if supercapacitor(s) is(are) used then
      charge supercapacitor(s) for SCchargingTime
    end if
    if $tryouts > 5$ then
      $tryouts \leftarrow 0$
      backoffTime $\leftarrow$ $randomTime2$
      sleep during backoffTime period
    else
      backoffTime $\leftarrow$ $randomTime1$
      idle (radio off) during backoffTime period
    end if
  end if
```

Algorithm 1: Emergency mode (implemented at ED node).

```
Require: expectedNumChildren: # registered children ($N$)
Require: numChildrenCurrentMTS: # pending EDs with TSs in this MTS
Require: numFutureChildren: # EDs with assigned TSs in future MTSs
Ensure: A segment with up $N = 0.8 * (mtsLength/tsLength)$ EDs converges
  return waitEDTime: how much time CH must wait hearing children
  if CH just booted then
    $waitEDTime \leftarrow 20 * mtsLength$
  else
    if $expectedNumChildren = 0$ (CH never contacted BS node) then
      if $numChildrenCurrentMTS = 0$ (no assigned TS in this MTS)
      then
        $waitEDTime \leftarrow 31 * tsLength$
      else
        $waitEDTime \leftarrow tsLength * (numChildrenCurrentMTS + 5)$
      end if
    else {(BS had already sent the number of EDs in this segment)}
      $aux \leftarrow numChildrenCurrentMTS + numFutureChildren$
      if $aux = expectedNumChildren$ then
        [(convergence achieved)]
        $waitEDTime \leftarrow tsLength * (numChildrenCurrentMTS + 2)$
      else {(potentially one or more EDs missing in this AM)}
        $waitEDTime \leftarrow tsLength * (expectedNumChildren - numFutureChildren + 2)$
      end if
    end if
 end if
```

Algorithm 2: Emergency mode (implemented at CH node).

finally succeeds, the ETS is extended again considering the new scenario of how many nodes already synchronized and how many are missing.

In this work, we will limit the verification of the correctness of these algorithms by analyzing the simulated results in Section 5. In our simulations and context, the goal is to have the network convergence achieved in less than the time of $(4 * sched_{\min})$, where $sched_{\min}$ is the smallest of the schedules among the ED nodes that are already synchronized with CH. If all nodes are in EM, $sched_{\min}$ is the default schedule used by CH temporarily without children. Because such values are on order of many minutes, it is clear that when a dual system (RDC/LDC modes) is used, the frequency of the switching process is clearly impacted by the long time (e.g., >15 min) necessary to converge the network under BETS. If this mode switching occurs a few times per day, this is not an issue. However, the need of more frequent switching requires a kind of support not natively provided by BETS. One potential and relatively easy way to mitigate this issue in dual systems is to sequentially add EDs to the segment (rather than all nodes at the same time) when the DDC system switches from RDC to LDC. Again, although such provision is not a BETS mechanism, it is clear that this procedure just requires a small development effort at the software side of the radio transceiver module.

The most important aspect behind these algorithms is the fact that the convergence can be achieved with any number of ED nodes provided that the CH is active for enough time, because the maximum continuous time the CH can wait for children is limited by mtsLength. Therefore, this parameter essentially governs the convergence feasibility. However, tsLength is ultimately the parameter that influences mtsLength. For instance, with a tsLength = 8 s and 100 ED nodes, the minimum mtsLength is 1000 s (16.6 min). As a result, the application schedule must be an integer multiple of this value, such as 20, 40, and 60 min. In our implementation, we opted by a maximum number of 30 nodes, leading to mtsLength of 5 min, which is more useful from the viewpoint of many existing environmental monitoring applications.

With smaller values of tsLength, much more nodes can be supported in a segment while practical values for mtsLength are still maintained. Therefore, one natural question is associated with the adoption of a high value for tsLength (e.g., 8 s) when it is well known that typical RF transceivers can realize the full transaction in less than half a second. Besides the additional time for a possible retransmission and for a secure gap between TSs, the answer lies in the choice of the hardware platform. In our implementation, a single transaction was actually achieved with less than 2 s, and finally 8 s were determined as a secure value based on experiments and taking into account our transceiver choice, power-gating latency [9], reliability aspects, and critical outdoor environments. The latter aspect is important for our ongoing project because we would like to achieve ED-CH distances very close to the informed by the manufacturers of the radio modules. When the messages in the network are separated by significant gaps, the multi-path effects in hilly regions are significantly reduced, as already highlighted, and this fact is observed empirically, as discussed in Section 5. Nonetheless, a higher tsLength typically will not affect the energy efficiency of EDs provided that network errors do not occur frequently. However, a higher tsLength can definitely impact the CH node because it must be active more time during each cycle. The next section discusses the energy efficiency of the CH node.

4.7. Energy Efficiency of the CH Node. The maximum energy efficiency at the CH side is achieved with a homogeneous scheduling. In this case, the nodes of one network segment always use the same AMs whenever they are expected to sense-and-send, and the goal of achieving the maximum number of inactive MTSs is trivially achieved. Although this fact does not represent any additional energy-related advantage for the ED nodes, it definitely extends the sleeping time of the CH which is the node with the most critical energy constraint in the network segment. Moreover, continuous and long sleeping periods maximize the efficiency of the power-gating technique because there are energy penalties associated to the switching transients.

However, considering that an energy-aware system running is on top of the BS/DS (which is a guideline of this framework), such system can potentially select different schedules for EDs. In this case, the already mentioned *dispersion* problem can occur and must be addressed by BETS. Although the schedules are sent by the main application running on the BS side, due to its cross-layer nature, the CH is effectively the node that controls the distribution of the schedules to its children. Based on this observation, it is possible to implement an algorithm at the CH side that can properly adjust (advancing or delaying) the next active-time information (via CH_CTRL message) sent to each ED in order to avoid dispersion. Such approach is used by Algorithm 3 which is a dispersion-free procedure for the CH node providing maximum energy efficiency for this node, as stated in the goal 4. There are some interesting aspects that need to be highlighted in the Algorithm 3.

(i) The time of the entire network segment is referenced uniquely by the CH clock. When the CH is initialized, the moment t_0 is defined and the MTS 0 has its beginning. No real-time information is exchanged through BETS: when an ED node transmits ED_MEAS, CH timestamps it with a real-time value based on its local clock. This scheme helps to maintain BETS very light but, as expected, measurements with time-stamp errors on the order of seconds can occur. Fortunately, it is rarely an issue for non-real-time applications running in the LDC mode of the network.

(ii) The least common multiple (LCM) principle is used to avoid dispersion. For instance, if *currMTS* is 13 and an ED node (just initialized) has an assigned schedule of 5, we can initially expect that it wakes up again at MTS 18. However, this is not the case. In Algorithm 3, the variable *aux* has the value $floor(13/5) = 2$, and because $nextMTS = (2 + 1) * 5 = 15$, this node will actually wake up again 2 MTSs ahead (MTS 15 = 13 + 2), not at MTS 18. This adjustment only occurs

```
Require: schedED_Table: ED schedules assigned by main app (BS side)
    fields: (ED_id, $x$), where $x$ is an integer multiple of mtsLength
Require: MTS_Table: MTS allocation table (old, current, and future AMs)
    fields: (ED_id, MTS_seq), where MTS_seq is the next AM for ED_id
Require: fixedDelta: readiness time for CH (from wake-up to hear EDs)
Require: currMTS: current value of MTS (numbered sequentially)
Require: currED: id of the ED on the current transaction
Ensure: TS overlapping does not occur
Ensure: Dispersion-free TS allocation: maximum energy efficiency
    return adjSch: next ED activation time (to be sent via CH_CTRL)
    // dispersion-free:
    $x \leftarrow \mathit{schedED\_Table}[\mathit{ED\_id}]$
    $\mathit{aux} \leftarrow \mathit{floor}(\mathit{currMTS}/x)$
    $\mathit{nextMTS} \leftarrow (\mathit{aux} + 1) * x$
    $\mathit{updateMTS\_Table}(\mathit{ED\_id}, \mathit{nextMTS})$
    // no TS overlapping, continuous TSs assignment:
    $\mathit{adjSch} \leftarrow \mathit{nextMTS} * \mathit{mtsLength}$
    $\mathit{adjSch} \leftarrow \mathit{adjSch} + \mathit{fixedDelta}$
    $\mathit{numEDsSameMTS} \leftarrow \mathit{getNumEntriesSameMTS}(\mathit{MTS\_Table}, \mathit{ED\_id})$
    $\mathit{etsAdj} \leftarrow (\mathit{numEDsSameMTS} - 1) * \mathit{tsLength}$
    $\mathit{adjSch} \leftarrow \mathit{adjSch} + \mathit{etsAdj}$
```

Algorithm 3: Dispersion-free TS allocation (CH node).

at the first assigned cycle for that node. From that moment on, the node continues following the expected scheduling. In short, the dispersion is avoided because this node has this wake-up sequence in the line of time: $(13, 15, 20, 25, \ldots)$.

(iii) After calculating the new AM for the node, an adjustment is performed by CH in order to avoid TS overlapping. In other words, if 3 nodes have the same cycles (schedules) to operate, we do not want all of them to send their messages at the same time, that is, at the beginning of that AM. To avoid this issue, the number of current EDs that share the same future AM is recorded. Every time CH allocates a new ED for that AM, it adds a specific delay in order to have all the nodes accessing CH in an efficient, sequential, and contention-free manner. The maximum energy efficiency is achieved for the CH node.

(iv) The algorithm is implemented without any historical control or complex procedures/data structures. Such simplicity provides the path to implement the CH side of BETS in hardware platforms already used by regular WSN nodes. This aspect is extremely important in a dual mode system (with RDC/LDC modes) because the CH nodes are assigned among regular WSN nodes. If the CH requires a very powerful main MCU (which is not the case), the assignment of CHs would be impacted.

5. Simulated and Empirical Results (LDC Mode)

In order to verify the correctness of the BETS algorithms and also to determine the constraints of the solution, experiments are performed. In addition to simulations, preliminary results from our field deployments are provided. The following are some questions of the particular interest in this context.

(i) Are the algorithms correct and is goal number 3 (ED energy-efficiency) feasible?

(ii) Assuming that CH was just installed, how much additional time is necessary to achieve convergence?

(iii) Given a certain probability of errors for the communication channel between ED and CH nodes, what is the associated energy penalty?

(iv) When will the network not converge?

5.1. Experimental Setup. For this work, we developed a specific network simulator for BETS (MATLAB environment). A single segment is simulated, but any number of EDs is supported. Due to its asynchronous nature, we omit the CH-BS communication (BTS length = 0). Each individual simulated scenario involves a minimum of 1,000 iterations. In relation to communication errors, a uniform distribution is considered, as explained in Section 4.6. Also, such probability is independent among nodes increasing the likelihood of errors in the network. The probability of communication channel issues is independent of the probability of collisions, better representing a real scenario. Also, for the simulations involving 1 min schedule, CH is assumed to not sleep for practical reasons. Specifically, there is a significant energy cost associated with the power activation of a node. If the activation/deactivation cycles associated with a hibernating node are very frequent, it is possible that the energy cost due to the transients is higher than the hibernating savings.

When not specified, the parameters used in the simulations are the same ones used in our real-world implementation of BETS also discussed in this section. Such parameters

Table 4: Default parameters for simulations.

ED_MEAS time	1 s	CH_CTRL time	1 s
ED_CTRL time	1 s	ED_MEAS timeout	3 s
ED_CTRL timeout	2 s	RandomTime1	5–11 s
RandomTime2	5–12 min	Measurements time	1 s
MaxTimeStartLastED	5 min	SCchargingTime	11.40 s
ExpectedNumChildren	value of N	tsLength	8 s
MaxContinuousEM	1 h		

are listed in Table 4. The parameter *maxTimeStartLastED* requires an explanation. When convergence tests are realized, the EDs are randomly turned on during a certain amount of time, and *maxTimeStartLastED* represents a limit for this time. For energy-related calculations, Table 2 is used and a 3.6 V, 19 Ah non-rechargeable battery is assumed as the single source of power. Accordingly, many results are given in terms of life expectancy for the node, and one can simply convert back to energy values in Joules assuming that 100% of the initial energy is actually used by the node. This assumption is realistic if the *power matching* technique presented in [9] and supported in this framework is used. With such technique, the current pulse effect discussed in Section 3 is avoided.

Note that by assuming the usage of non-rechargeable battery as the single power source, it becomes straightforward to verify if the solution would reach a very long lifetime independently of the existence of an energy scavenging system for the nodes. Similarly, such estimated lifetime considers the LDC-only mode. Therefore, the energy costs associated with a more critical WSN application (RDC mode) are not computed. In short, both user-defined energy unknowns from the viewpoint of the framework, the possible amount of harvested energy, and the energy spent in a demanding WSN application are removed in this analysis. However, in a real-world implementation, such information naturally must be included according to the available energy resources and application demands. In our current project, the nodes are being deployed with non-rechargeable batteries, and the system is currently LDC-only due to the low duty-cycle characteristics of the application. In this case, the estimated values for lifetime provided in this section are exactly the ones used in the energy analysis and decisions under this project.

Related to the empirical investigation, the BETS solution has been implemented in eight distinct outdoor networks since August 2011. Six adjacent sites have a total of 150 nodes (110 already deployed) covering a 36 km × 36 km continuous area. The results presented here are from the two oldest networks in terms of operation time. The first network features 1 BS, 1 CH, and 26 ED nodes. The CH and EDs are attached to 3 soil moisture sensors. The maximum ED-CH distance is around 150 m. The irregular topography and the existence of obstacles (trees and plants) are the main challenges for this site. The second site, a cow farm, is composed of 21 ED nodes, and the maximum ED-CH distance is around 350 m. Previously, we had faced problems in this site related to (a) solar panels versus cows, (b) rechargeable-batteries versus extreme temperatures, (c) complex support for unattended 802.15.4-based routers, and (d) scalability/overhead issues associated with the ZigBee protocol and sparse networks [2, 4]. Accordingly, the BETS design was strongly influenced by the lessons of that work.

5.2. Performance Evaluation.

In this section, the experimental results are discussed in relation to three aspects: (a) convergence time, (b) impact of communication errors, and (c) impact of heterogeneous scheduling.

Convergence Time. In Figure 13, the convergence time is given as a function of the number of nodes for different node's schedules, and a homogeneous scheduling is assumed. Also, only communication errors due to the contention collisions are considered. The convergence value here represents the additional time, besides the schedule value, for the last ED node to converge. Because all the nodes are randomly turned on, such scenario represents the worst-case which is associated with a stronger channel contention for the initial moments. For dual mode systems (RDC/LDC), such convergence impact can be eliminated by providing a progressive activation of the nodes, as already discussed.

For clarity, only the variance of the 20 min schedule case is presented. As expected, higher duty-cycles aggravate the convergence. For instance, a segment with 30 nodes takes around 15 min to achieve convergence if 20 min schedule is used. However, if a 5 min schedule is defined for the nodes, the converge can take around 35 min. Similarly, a higher number of nodes impact the convergence time. A segment with 20 nodes typically converges in less than 15 min even if a 5 min schedule is in place. On the other hand, for the same schedule, it is possible that a network with more than 30 nodes only achieves convergence after hours. It is important to highlight that if CH randomly restarts, a scenario similar to this simulation will occur. Therefore, such convergence analysis provides important insights for the parametrization of BETS in a given segment. In our real-world deployments (20 min-sched), we do not observe convergence time higher than 1 h, and the average value for convergence is found to be around 40 min. Such value is close to the upper bound of the variance line for 20 min cycles in the figure. Because, this simulation does not include any communication channel error besides the collisions, the theoretical model has a fair agreement with our empirical evaluation. Finally, it is important to highlight that a nonconvergent network does not imply significant lack of functionality. Even for the mentioned cases involving 40 min for convergence time, typically the majority of the nodes synchronize at the first MTS. However, it takes more time to have all the nodes synchronized.

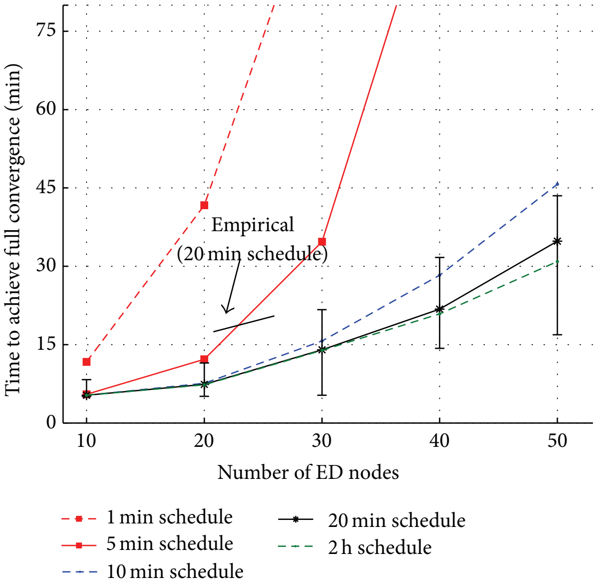


Figure 13: Convergence time assuming that all EDs are turned on randomly during a 5 min period.

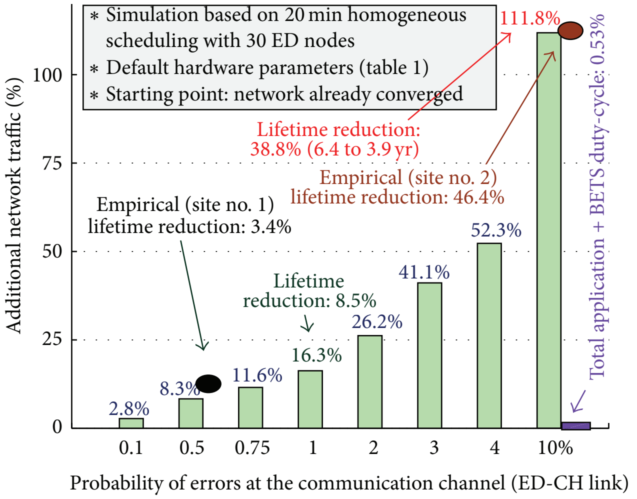


Figure 14: Impact of the communication channel error (ED-CH link) on the BETS network performance.

Impact of Communication Errors. Once the convergence at the network segment is achieved, BETS becomes highly deterministic, and its theoretical network overhead is negligible (e.g., $\ll$0.75%) assuming low duty-cycles applications and no communication errors. However, for the next simulation, we want to verify if this design goal number number 3 also holds when the probability of communication errors is high (e.g., 5%). It is assumed that the network has already converged, and 30 EDs follow 20 min cycles. In Figure 14, the additional network overhead is given as a function of the channel error probability. This overhead represents the additional number of messages due to the retransmissions and also possible additional collisions. These secondary collisions occur because a non-convergent node can transmit at the TS assigned to other node. As shown in the figure, even a small error rate, such as 1%, causes an overhead of 16.3%. A higher error rate, such as 10%, doubles the network traffic. However, when we calculate the effective duty-cycle of BETS for this worst error case, it turns out to be around 0.53%, and the design goal number 3 is shown to be satisfied.

Nonetheless, if the mentioned error rates are not temporary, a strong lifetime reduction for the nodes is expected. This result highlights that the lack of extensive error control in BETS has the clear trade-off of making the protocol very sensible to communication channel errors. The lifetime impact is slightly smaller than the additional traffic rate, but it is still very high. For our empirical investigation, we analyzed the data that arrived at the BS side. Because it is possible to easily detect duplicates, missing data, and the number of retries that each node had experienced, an accurate energy profile for the nodes is feasible. The results of such analysis are also shown in Figure 15 but now including the options of 20 and 40 nodes in the segment. It is clear that the mentioned trade-off is aggravated with a higher number of nodes. For instance, consider the 4% error rate case: the network with 40 nodes has its life expectancy shortened by more than 1 year compared with the case where the network only has 20 nodes. Therefore, the effect of the channel error on the BETS performance is strongly aggravated when more nodes are involved. This fact is explained by the best-effort approach used by BETS to achieve convergence: the process is relatively slow when the number of EDs is relatively high. In other words, it can take many minutes for a node that experienced communication error to converge again and, while in EM state trying to contact CH, it is wasting energy.

Returning to the analysis of Figure 14, it also has empirical results to be discussed. For sites number 1 and number 2, the average error rates of around 0.6% and 11%, respectively, are calculated based on measured network metrics. These results are in agreement with the fact that site number 2 has very high ED-CH distances. Now, using Figure 15, it is possible to

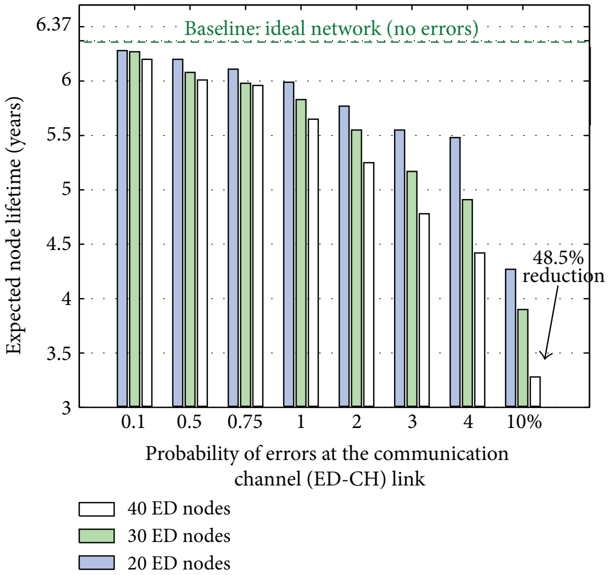


Figure 15: Impact of the communication channel error (ED-CH link) on the energy performance. Similar scenario of Figure 14 for different number of nodes.

infer about the expected lifetime of the nodes. For instance, for site number 2 (21 nodes, 11% average error rate), it is possible to forecast a lifetime of around 4 years. For an existing WSN solution to be superior in terms of energy efficiency, it needs to have a total network overhead smaller than 1.8% (see Figure 7). To date, site number 2 has been in continuous operation for 24 months and, so far, the solution is comparable to any existing solution that has an effective network overhead of less than 4%. These results are pretty significant considering the coverage area and the number of nodes involved: it is a sparse network and many WSN protocols potentially would fail in such site [4]. Moreover, just a single CH node is being used and the energy consumption among ED nodes is guaranteed to be homogeneous, assuming that they have the same application schedule and the same average communication error rate. Finally, although the energy costs are significant with high communication error rates, our implementation of BETS proved to also have good communication reliability: the data losses for sites #1 and 2 are smaller than 1.4 and 1.8%, respectively. These numbers are relatively superior than the average for outdoors deployments reported in the WSN literature.

Impact of Heterogeneous Scheduling. So far, the simulations considered homogeneous scheduling. However, in some scenarios an adaptive sensing scheduling is desired [2, 14]. In the next simulation, the energy performance of BETS under homogeneous and heterogeneous scheduling is evaluated. In Figure 16, the relative energy consumption of the CH node for a 24-hour period is shown as a function of different scheduling schemes. The reference case is a 20 min schedule involving 30 nodes. The goal is to estimate how much energy (additional or reduction) is associated if the scheduling scheme changes. The first 3 left most cases involve the same 20 min schedule and 10, 20, and 30 nodes. Next, 15 nodes in a 10 min schedule. The following case is special: 20 nodes in a heterogeneous scheduling, half following a 10 min schedule, and half a 20 min schedule. The last case (rightmost) involves only 8 nodes but with a more frequent 5 min schedule. It is assumed a convergent and error-free segment.

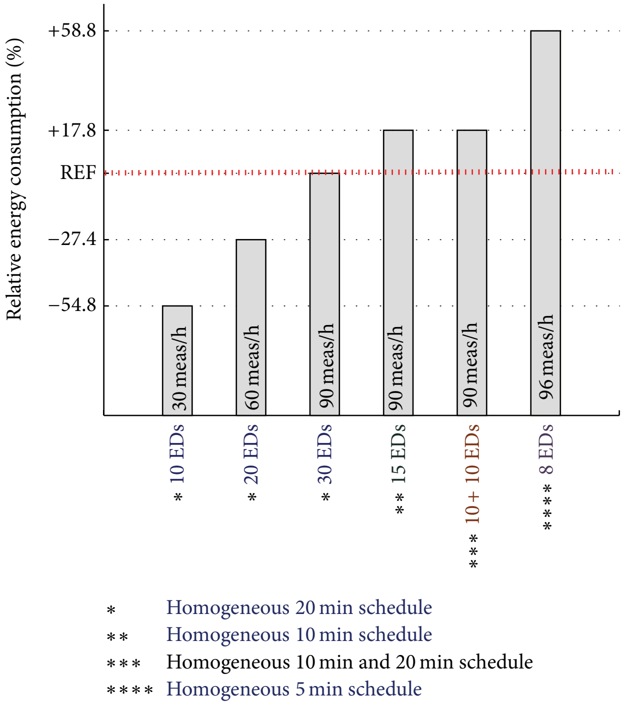


Figure 16: CH energy consumption for homogeneous and heterogeneous schedulings.

The number of ED_MEAS messages received per hour by the CH node is already calculated. Note that in Figure 16, the same number of ED_MEAS messages received by CH in three distinct scheduling schemes, 90 meas/h, does not imply the same energy consumption of the CH node. The underlying factor that mainly governs the energy performance of the CH node is the number of AMs for a certain period of time, such as 24 hours. If homogeneous scheduling is in place, the optimum scenario in relation to the number of AMs is achieved. However, for heterogeneous scheduling, the number of AMs can potentially increase, and the calculation of the energy spent by the CH node is more complex, as explained next.

(i) In each MTS, there is an extra time (dynamically value determined by software) allocated at the ETS as a provision for possible communications failures. The higher the number of AMs is, the higher the sum of these time provisions spent by CH and worse its energy efficiency are.

(ii) Before the ETS beginning, the CH node must be ready for the radio reception and a certain amount of such *readiness* time is provided. In our implementation, such CPU time has an average of 30 s (it also performs sensor measurements), and the corresponding consumed energy is relatively high. In short, the higher the number of AMs, the higher the energy spent with this processing phase.

(iii) Hardware state transition: the higher is the number of AMs, the higher is also the energy cost due to the transients while turning-on the modules [10]. In particular, such cost is critical when power-gating techniques are used.

The basic scenarios associated with an increase of the number of AMs are illustrated in Figure 16. An interesting aspect to highlight is related to the three leftmost cases using the same 20 min schedule. A linear relation exists in relation to the number of nodes and energy consumption. However, there is a hidden fixed energy cost in all cases, and it is related to the establishment of AMs. In this case, the number of AMs is exactly the same (72 per day) and the mentioned fixed cost is the same. This analysis indicates that the higher the number of EDs sharing the same AM, the higher the CH energy efficiency. The next two cases involve 15 nodes (homogeneous scheduling) and 10 + 10 nodes (heterogeneous scheduling). The number of AMs involved in both cases is the same (144 per day). The heterogeneous scheduling is not adding more AMs because 20 is a multiple of 10, and all AMs related to the nodes that follow the 20 min schedule are also shared with the nodes that follow the 10 min schedule. Therefore, the heterogeneous scheduling is not imposing an additional cost comparing these two cases, as proved by this simulation.

Finally, if we compare one of these two just mentioned cases with the one with 30 nodes, the number of ED_MEAS messages received by CH per hour is exactly the same (90). However, the number of AMs for the case with 30 nodes is smaller (72 per day) and this fact explains why the CH is consuming less energy compared to the other case. In short, to save energy at the CH side, both the number of EDs and the number of AMs cannot be very high, and the latter aspect is directly related to the employed scheduling scheme.

The analysis of the results in this section strongly suggests that the LDC mode can be better exploited in many scenarios where the network is not continuously under high demand. Nonetheless, the proposed framework based on a switching RDC/LDC scheme or simply on the LDC-only mode can still be enhanced by integrating it with state-of-the-art approaches involving energy efficiency for WSNs, as will be discussed next.

6. Future Directions

As already discussed in Section 2, the energy-management efforts can be applied at different levels, and the components of the proposed framework are implemented in all these levels: node, network, and at a centralized server. However, the framework can also be extended by being merged to other state-of-the-art efforts, and this discussion is presented in this section.

In this work, in particular for scenarios where the power depletion is a real risk, the importance of using some sort of *backup* energy reservoir in conjunction with an energy-harvesting system has been highlighted. So far, the focus has been on the nontraditional use of primary cells. The main reason for this choice is the combination of a high energy density, robustness in relation to extreme temperatures, lifetime, and relative low cost. However, besides this option, other forms of backup energy storage have been reported. For instance, in [33], a survey on multisource energy harvesting systems is provided and different combinations of energy storage components are discussed, including the recent fuel cell technology. In this context of hybrid energy system, the same work highlights the advantages associated with the adoption of the Smart Power Unit (SPU) [16] or System A architecture. Such energy system architecture has its own MCU, and it corresponds to the energy-management subsystem discussed in Section 3.2, shown in Figure 5.

Still at the node level, besides the mentioned efforts involving the hardware of the node, it is also possible to increase its energy efficiency by means of software actions. In this track, techniques such as data aggregation and data compression can be investigated [34]. In this case, the main goal is to exploit the data correlation among nodes or data sets in order to reduce the network traffic and ultimately the energy spent by the nodes. In many cases, such techniques are associated with penalties in terms of data quality that must be properly evaluated as acceptable or not. For instance, based on the compressive sensing (CS) theory, it can be feasible to represent sparse signals with fewer samples than required by the Nyquist sampling theorem [35]. Accordingly, the problem of monitoring soil moisture is studied following this approach with excellent results [36]. Similarly, many WSN applications can achieve better energy efficiency by adopting a proper CS technique. The bottom line in this discussion is that, in order to achieve a higher energy efficiency, typically it is necessary to sacrifice one or more QoS metrics for the WSN node. In our framework, the focus is on acceptable/manageable penalties in terms of network performance. Similarly, CS and related techniques are associated with different penalty levels in terms of data quality. Such conclusion is in accordance with the previous discussion about the energy effort tripod concept (Figure 2), where the importance of investigating how flexible the *application* requirements are also highlighted.

Returning to the analysis of the proposed framework, some future directions and open research aspects are the following.

Temporal Segmentation. One can think that the RDC/LDC switching is an action that affects all the network as a whole. However, it is not necessarily the case, and the DDC operation can actually be applied only to a small part of the network. In this way, a temporal and very efficient network segmentation can be adopted assuming that a certain level of node redundancy is also in place. For instance, assume that it is detected that 15% of the nodes in a network are currently approaching critical remaining energy levels. By assigning the LDC mode only to this group of nodes, it is possible to *remove* them from the regular node operation without losing control of them. In this example, they can continue to regularly report their state to a central application (e.g., every 6 hours). Note

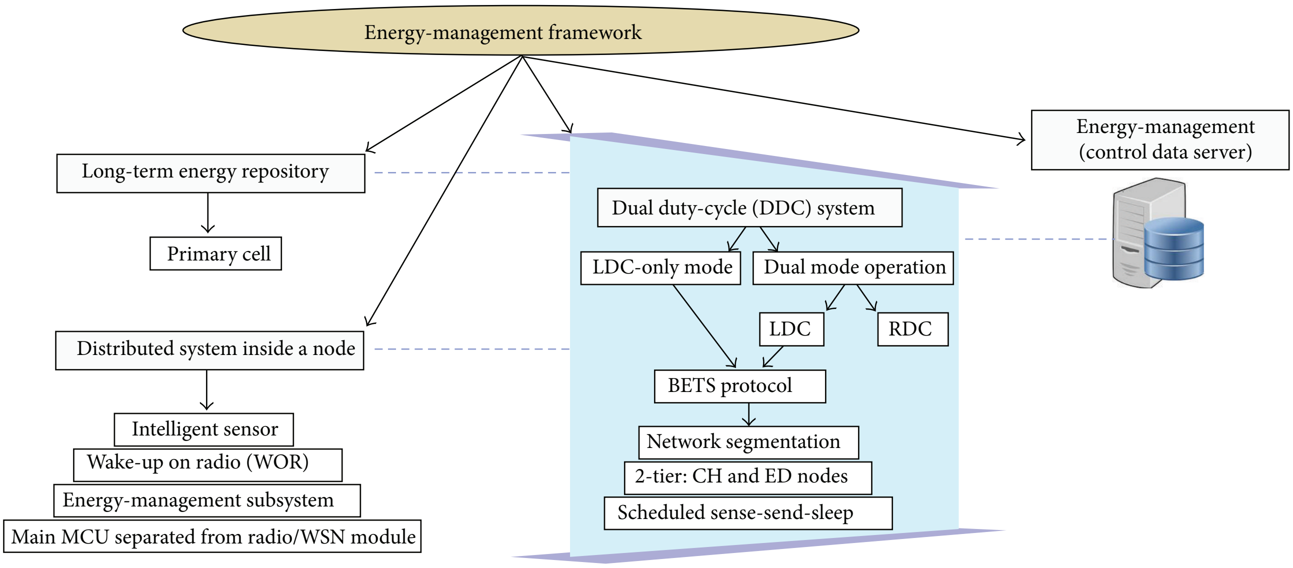

Energy-management framework
Long-term energy repository
Primary cell
Distributed system inside a node
Intelligent sensor
Wake-up on radio (WOR)
Energy-management subsystem
Main MCU separated from radio/WSN module
Dual duty-cycle (DDC) system
LDC-only mode
Dual mode operation
LDC
RDC
BETS protocol
Network segmentation
2-tier: CH and ED nodes
Scheduled sense-send-sleep
Energy-management
(control data server)

Figure 17: Proposed Energy-Framework.

that this solution is significantly superior compared to the majority of existing WSN solutions because while the existing WSN application is preserved, the lifetime of the nodes can be realistically extended. Moreover, subsets of nodes can be periodically selected to go to LDC mode.

Autosegmentation. While the previous idea is mainly based on a central server as the trigger agent to perform the RDC/LDC switching actions, it is also possible to let such decision be commanded by the node itself. For instance, assume that a node detects that it is approaching a critical remaining energy level. While in RDC mode, it can send a message to a central server warning its decision to switch to LDC mode, and it can potentially receive information about the existing CH node to be contacted and additional details related to the LDC setup (RF channel, ID, etc.). In this way, a virtual *parallel* network is autobuilt where the associated nodes continue to report their sensing data or simply their health state in a relative low pace to a central server.

Two Radio Transceivers. One interesting way to implement a DDC node is the adoption of two radio transceivers, one for the LDC mode and the other for the RDC mode. Such *out-of-band* solution has the advantage of easy implementation, and it is highly recommended if the application requires frequent RDC/LDC switching.

7. Conclusions

An open energy-management framework for WSNs is proposed in this work with a strong emphasis at the realistic achievement of a functional and reliable solution with a very long maintenance-free lifetime (e.g., >5 years) for the nodes. The components of the proposed framework are shown in Figure 17. By means of a detailed and systematic preliminary analysis, it is shown that energy scavenging systems are typically necessary to achieve this goal. However, in order to also increase the reliability of the solution, a long-term energy repository is recommended. The traditional candidate for this role is a rechargeable battery, but considering the mentioned very long-term lifetime, a primary cell is a better choice. Besides the selection of the proper energy resources for the node, it is also very important to control the energy used by the nodes and in the network in general. An excellent strategy to achieve an energy-efficient solution is to balance the efforts at the node, network, and application levels. The latter kind of effort is realized by means of an energy-management system hosted in a central data server. By energy measurements received from the nodes, by estimation techniques, or both, the centralized energy-management system can evaluate the current and future remaining energy at the nodes and depending on the application QoS metrics, it can typically activate and deactivate sensing nodes based on location and time. In this work, the emphasis is given on the energy-management efforts at node and network levels.

It is proposed that multiple MCUs compose the sensor node, that is, a distributed system inside a node. With the technological advances, such proposal does not imply a higher energy consumption but a better way to control individual modules of a node although adding complexity and costs to the WSN project. Inside this node architecture called dual duty-cycle (DDC) node, the traditional WSN node is no more considered the main MCU of the node, but simply a radio transceiver module. Moreover, besides the mentioned main MCU and radio transceiver modules, the energy-management subsystem also has its own MCU. In this context, recent availability of intelligent sensor probes is also discussed, and this component can represent another MCU in such DDC node.

Besides the focus on energy-related hardware aspects, this work also provides guidelines related to the energy efforts at the network level. A dual duty-cycle (DDC) system is proposed as part of the framework. From the viewpoint of a DDC system, low duty-cycles (LDC) data-collection applications are classified as belonging to the LDC-only class. However, many WSN applications are not LDC-only, and they are classified as regular or high duty-cycle (RDC) applications. In order to achieve maximum energy efficiency, the network must operate in LDC mode the majority of the time, if possible. LDC-only class of applications permanently satisfies this goal. However, for RDC applications, it is still possible to have the network operating temporarily in LDC mode and returning back to RDC mode. This is the case, for instance, of indoors WSNs deployed in office buildings and operating during nonoffice hours. In order to save energy during this period of time, the network can operate in a form of *stand-by* mode, and this is actually what the LDC mode provides for the network. As expected, the network eventually will return to the RDC mode, and the mechanisms of the DDC system provide such operational mode switching.

While operating in LDC mode, the nodes are actually running the code at the main MCU, not the code at the legacy WSN nodes. Such code includes the novel cross-layer protocol called best-effort time-slot allocation (BETS). Under the BETS protocol, the nodes assume a different logical topology according to the BETS logical network segmentation and its 2-tier architecture. By simulations and empirical results, BETS is proved to be very energy efficient and typically has an effective network overhead smaller than 1%. Its main drawback is a low data throughput which is the reason why this solution is not applied for the RDC or regular mode of operation of many WSN applications. Nonetheless, because it is possible to switch between LDC and RDC modes, many current WSN platforms can still be enhanced with the proposed energy framework. For instance, when it is allowed to have the network temporarily operating with low data throughput, the LDC mode can be activated, and the network is then controlled by the BETS protocol. In this way, the network can experience its best energy performance for the duration of the LDC period.

Finally, some ways to take advantage of the LDC mode in networks that mainly operate in RDC mode are highlighted. For instance, when a node approaches critical energy levels, it can autoswitch its operating mode to LDC thus reporting measurements (or health status) in a very low pace fashion while recovering from its energy depletion. Nonetheless, as identified in this work, the potential main drawback associated with the RDC/LDC mode switching is the need of some WSN applications of quickly resuming the hibernation mode when a nonregular event occurs. The key-component is an ultra-low-power wake-up on radio (WOR) module for DDC nodes. Although recent advances in the WOR area promise to address such need, additional research effort is still necessary in order to integrate WOR modules into DDC nodes.

Acknowledgments

This work was performed at the University of Southern California and at the University of Michigan under support from the National Aeronautics and Space Administration, Earth Science Technology Office, and Advanced Information Systems Technology Program. Agnelo R. Silva is also supported by the Brazilian National Research Council (CNPq) scholarship under the Brazilian Programme *Science without borders*.

References

[1] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 463–468, Los Angeles, Calif, USA, April 2005.

[2] M. Moghaddam, D. Entekhabi, Y. Goykhman et al., "A wireless soil moisture smart sensor web using physics-based optimal control: concept and initial demonstrations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 3, no. 4, pp. 522–535, 2010.

[3] M. Moghaddam, X. Wu, M. Burgin et al., "Ground network design and dynamic operation for validation of spaceborne soil moisture measurements: Initial developments and results," in *Proceedings of the Earth Science Technology Forum conference (ESTF '10)*, June 2010.

[4] A. Silva, M. Liu, and M. Moghaddam, "Ripple-2: a non-collaborative, asynchronous and open architecture for highly-scalable and low duty-cycle WSNs," in *Proceedings of the 1st ACM Annual International Workshop on Mission-Oriented Wireless Sensor Networking (MiSeNet '12)*, pp. 39–44, ACM, New York, NY, USA, 2012.

[5] J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra-low power wireless networking," *IEEE Computer*, vol. 33, no. 7, pp. 42–48, 2000.

[6] IEEE 802. 15. 4 Standard, "Part 15. 4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)," IEEE, Piscataway, NJ, USA, 2006.

[7] ZigBee Alliance, *ZigBee Specifications*, ZigBee Standard Organization, San Ramon, Calif, USA, 2008.

[8] A. S. Weddell, N. R. Harris, N. M. White et al., "Alternative energy sources for sensor nodes: rationalized design for long-term deployment," in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (IMTC '08)*, pp. 1370–1375, Victoria, Canada, May 2008.

[9] A. Silva, M. Liu, and M. Moghaddam, "Power-management techniques for wireless sensor networks and similar low-power communication devices based on nonrechargeable batteries," *Journal of Computer Networks and Communications*, vol. 2012, Article ID 757291, 10 pages, 2012.

[10] A. G. Ruzzelli, P. Cotan, G. M. P. O'Hare, R. Tynan, and P. J. M. Havinga, "Protocol assessment issues in low duty cycle sensor networks: the switching energy," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '06)*, pp. 136–143, Taichung, Taiwan, June 2006.

[11] H. Danneels, V. D. Smedt, C. D. Roover et al., "An ultra-low-power, batteryless microsystem for wireless sensor networks," in *Proceedings of the 26th European Conference on Solid-State Transducers (EUROSENSOR '12)*, Krakow, Poland, September 2009.

[12] EnerChip Smart Solid State Batteries, http://www.cymbet.com/products/enerchip-solid-state-batteries.php.

[13] R. Misra and C. Mandal, "ClusterHead rotation via domatic partition in self-organizing sensor networks," in *Proceedings of the 2nd International Conference on Communication System Software and Middleware and Workshops (COMSWARE '07)*, Bangalore, India, January 2007.

[14] J. C. Lim and C. Bleakley, "Adaptive WSN scheduling for lifetime extension in environmental monitoring applications," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 286981, 17 pages, 2012.

[15] D. Zenobio, K. Steenhaut, M. Celidonio, E. Sergio, and Y. Verbelen, "A self-powered wireless sensor for water/gas metering systems," in *Proceedings of the IEEE International Workshop on Energy Harvesting for Communication*, pp. 5772–5776, June 2012.

[16] M. Magno, S. Marinkovic, D. Brunelli, E. Popovici, B. O'Flynn, and L. Benini, "Smart power unit with ultra low power radio trigger capabilities for wireless sensor networks," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE '12)*, Dresden, Germany, 2012.

[17] S. J. Marinkovic and E. M. Popovici, "Nano-power wireless wake-up receiver with serial peripheral interface," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1641–1647, 2011.

[18] Atmel Corp, "Sleepwalking helps conserve energy," http://atmelcorporation.wordpress.com/2013/04/16/sleepwalking-helps-conserve-energy/.

[19] P. Dutta, D. Culler, and S. Shenker, "Procrastination might lead to a longer and more useful life," the 6th Workshop on Hot Topics in Networks (HotNets-VI '07), 2007.

[20] K. Lu, Y. Qian, D. Rodriguez, W. Rivera, and M. Rodriguez, "Wireless sensor networks for environmental monitoring applications: a design framework," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1108–1112, Washington, DC, USA, November 2007.

[21] G. Halkes, *MAC Protocols for Wireless Sensor Networks and Their Evaluation*, 2009.

[22] S. Farahani, *ZigBee Wireless Networks and Transceivers*, Elsevier, Oxford, UK, 2008.

[23] "Cc2531 usb evaluation module kit," http://www.ti.com/tool/cc2531emk.

[24] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, pp. 6869–6896, 2009.

[25] H. R. Bogena, J. A. Huismana, H. Meierb, U. Rosenbauma, and A. Weuthena, "Hybrid wireless underground sensor networks: quantification of signal attenuation in soil," *Vadose Zone Journal*, vol. 8, no. 3, pp. 755–761, 2009.

[26] Z. G. Kovács, G. E. Marosy, and G. Horváth, "Case study of a simple, low power WSN implementation for forest monitoring," in *Proceedings of the 12th IEEE Biennial Baltic Electronics Conference (BEC '10)*, pp. 161–164, Tallinn, Estonia, October 2010.

[27] L. Selavo, A. Wood, Q. Cao et al., "LUSTER: wireless sensor network for environmental research," in *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 103–116, November 2007.

[28] M. A. Pasha, S. Derrien, and O. Sentieys, "Toward ultra low-power hardware specialization of a wireless sensor network node," in *Proceedings of the IEEE 13th International Multitopic Conference (INMIC '09)*, pp. 1–6, Islamabad, Pakistan, December 2009.

[29] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 181–192, ACM Press, Los Angeles, Calif, USA, November 2003.

[30] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, pp. 65–72, Denver, Colo, USA, April 2005.

[31] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: a hybrid MAC for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 511–524, 2008.

[32] S. Mehta and K. Kwak, "H-MAC: a hybrid MAC protocol for wireless sensor networks," *International Journal of Computer Networks & Communications*, vol. 2, no. 2, Article ID 1087117, 2010.

[33] A. Weddell, M. Magno, G. Merrett, D. Brunelli, B. Al-Hashimi, and L. Benini, "A survey of multi-source energy harvesting systems," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE '13)*, pp. 905–910, 2013.

[34] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[35] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 134–142, Nashville, Tenn, USA, April 2006.

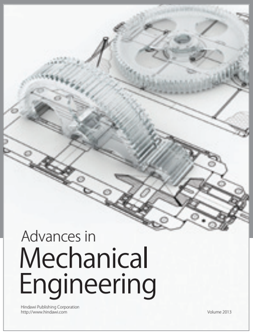
[36] X. Wu and M. Liu, "In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing," in *Proceedings of the 11th ACM/IEEE Conference on Information Processing in Sensing Networks (IPSN '12)*, pp. 1–11, Beijing, China, April 2012.

The Scientific World Journal

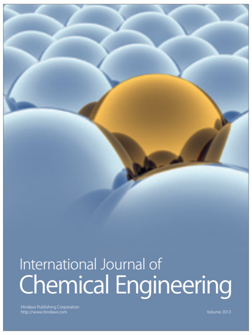
Journal of
Electrical and Computer Engineering

International Journal of
Rotating Machinery

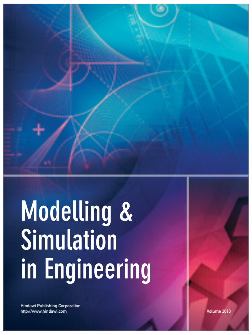
Active and Passive Electronic Components


Advances in
Mechanical Engineering

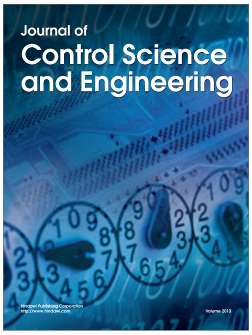
International Journal of
Distributed Sensor Networks

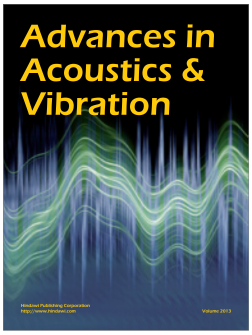

International Journal of
Chemical Engineering

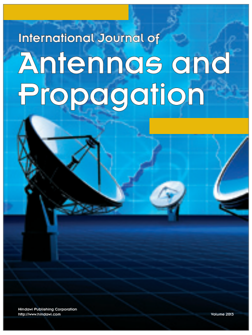
VLSI Design

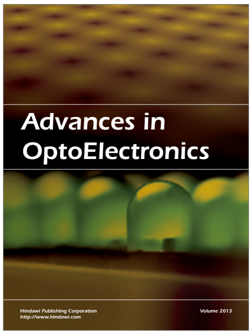

Modelling & Simulation in Engineering

Journal of
Sensors

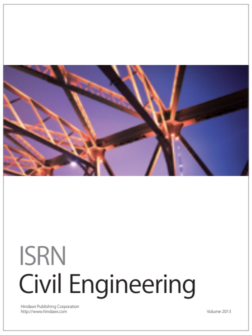

Journal of
Control Science and Engineering

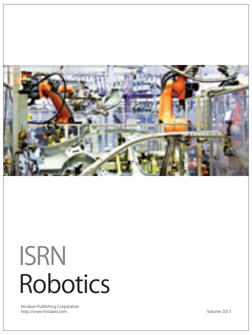

Advances in Acoustics & Vibration

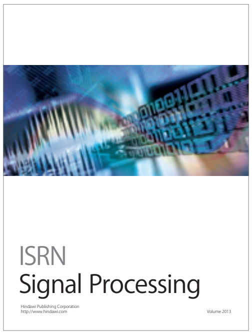

International Journal of
Antennas and Propagation


Advances in OptoElectronics

ISRN
Electronics


ISRN
Civil Engineering


ISRN
Robotics


ISRN
Signal Processing

ISRN
Sensor Networks