

## Research Article

# Power/Energy Estimator for Designing WSN Nodes with Ambient Energy Harvesting Feature

Nicolas Ferry,<sup>1</sup> Sylvain Ducloyer,<sup>1</sup> Nathalie Julien,<sup>1</sup> and Dominique Jutel<sup>2</sup>

<sup>1</sup>Lab-STICC/Université de Bretagne Sud, Centre de Recherche, BP 92116, 56321 Lorient, France

<sup>2</sup>ERYMA Security Systems/Parc Technellys, La Montagne du Salut, CS60009 Caudan, 56607 Lanester, France

Correspondence should be addressed to Nicolas Ferry, nicolas.ferry@univ-ubs.fr

Received 4 June 2010; Revised 13 October 2010; Accepted 7 December 2010

Academic Editor: Friederich Kupzog

Copyright © 2011 Nicolas Ferry et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wireless Sensor Networks (WSNs) consist of spatially distributed autonomous sensors to cooperatively monitor physical conditions. Thus, the node battery autonomy is critical. To outperform it, most WSNs rely on the harvesting capability. As nodes can recharge whenever energy is available, the problem is to determine at design time the node autonomy. For our project, we solve it by creating a power/energy estimator that simulates business scenarios to predict node autonomy; the estimation concerns both power and energy features. Based on node architecture configuration, its Dynamic Power Management (DPM) policy, and environmental conditions, we present a simulator that helps identify power consumption hot spots and make critical choices during the system design. It also helps to scale the energy storage system as well as the energy harvesters correctly. The hardware part is modelled using the FLPA methodology to develop different node component models with a variable accuracy. For the logical part, we developed a specific DPM by integrating meteorology and weather forecast behaviours. The novelty comes from the ability to simulate the WSN harvesting capability and to estimate at runtime the remaining duration of each service.

## 1. Introduction

Wireless Sensor Networks (WSNs) consist of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions [1]. Their target applications are very wide and cover both civil and military domains. Application contexts are various, for example, they can be used in home for earthquake detection, home automation [2], or in the field of automation for parking guidance [3]. The industrial field uses them for monitoring environment (temperature, pressure, hygrometry, luminosity, etc.), while they are used in the environmental field for UV, radioactivity, pollutants, heavy metals, ozone, greenhouse gas, fire surveillance, and high-precision agriculture. The health domain used them for followup patient healthcare [4], the security domain for area or perimeter surveillance, and the field of transportation for automation and accident prevention. In the military domain, WSNs are used in electronic warfare for recognition of troop movement, equipment, and battlefield recognition. They are also used for menace detection of chemical, biological,

nuclear radiation, and for weapon guidance of tactical missiles. However, probably in all domains, the potential of WSNs has not been yet fully explored.

The recent *enhancements* made on new wireless radio frequency transmitters/receivers as well as *technology advances* has brought about several advantages, such as the reduction of the component size and the power consumption. This implies that devices can be more compact, lowcost, and light weight for environment portability [5]. Consequently, nodes aim to achieve maximum autonomy, and this directly relies on the energy storage system [6]. Therefore, global system lifetime and battery autonomy are *critical* (i).

As WSNs are dispersed in the environment, they must be *autonomous*. To increase lifetime, most WSNs rely on their harvesting capabilities, which consist of collecting energy from ambient power sources [7–10]. This solution benefits from “free” environmental energy according to the “green” philosophy, which is to reduce the carbon footprint and to improve reliability of power supply automations. A key feature for evolving in a multisource environmental context is to be *adaptive* (ii).

In WSNs, the power is supplied by an adapted energy storage system [6], like a battery and/or a supercapacitor [11], to handle ambient power. Recharging capability can be serviced by using single or multiple harvesting sources [9, 10]. Harvesting devices range from solar panels [12], wind generators [10] to vibration scavengers [13], and all other devices using a potential energy difference. These devices can provide additional energy to the system, at different energy scales (nW, mW, W), but globally speaking, each source depends on the geographical and environment situation. Thus, the ambient energy profile is highly *variable* (iii).

Solutions are under development to solve these problems. Currently, *Low Power* (LP) design can be used from the beginning and during the entire development process. It is first used to exploit power reduction/optimization techniques to increase system lifetime. This approach is generally classified as *Static Power Management* because it is performed at design time. Secondly, LP is used to adapt the system behaviour to the environment, variable by nature. This is done by using a *Dynamic Power Management* (DPM) [14].

*Static Power Management* can be applied at different layers and on different system subparts. Research explores battery storage systems [6], energy harvesting systems [15], power converters [16], processor units [17], system-level scheduling [18], and other power management opportunities. For hardware components, a difficulty is to extract significant high-level criteria to characterize the power features of the *hardware configuration* (iv).

DPM defines a *Power Manager* to adapt the system behaviour by using different power modes, or by varying the frequency and/or voltage, to dynamically adjust to the system requirements in performance and power consumption. A power manager uses a control procedure based on real measurements or assumptions of the system loads. This procedure is called a *policy*, and it is responsible for applying the *Power Management Strategy* (PMS) (v).

At this step, we can notice that the energy storage system is, from (i), crucial because it handles energy storage and the on-demand supply current. Secondly, the ambient energy is heavily coupled with the environment which is, from (iii), very variable. Low-power approach can be implemented at several levels: at the component granularity level, power management depends on the available power opportunities on the (iv) hardware configuration. At the logical level, DPM defines a strategy (v) to adjust power modes: Idle, Sleep, and Dynamic Frequency Voltage Scaling (DVFS) in real-time (ii) depending on the application tradeoff requirement between performance and power consumption.

For the “CAPNET” project, we must handle realistic business scenarios for the fire brigade of Brest, France. The project goal is to evaluate the gain of migrating from a wired solution to a wireless one. This is why the team must create a software simulator and a hardware demonstrator. The system is an autonomous wireless sensor network prototype using multiple energy harvesting capabilities with a predictive DPM.

Our contribution is to specify and implement an “*estimator*” able to play and record *business scenarios* to estimate

both the power consumption and the autonomy of the nodes. This software will help in evaluating hardware choices by estimating the impact of some component parameters on the power consumption, help in the scale of the battery system and the harvesters accordingly, and help in optimizing the PMS to finely tune the DPM [14]. The key enhancement is to propose an environment prediction model using meteorology to simulate harvesting capability during a scenario. At real-time, integrating *weather forecasts* (WFs) enable us to predict the remaining node autonomy for different DPM services. Therefore, the supervisor user can decide to take control to change the behaviour of a particular node.

To realize the simulator, the approach is to use the *Functional Level Power Analysis* (FLPA) methodology [19]. The method permits the extraction, characterisation, and validation of the power model for each unit of the hardware configuration. One major advantage is the *model adaptive accuracy* which consists of adapting models from generic to complex ones based on physical hardware experiments. Simulation can be processed with more accurate models depending on the required error, accuracy, time, cost, or choices made for the project. This work will help calibrate the production of real-world well-scaled WSNs [2].

The paper is organized as follows: in Section 2, we will present the CAPNET project. In Section 3, we will describe the state-of-the-art about power estimation tools. In Section 4, we will describe the modelling approach and the methodology used to create the WSNs node models. In Section 5, we will detail the power and energy Estimator functional behaviour and its capabilities. In Section 6, we will show model characterisations and results. In Section 7, we will experiment a business scenario to estimate the CAPNET node autonomy, after which we will conclude the paper.

## 2. The CAPNET Project

CAPNET is an autonomous wireless sensor network prototype using multiple energy harvesting capabilities. Its objective is to study and implement a simulator and a demonstrator that show the interest of the wireless approach over the wired one. The project is expected to reduce the cost by a factor of ten over the previous technology. Nodes must support a recharging feature by harvesting multiple environmental power sources. The specification includes special aspects such as an integrated alert system able to warn a supervisor in less than one second. The WSN uses a linear topology ranging from 100 metres to 2 kilometres. Each node communicates from/to direct neighbours. The nodes are spaced at intervals of 100 metres and support a bidirectional data transport protocol. If a node presents a dysfunction, the system is able to detect and reconfigure itself to automatically skip one (or up to two) dysfunctional nodes. The detection occurs when a timeout is triggered due to an unresponding node. Then, the node increases the amplifier range to reach the next node distance. WFs are transmitted to the nodes in order to switch the battery charger to the maximal available power source. WFs can also inform the supervisor of the remaining node lifetime for each DPM



FIGURE 1: CAPNET topology.

service. We have integrated the WF into the specific DPM called the WF-DPM.

**2.1. Applications.** The network topology is linear as shown in Figure 1. This topology has been selected due to the project applications. This choice guarantees an optimized alert timing for all the nodes from the farthest to the nearest.

The system must handle various applications like fire services, frontier and perimeter surveillance, (water, gas, and petrol) network monitoring, environment observation (earthquake, water overflow, and meteorology), automatic meter reading, road network surveillance, industrial control command. Customers provide representative use case scenarios corresponding to each application. The representative scenario presented here has been established in cooperation with the fire brigade of Brest, France.

**2.2. Scenario.** A problem appears on a carrier transporting highly dangerous freight. A fire brigade must help the carrier docked at the port of Brest, France. Products are flammable and can reject toxic gas into the town atmosphere. It is very threatening due to the presence of heat, toxic, or explosive gas thus presenting a risk of explosion. To monitor and prevent the risks, a secure perimeter is defined in an area range of 300 to 500 metres. All people entering this perimeter send an alert signal to the supervisor who can view snapshots or video of the scene. The WSN equipment is deployed in the area in a circle layout. This is to intercept the gas, so as to avoid spreading. For this mission, nodes embed traditional meteorology sensors and also special gas detectors, a toxicity meter, and for special nodes a thermal camera to track hotspots. The system must be able to work continuously outside and in all types of weather.

**2.3. Contributions.** In this project, our work is to specify and realize a power and energy estimator that will permit the playback of real-case scenarios. Our needs are to predict the remaining node autonomy with harvesting capability. But estimating the power peaks is also necessary for the battery model accuracy. The simulator results provide the total node

autonomy and the different energy consumption for different groups of internal components. Moreover, by integrating WF into the DPM, we can predict the Ambient Potential Power Value (APPV) and then give the maximal time duration per service. Therefore, the simulator can estimate the power and energy consumption for each component according to their control attributes, the weather environment, the energy provision, the temperature degradation, and the power management strategy (PMS).

To do this, we use a modelling approach by applying the FLPA methodology (see Section 4.1). First, we establish the global node power model that describes the entire simulator (see Section 4.2). Secondly, we define a PMS which can switch the power source depending on WF predictions (see Section 4.3). Then, we build different models based on real power consumption measurements for each device (see Section 4.4). Finally, we build the estimator able to playback the scenarios and give the total energy consumption per node (see Section 5). When manufacturers build the final hardware demonstrator, we will be able to compare the absolute error between the simulator and the demonstrator. This will give us feedback on the behaviours of the real WSNs and the simulated WSNs to be able to enhance models if required to better fit to the reality.

Before beginning, we will go over the state-of-the-art of WSN power estimation tools.

### 3. State-of-the-Art

Low-power design could be applied at several levels in electronics and WSNs. Some books are good starts for understanding low power methodology [20], system-level optimizations [21], and algorithms for power saving [22]. If we focus more particularly on WSNs, interesting research works have detailed specific techniques. We could cite battery modelling [23], Maximum Power Point Tracking (MPPT) optimization [16], DPM [14], vibration harvesters [13], and network modelling [24] to design a WSN [2, 25]. All these technological headways are coupled with constant new WSN prototypes. We present in this section, some interesting WSNs, focussing our attention on their features.

Helimote [8] has a solar panel and two AA Ni-MH batteries for a 3 V current output. Its Solar panel is directly connected to its battery through a diode, so energy harvesting starts after the solar output is higher than 0.7 V than the battery voltage. Helimote does not perform MPPT to achieve better harvesting efficiency. Consequently, the solar panel is voluminous to obtain a higher voltage than the battery. Helimote uses an ATmega 128 L with a Chipcon 868/916 Mhz transceiver.

Prometheus [7] uses a solar panel to charge a 22 F supercapacitor from which the system draws current when there is enough power. The system uses the secondary buffer Li-Polymer battery when the charge of the supercapacitor is lower than a threshold. Also, when the supercapacitor is higher than a given threshold, it charges the battery. However, like Helimote, the solar panel is directly hardwired to the supercapacitor and does not harvest energy efficiently. All control tasks are run on a MSP430 MCU. Prometheus TelosB



uses a TI MSP430 with a Chipcon 2240 2.4 Ghz Wireless transceiver. Prometheus power consumption is estimated ranging from  $205 \mu\text{W}$  at 1% duty cycle and 20 mA at 100%.

Everlast [11] has only a solar panel and a supercapacitor, but no battery. Thus, it does not suffer from battery degradation. It embeds an MPPT circuit for optimal power harvesting that charges a 100F supercapacitor. However, it requires MCU to perform the MPPT algorithm calculation. Everlast uses a PIC16LF747 coupled with a Nordic nRF2401 2.4 Ghz Transceiver. Everlast power consumption is 0.68 mW in minimum mode and 52.4 mW in maximum mode.

PUMA [9] has multiple power sources. It uses a power routing technique to harvest simultaneous multiple power sources. The high energy harvesting is achieved with a combination of MPPT and power defragmentation. However, it requires MCU control based on input from light and wind sensors. PUMA uses a PIC18F6680 with a wireless 802.11b card. Duranode consumption is 0.5 W in minimum mode and reaches 2 W in maximum mode.

AmbiMax [10] is a multiple harvester system with solar and wind experiments. It includes MPPT tracking using an hysteresis comparator coupled with an additional sensor to hardware calculated MPPT. The harvested power is sent to a boost regulator before charging a Li-Polymer battery. Ambimax charges 12.5 times faster even at lower supply levels, where previous techniques stop. Ambimax uses a MCU8051 and an integrated 2.4 Ghz Radio. Its power consumption is less than 200 mW.

PicoCube [5] is the smallest WSN known. It is a  $1 \text{ cm}^3$  sensor node powered by harvested energy. It uses a MSP430-F1222 with a FBAR custom RF transceiver, the power consumption is  $6 \mu\text{W}$  in mean and reaches 6 mW when transmitting. More WSNs with their technical features are referenced at [26].

Now, if we examine available tools, there are several power consumption platforms which can analyze systems at specific power levels [27]. As can be seen in Table 1, some tools can predict the power consumption of microprocessors, while others can predict the overhead of an embedded OS. We have not included the works about WSNs simulation which mainly simulate network consumption by RF protocol, transport, and so forth. The reason for this is that in this project, the RF simulation has been given to a partner lab.

Our need is to predict the remaining node autonomy with harvesting capability. In Table 1, most of the tools focus on particular microprocessors [30–32] or hardware platforms and do not consider the entire node [30–32, 34, 36]. Building the whole simulation from scratch at transistor level with SPICE [28] would be definitely a hard job. Also, most WSNs include a DPM which must command/control power modes, shutdown devices, apply tasks scheduling, and so forth. Actually, most tools do not model the DPM behaviour and are more hardware oriented [37, 38]. New research projects are trying to model the WSNs in the whole by using macromodelling [39], but these modelling concerns simple WSNs. Another discrete-time analysis framework [40] and advanced commercial low-power synthesis tool [35] exist, but these tools are not adapted to model harvesting WSNs.

TABLE 1: Power estimation tools.

Power tool	Platform	Analysis level
SPICE [28]	General-purposes circuit design	PCB/Transistor
Simbed [29]	Motorola M-CORE platforms	Architectural
Skyeye [30]	MIPS platform	Instruction
eSimu [31]	ARM9 platform	Instruction
EMSIM [32]	StrongARM platform	Instruction
Softwatt [33]	Embra, Mipsy, MXS platforms	OS
powerTOSSIM [34]	TinyOS simulator on Mica2	OS
PowerOpt [35] ChipVision	Low-power system synthesis C/SystemC to VerilogRTL	Architetur + instruction
SoftExplorer [36]	CPU power Estimation tool	Functional + instruction
Interconnect Explorer [37]	Interconnections tool	Architectural
C.A.T [38]	Power estimation platform	All

Thus, to simulate an entire WSNs node, we must care about other internal node components like DC/DC, harvesters, microprocessors, radio, or sensors but also the environment and the DPM policy. And currently, none of the tools responds completely to our needs. This is why we want to develop a generic tool which can include specific models into a single estimator.

A generic WSNs power estimator is required to verify if some choices are relevant and contribute to system consistency. Recording and replaying scenarios permit to experiment consumption evolution by varying some parameters of the model. Although some components have already been chosen and can clearly be measured to provide an accurate model, other components are not, and we must rely on generic models. Consequently, the two approaches must be considered depending on the components which have been selected. The global modelling approach is described in the next section.

## 4. Modelling Approach

We make the assumption that all nodes can be modelled independently. We use a decomposition approach that divides each element of the system. Thus, we assume that a node is decomposed into several internal components. We apply an existing methodology to model elements of the WSNs node architecture, called the FLPA methodology.

**4.1. FLPA Methodology.** The *Functional Level Power Analysis (FLPA)* methodology [19] can be adapted to any component and has already been successfully applied to extract processor [41] and FPGA [42] power consumption models. This methodology is based on physical measurements in order to guarantee realistic values with good accuracy.

TABLE 2: Scenario model example.

Name	Nature	Type	Start	Period	Dur.	Rep.
Gas measurement	Auto Measures	Periodic	2	10 min	2 s	500x
Intrusion area event	Alarm Niv1	Periodic	1	300 min	10 s	32x
Intrusion confirmation	Alarm Niv2	Periodic	3	2 hours	4 s	10x

The FLPA methodology has four main parts, which are given below.

- (1) A primary *functional analysis* describes the system as a grey box with high-level parameters having an effective impact on the power consumption.
- (2) The *power characterisation* step explains the power consumption behaviour (obtained by physical measurements) when each parameter varies independently.
- (3) Then, the complete *power model* is created with a curve fitting or regression approach; it expresses the overall power consumption variations related to all the parameters.
- (4) Finally, the *model accuracy* is estimated from a new measurement set which provides the absolute error.

With this methodology, it can take less than a month to build a power model of a complex processor. At project startup, all components or decisions have not been validated yet, so we define a variable accuracy granularity depending on project advancement. Thus, model accuracy evolves during project maturation by evaluating the cost to develop a more accurate model. Two cases are used.

- (i) If the physical element is *not available*, we search for an existing model (datasheet, analytical law, etc.). Then, we build a generic model which is integrated into the simulator.
- (ii) If the physical element is *available*, we proceed with the FLPA methodology to extract an efficient model.

**4.2. Global Node Power Model.** We start by defining, in Figure 2, what will be the global node model.

The node model is composed of two layers: the logical and the hardware layer. Both are tightly coupled and are synchronized to work together. This model defines multiple models for the DPM and the hardware configuration.

**4.3. The Logical Layer Power Model.** The logical level defines the node environment as several external models: scenario events, RF events, meteorology, and weather forecast models. The DPM model describes the energetic strategy developed on the node.

**4.3.1. Scenarios.** In Table 2, scenarios are composed of rule-based events which control the entire sequence. Event

TABLE 3: RF model example.

Time	Supervisor	Node 1	Node 2	Node X
$t_0$	Tx_ReqData			
$t_1$		Rx_ReqData		
$t_2$		Tx_ACK	Rx_reqData	
$t_3$	Rx_ACK	Tx_Measures	Tx_ACK	Rx_reqData
$t_i$	...	...	...	...

signals define the start time, the duration, the periodicity for periodic events, and the numbers of repetitions. This produces a signal that characterizes an event context like a sensor notification or other particular behaviour (fire explosion, secure area intrusion, etc.). Events are consumed by the DPM which verifies the rules at each discrete time and consequently performs the corresponding actions. Basically, the DPM impacts the devices and changes their model values on the fly.

**4.3.2. RF Events.** The RF model describes the communication protocol as a set of RF events. Events define the incoming and outgoing packets exchanged by the node. The RF model is provided by our partners working on the RF modules and the RF simulation. Table 3 presents the RF matrix model. Here, the time resolution is in milliseconds. The energy is estimated by cumulating over a one-second period the time spent in each modes (RX, TX, Sleep, or OFF) with their equivalent power model attributes. The model will be refined when the RF partner transmits the final model.

**4.3.3. Meteorology.** Since energy is scavenged from the environment, the node model must include a model for each power source type. These environmental sources can be provided in multiple ways including solar, wind, thermal, radio, vibrations, and mechanical constraints. Currently, CAPNET focuses on solar, wind, and thermal power sources. Meteorology context is constructed with the Meteo-builder integrated with the simulator. The context can be created manually, scanned from a website, or extracted from a meteorology file recorded day after day. The Table 4 presents a model example for the city of Lorient, France.

With Date/Time: the timestamp,  $T$ : the temperature (in °C),  $W_{dir}$ : the wind direction (in °),  $W_{spd}$ : the wind speed (in Km/h),  $R$ : the rain measurement (in mm),  $H$ : the humidity degree (in %),  $P$ : the atmospheric pressure (in hPa), and  $S$ : the sunset irradiance (in W/m<sup>2</sup>).

TABLE 4: Meteorology model example.

Date/time	$T$	$W_{dir}$	$W_{spd}$	$R$	$H$	$P$	$S$
30/06/2010 11:00	19	262	10	0	74	1022	229
30/06/2010 14:00	22	251	10	0	66	1021	228
30/06/2010 17:00	22	241	15	0	64	1020	275
30/06/2010 20:00	21	241	15	0	67	1019	45
Date/time <sub><i>i</i></sub>	—	—	—	—	—	—	—

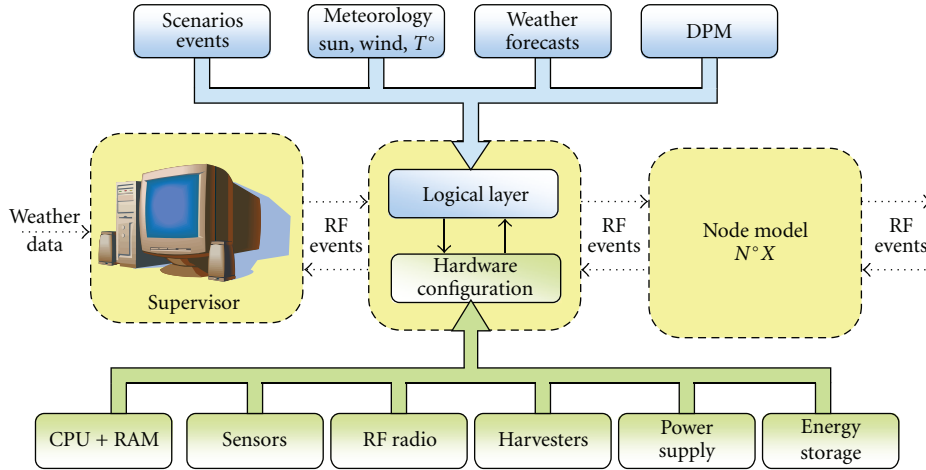


FIGURE 2: The CAPNET node model.

**4.3.4. Weather Forecasts.** The particularity of this project is the modelling of not only the meteorology but also the environment by exploiting WF. As meteorology forecasters have already done heavy, intensive computation to predict the weather, we reuse standard models provided by GFS runs [43], and Meteo-France runs [44]. Therefore, the error of these models is propagated directly from the predictions. This model particularity comes from the real-time update of the data models and their granularity. The node power management strategy can use these predictions on 7-day forward moving windows and combines backward windows based on real weather measurements. The WF model is similar to the Meteorological one, but with forward time stamps.

**4.3.5. DPM.** The DPM policy controls the different power states enabled for the system. This model is not the microprocessor program but mainly a subset from the energy/power point of view. The DPM has been modelled as a Finite State Machine (FSM) composed of a set of states and transitions. Each state represents a particular system consumption state (IDLE, Sensor1 Active and Sensor2 inactive, etc.).

This DPM model introduces the capacity to exploit the WF model. The nodes have the ability to switch their power source to the most advantageous energy microsource. The system uses a hardware node power switch like DuraNode [45], but here, WFs are used to determine which power harvesting source will have the highest energy availability. WFs can also be used to deactivate devices, like the camera,

if the energy level required is too low and the WFs do not predict enough ambient energy.

The DPM process works as follows: DPM process starts by reading external information: scenario and RF events, which act as stimulus on it. They update DPM control parameters and configuration device states. Then, the DPM combines current state, power status indicators, and weather forecasts predictions to decide transitions to another power state. Each power state defines a functional power mode to apply to components: CPU modes, frequency, clock gating, shutdown, RF amplification level, and so forth. Thus, the DPM updates configuration parameters before the estimation of the new hardware model values.

The DPM shown in Figure 3 defines the power management strategy for person detection in the intervention area.

After initialization, the system enters the Sleep processor state; only Sensor1 is active. When Sensor1 generates an alarm, Sensor2 is powered up to verify the information. If Sensor2 confirms the alarm, an alert is sent to the supervisor. The supervisor can decide to power up special devices or to start the camera mode to take a video of the scene. The video is in fact several JPEG snapshots taken one by one and sent to the supervisor. In intervention, the supervisor can decide to stop or power down devices on demand. When both alarms are, stopped the system returns to Sleep mode.

The DPM service duration estimation method is done by underlining paths in the DPM.

Here, we can extract four main paths corresponding to a service. The system's *Sleep* mode is the first service with no

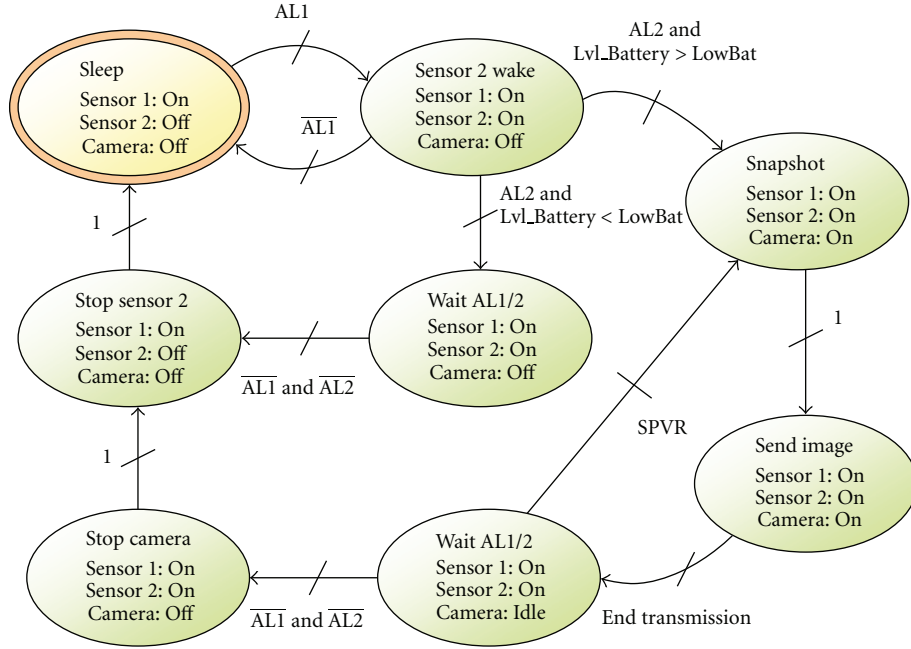


FIGURE 3: The CAPNET DPM model.

activity. The second service is *S2Verif*; in this path, the system detects an alarm with the Sensor1. This wakes Sensor2 up but Sensor2 does not detect a problem. So, after a certain amount of time, the system goes back to Sleep mode.

In the third service, Sensor1 detects an alarm and wakes up Sensor2. Sensor2 detects an alert and sends an alarm level 2. If the battery is under the low state threshold, the system enters into the *S2LowBat* service, sending alarms and waiting for alarm releases.

Finally, the fourth service is the same as the third service except that the battery is above the low state threshold. Then, the camera is turned on, and the system takes a snapshot, sends it to the supervisor, and enters into a waiting state. Depending if the supervisor has activated the SuPervisor Video Request (SPVR), the node can send a video by entering a loop. Waiting stops when the supervisor confirms the off state, and both sensors have been released. The system then switches the camera and sensors off before returning to the *Sleep* mode.

The service duration  $S_d$  can be expressed as

$$\text{Coef} = \frac{\text{Bat}}{\sum_{i=1}^n P_{t_i} \cdot T_{t_i}} \quad \begin{array}{l} \text{with } P_{t_i}: \text{task power in } W, \\ \text{and } T_{t_i}: \text{task duration in } s, \end{array} \quad (1)$$

$$S_d = \text{Coef} \cdot \sum_{i=1}^n T_{t_i} \quad \begin{array}{l} \text{with } S_d: \text{service duration in } s, \\ \text{and } T_{t_i}: \text{task duration in } s. \end{array}$$

For example, by taking a 12 V, 2 A h, 86400 Joules Battery  $P_{t_1} = 0.3$ ,  $P_{t_2} = 0.653$ ,  $P_{t_3} = 0.653$ ,  $P_{t_4} = 0.353$ ,  $P_{t_5} = 1.2$ ,  $P_{t_6} = 1.4$ ,  $P_{t_7} = 1.2$ ,  $P_{t_8} = 0.653$  in Watt and  $T_1 = T_2 = T_3 = T_4 = T_5 = T_6 = T_7 = T_8 = 1$  second, we have

$$(i) \text{ Sleep: } 0.300 \cdot T_1 \geq \text{Sleep maximal duration is 80 h.}$$

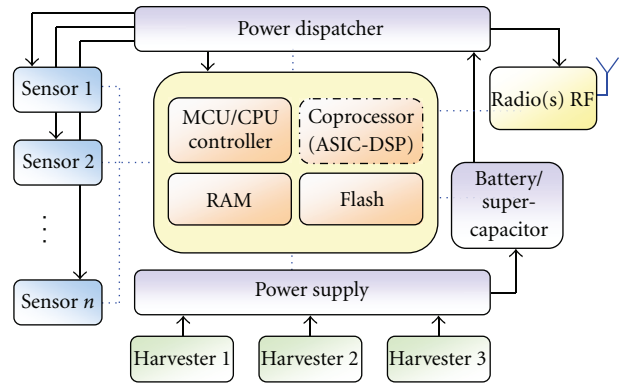


FIGURE 4: The CAPNET hardware node model.

$$(ii) \text{ S2Verif: } 0.300 \cdot T_1 + 0.653 \cdot T_2 \geq \text{S2Verif maximal duration is 50 h.}$$

$$(iii) \text{ S2LowBat: } 0.300 \cdot T_1 + 0.653 \cdot T_2 + 0.653 \cdot T_3 + 0.353 \cdot T_4 \geq \text{S2LowBat maximal duration is 49 h.}$$

$$(iv) \text{ Video: } 0.300 \cdot T_1 + 0.653 \cdot T_2 + (1.2 \cdot T_5 + 1.4 \cdot T_6 + 1.2 \cdot T_7) \cdot \text{NbSnapshots} + 0.653 \cdot T_8 + 0.353 \cdot T_4 \geq \text{Video maximal duration is 18.94 h with 22736 snapshots.}$$

The supervisor can take control of the node whenever they want, forcing it to use a special service. With these report indicators, they are alerted of the maximum service duration and can take adapted decisions.

**4.4. The Hardware Configuration Power Model.** The hardware model is basically extracted from the main system hardware components. Figure 4 shows a hardware node composed of control units, a storage energy system, a sensor



set, a harvester set, power supply converters, and radio transceivers.

From this, the FLPA methodology is applied to each component. This helps the designer to define high-level parameters that characterize the power consumption and decrease the complexity by splitting the node into elements. Each model parameter is initialized in the configuration according to the respective datasheets and the simulation requirements.

**4.4.1. Sensors.** Each sensor uses specific power modes, power supply, and quality of service parameters, which are modeled and controlled by the power management. However, sometimes real measurements could reveal datasheet distortions due to a mean function or different experimental conditions. Therefore, datasheet models must be validated before using them. At runtime, the node can also integrate models and, therefore, must be adapted to run on a specific platform. For example, some complex models cannot be evaluated on a low-performance microcontroller because of RAM size or performance constraints. This is why most models must be adapted dynamically with FLPA if they are integrated into the final platform.

**4.4.2. Harvesters.** Each harvester system is principally modelled by determining a standard *I-V characterization* (intensity versus voltage); therefore extracting the maximal *P-V curves* (power versus voltage) point law associated to it. Thus, a *Maximum Power Point Tracking* (MPPT) could provide the best power adjustment [16] for the harvesting device. Harvesters input parameters are supplied by the meteorology model.

**4.4.3. Microprocessor/RAM.** Microprocessors often offer low power modes or a Dynamic Frequency and Voltage Scaling (DVFS) ability [46]. Microprocessor model is obtained by using FLPA measurement methodology on real chips programmed to play the DPM. The CPU model defines multiple power levels (Normal, Idle, Sleep, and Doze modes) and switches between power modes depending on the DPM activity. If a more complex CPU model is required, we can use the SoftExplorer [36] tool to extract more accurate model coefficients. If external RAM is present, the model has the ability to shutdown the RAM when in low-power mode. An enhancement could be to include OS task scheduling activity for anchoring DVFS.

**4.4.4. Radio Modules.** Statistically, the radio device is the most power-hungry component. The entire power consumption modelling is delegated to RF experts of a partner laboratory. Radio is modelled using their different power states and activity (RX, TX, Forward, idle, sleep, etc.). Power modes and timings are integrated as input values in the simulator. Of course, RF Scenarios and PE scenarios must be synchronized to describe the same reality.

**4.4.5. Power Supply/Power Dispatcher.** The goal of the power supply unit is to store energy harvested from the environment into the battery. The power dispatcher splits system

power supply into several voltages for each component specification. Both use mainly DC/DC converters to step down or step up the voltage to provide regulated voltage. Converter efficiency is variable, and low leakage current can appear. It depends on the manufacturer, the temperature, the input voltage, and the output current.

**4.4.6. Energy Storage System.** One of the most important models is that of the energy storage system. The battery behaviour is by far the most complex [47]. Most battery models [23] predict the discharge time with a constant current, but the system uses environment harvesting technologies and can recharge itself while discharging. Therefore, we use a battery model which handles both charge and discharge variable currents. This one is similar to the model developed by [48], but we use different curve fitting mathematical equations. As a battery presents poor reactivity to pulses, we note that recent works on *supercapacity* [49] also show good alternatives to assist main battery systems for harvesters [7, 10].

Now that the node model context has been presented, we will detail the power and energy estimator itself which exploits the models described in this section.

## 5. The Power and Energy Estimator

The goal of the simulator is to help at design to identify major *power consumption hot-spots*. As the project is constantly evolving, models are dynamically revised with the FLPA process depending on the desired accuracy. Thus, the simulator can estimate both the node power consumption and predict the system autonomy. The simulator accuracy mainly depends on model specific errors.

On the other hand, simulating a DPM permits one to adjust the strategy and fine tune DVFS and DPM opportunities. Thus, designers can evaluate the relevance of the developed DPM policy. Then, finally, we can give a nearly realistic estimation of the global system viability for a configuration.

Figure 5 presents the prototype of the Power/Energy Estimator (PEE). In the main part, the screen describes the node major hardware components, specific to studied WSNs. The right part includes the controls which command the system parameters in manual mode. Controls are updated in real-time in automatic mode, when playing a scenario. The bottom region displays the energy storage system evolutions during the simulation.

**5.1. Simulator Implementation.** The PEE is built with Labview from National Instruments [50]. It combines a graphical language with massively parallel programming and object paradigm support.

The PEE is based on the global node model presented in Figure 2. Thus, the simulator engine is composed of the two layers. First, the power management policy layer reacts and describes DPM activities. Second, the physical layer includes all the hardware component models.

The DPM layer models the PMS by using an FSM defined by a states-transitions schema. The DPM follows the scenario



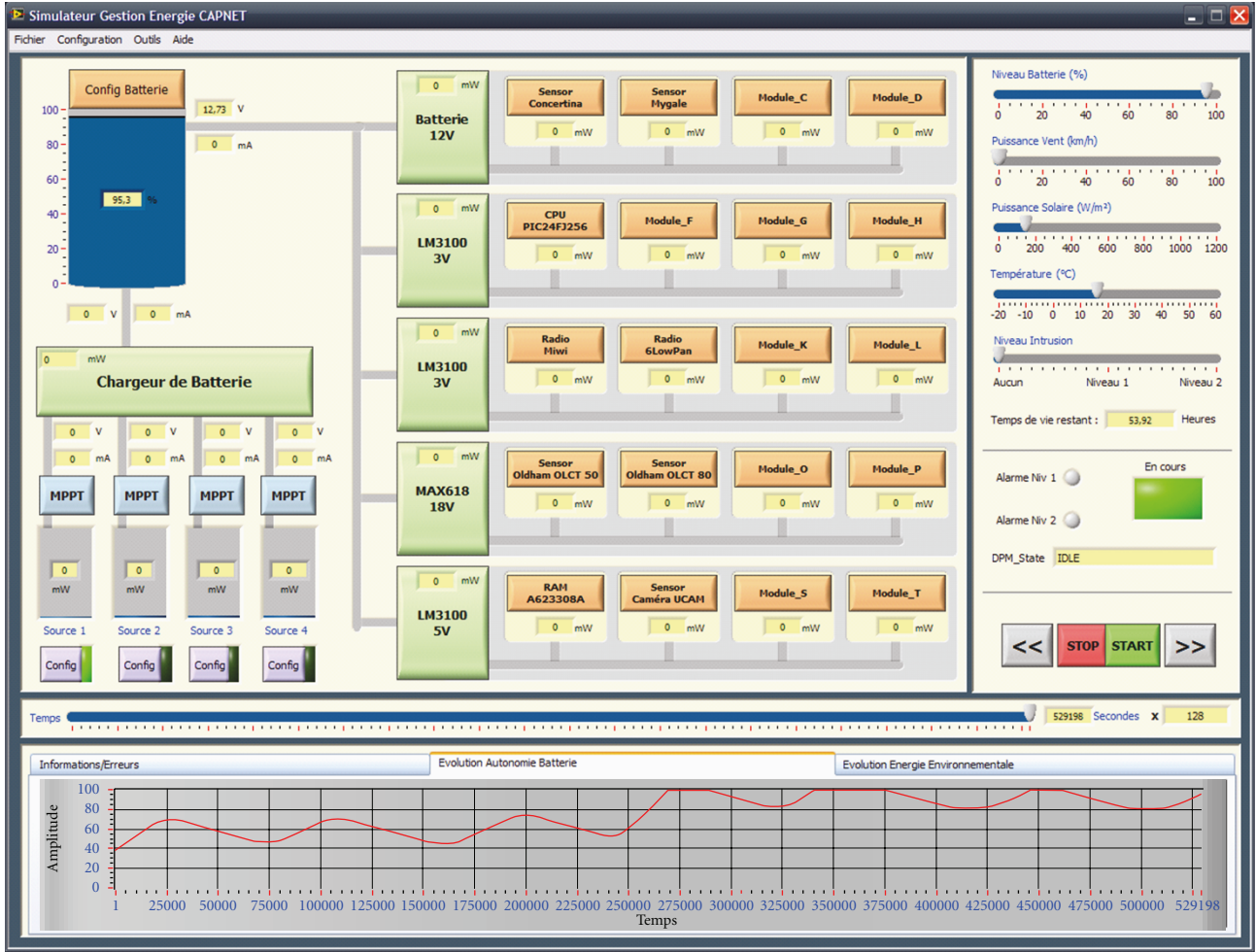


FIGURE 5: Power and energy estimator.

which defines the interaction with the internal and external events. The policy decides on the behaviour to adopt, and it updates the device states/parameters accordingly in the hardware configuration.

The Hardware layer is based on node components and their characteristics. Each component or peripheral is called a device. Devices are classified according to their types: DC/DC converters, batteries, microsources, CPU, radio, and sensors as specified by the hardware model. Each device is modelled by a class specialization that defines the device behaviours in terms of power consumption. A configuration is a set of channels on which devices are mapped. Each device class embeds the properties, the parameters, and the methods to evaluate and update its model.

The power simulator is managed by the “manager” which commands the entire simulation evolution. The manager controls playing/recording of scenarios by using a discrete time representation which divides time into time-slice quantas. The manager task is to apply the simulator process which synchronizes both layers.

**5.2. Simulator Process Description.** The PEE process is shown in Figure 6. The user creates or loads a project which includes

the scenario to be studied. In fact, it is the definition of the different models, the configuration, the global simulation parameters, and the results destination.

When the user decides to start the simulation, the manager launches the simulation at a defined temporal resolution. When playing, the real-time DPM module (RT.DPM) calls at fixed frequency the sample DPM function. This function processes the reading of the node external models and collects the data to dispatch them to the appropriate devices. In detail, Read\_Scenario reads the scenario events. Read\_Meteo reads the meteo coefficients and the WF predictions. Read\_RF reads the RF environment to collect the radio packets. Dispatch\_Events applies the causality part of events. For example, when the presence of gas is defined by an event, the simulator must update TX-12 gas sensor value accordingly. This step updates the input parameters and thus prepares the hardware model. In practice, Dispatch\_Events mainly updates sensors, but events can potentially have an impact on any parts (cause of failure, etc.). The Dispatch\_Meteo prepares and converts meteo data for the harvesters and the global node temperature (because models can integrate model temperature degradation).

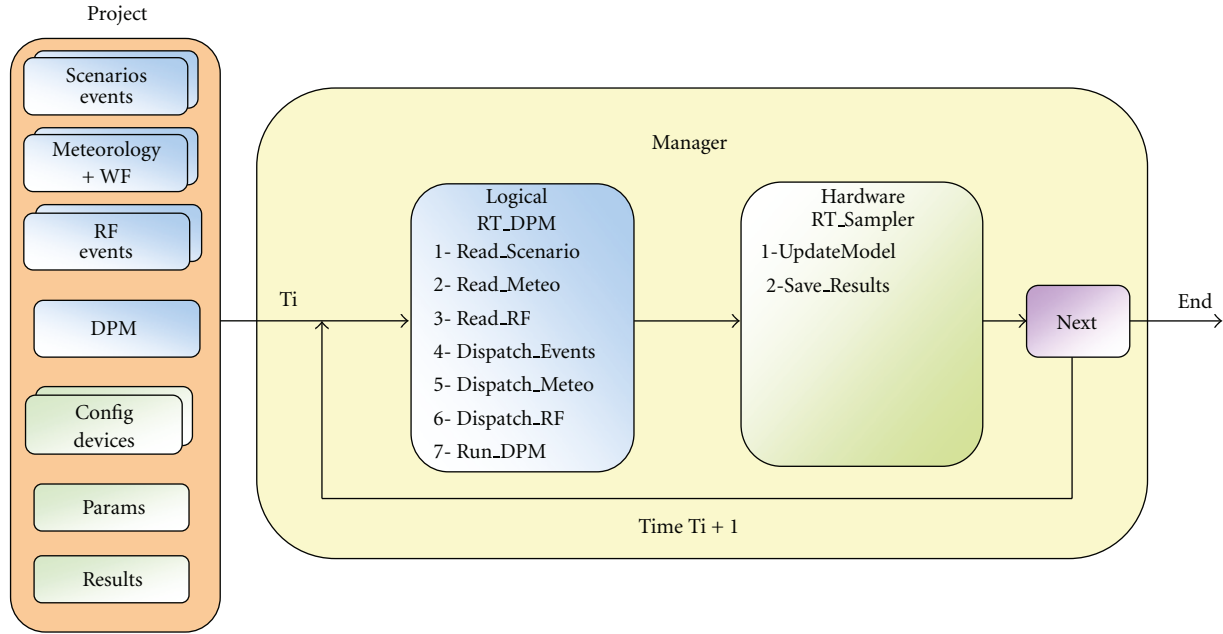


FIGURE 6: Simulator process.

WF information is transferred to the DPM. Then, Dispatch\_RF sends the packets to the appropriate hardware radio module. The logical stage purpose is to collect and prepare the input parameters, to run the DPM module. At this time, no hardware models have been estimated yet. Finally, the run\_DPM method is called. The DPM solves the transition conditions and enters a new power mode. The DPM executes a behaviour resulting in the update of command variables and hardware component parameters.

The real-time sampler module (RT\_Sampler) is started after the DPM. Firstly, the sampler evaluates each device model with the new fixed input parameter values. This is done by calling the redefined UpdateModel method which reevaluates the model. Each model has dedicated parameters but share common variables: the supply voltage indicator, the current/intensity, the consumed power, and the component energy magnitude value (different for each sensor or microsource). Secondly, these output variables are recorded as one goes along in the results destination.

When RT\_DPM and RT\_Sampler have terminated, the next function is evaluated to determine if simulation time is exceeded or battery level has reached a low threshold. If simulation continues, the simulation time is incremented and the process restarts in loop.

**5.3. Simulator Results.** When the simulation is terminated, the result files contain data over time which can be directly analysed to present the results. Results are grouped into sections: sensors, microsources, MPPT, DC/DC, CPU/RAM, and radio components. For each component, the simulator provides the equipment value, the intensity, the voltage, and the power curves. Thus, the maximum power peaks can be used to study the impact on the battery for high-discharge sequences. The total harvested energy and the total

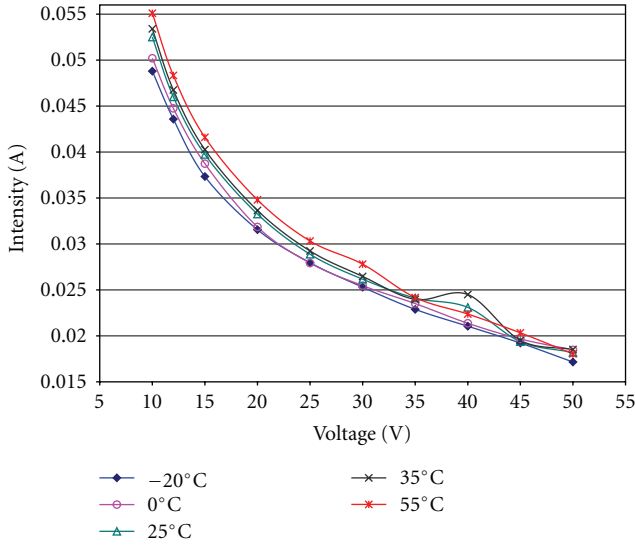
consumed energy are also determined by integrating the power over time. The energy directly relies on the node autonomy which can be presented in Watt hours. Because batteries have specific behaviours [47], we cannot model the total node autonomy with basic sum operators. Instead, we must rely on a realistic battery model that handles at realtime simultaneous variable charge and discharge currents [48, 51].

## 6. Model Characterization

For each model, we applied the FLPA methodology described in Section 4.1. In this section, we present the results for two sensors, a camera, a solar microsource model, and the battery.

**6.1. ERYMA 4S Sensor.** The ERYMA 4S sensor is an intrusion detector based on ground vibrations. A walking person produces vibrations that are recorded and transmitted to a local unit. This unit analyses the measurements and generates an alarm. Power supply ranging from 10 to 50 Volts. Functional temperature range is  $-40^{\circ}\text{C}$  to  $+65^{\circ}\text{C}$ . For this power experimentation, we applied the FLPA methodology to measure and characterize the sensor. For the first step, we proceeded with a series of measurements by varying the sensor input voltage. These measurements give the intensity drained by the device. The process has been repeated for several temperatures:  $-20$ ,  $0$ ,  $+25$ ,  $+35$ ,  $+55$  in  $^{\circ}\text{C}$  by using an incubator. We have obtained a series of  $I$ - $V$  measures presenting the  $I$ - $V$  characterization curves for different temperatures shown in Figure 7.

We notice that the device seems to have a particular behaviour at 40 V. This is probably due to a calibration effect. As we know, the power is given by the formula:  $P = U \cdot I$ ,

FIGURE 7: 4S sensor  $I$ - $V$  characterization.

so it is trivial to determine the sensor power consumption. Thus, we calculate the power consumption for each curve. This gives the consumption for a selected voltage and an operating temperature.

For the 4S sensor, power consumption is the lowest at minimum temperature ( $-20^{\circ}\text{C}$ ). We note that the power consumption increases from 525 mW at 10 V to 910 mW at 50 V. It is a 73.33% increase at  $25^{\circ}\text{C}$ . We notice that there is low power variation as the temperature increases. The maximal variation is 11% when the temperature rises from  $-20^{\circ}\text{C}$  to  $+55^{\circ}\text{C}$  at 10 Volts.

With these measurements, we extract the characteristics for which the sensor is the less consuming. We get the equation model (2)

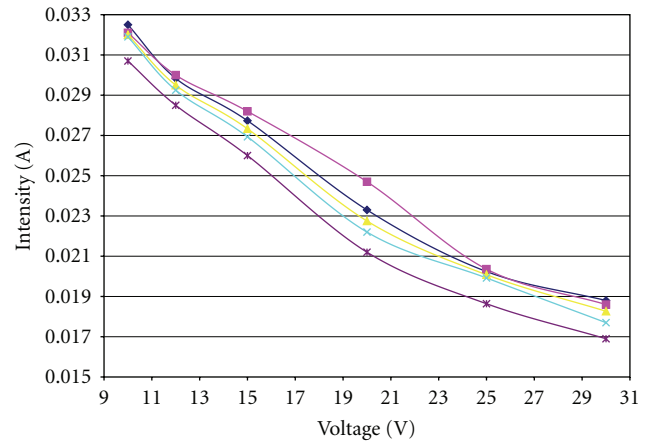
$$y = 0.2041 \cdot x^{-0.6209} \quad \text{with } x \text{ in V, } y \text{ in A.} \quad (2)$$

As can be noticed, the curve variation is nearly linear in temperature. Thus, a linear variation coefficient can be defined.

Once the final model is built, the absolute error is measured. For this sensor, the mean error is 1.7%, and the maximum peak error is 11.9%. As the sensor will probably be used in 10–15 V like the battery nominal voltage, the consumption will range from 488 to 624 mW. The maximum error is 1.74%, this corresponds to less than 1 mA error from the real behaviour.

**6.2. ERYMA S912 Sensor.** The ERYMA S912 sensor is an intrusion detector based on fence vibrations. A person climbing a fence is detected and a signal is sent to the local unit. The unit analyses the measurements and generates an alert.

Like the previous sensor, we measure the behaviour of the S912 sensor located in an incubator at:  $-20$ ,  $0$ ,  $+25$ ,  $+35$ ,  $+55$  in  $^{\circ}\text{C}$ . We vary the sensor input voltage from 10 to 30 Volts, and we record the intensity drained by the device.

FIGURE 8: S912 sensor  $I$ - $V$  characterization.

We have obtained a series of  $I$ - $V$  measures presenting the  $I$ - $V$  characterization curves for different temperatures shown in Figure 8.

We notice that the power consumption increases as the input voltage grows, from 320 mW at 10 Volts to 548 mW at 30 V. This is an increase of 71.25% at  $25^{\circ}\text{C}$ . The power consumption has a 14% decrease when the temperature evolves from  $-20^{\circ}\text{C}$  to  $+55^{\circ}\text{C}$ . We can remark that the curve fits near a log function modelled by (3)

$$y = -0.013 \cdot \ln(x) + 0.0607 \quad \text{with } x \text{ in V, } y \text{ in A.} \quad (3)$$

Moreover, we can remark that the increase of temperature causes a power consumption variation from 4% to 14% depending on the voltage. This increase can be modelled as a linear equation.

Once the final model is built, the measured absolute error is 1.9%. The maximum peak error is 4.7%. The sensor will probably be used at 10–15 V like the battery nominal voltage. In this case, the consumption will range from 307 to 423 mW. Here, a higher temperature gives lower consumption.

**6.3. MicroCam Camera.** The MicroCam device is a miniature colour camera able to acquire a JPEG image through a serial or USB connection. Created for robotics and video surveillance, the camera can send 0.75 fps video or compressed images ( $80 \times 60$  to  $640 \times 480$ ). The device has low power consumption.

Measurements have been taken at ambient temperature ( $15^{\circ}\text{C}$ ). We measure the intensity drained by the device at different phases by using the default software. We have obtained an intensity curve at different phases shown in Figure 9.

We notice that the power consumption increases, while images are acquired and transmitted. There is no consumption difference when changing image resolution or transfer speed. The intensity consumption is 60 mA in standby and 64 mA for an image acquisition. Thus, it does not vary that

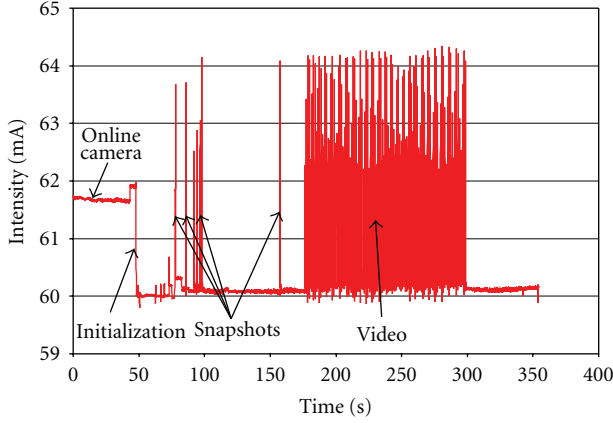


FIGURE 9: MicroCam sensor consumption.

much and can be assimilated as a constant model of 61 mA. In this approximation, the error value is 1 mA overestimated in standby mode and 3 mA underestimated in activity. But we experiment a stability problem at 3.3 Volts with the TTL that forces us to fall back on a 5 Volt DC supply. With this voltage, the power consumption reaches 305 mW.

**6.4. Suntech 10 W Solar Panel.** The Suntech panel is a multicrystalline silicon solar cell offering 10 Watt maximal power. It uses a textured cell surface and bypass diode design to harvest maximum sunlight by preventing partial shadowing attenuation. The Suntech total efficiency is 8.8%.

Measurements have been taken on location with real weather contexts. For this, we use a Kimo SL200 Pyranometer to measure the solar intensity reference. The solar panel is oriented at 35° to the south direction. It is the optimal angle in the applicative region.

We use a programmable power supply to measure the intensity provided with different programmed ranges from 0 to  $V_{oc}$  Volts. As one solar measurement takes up 10 seconds to record on the entire range, solar sunset could slightly vary, and thus introduce jitter. We have obtained an intensity-voltage ( $I$ - $V$ ) curve for different sunset irradiation powers ranging from 150 to 930 W/m<sup>2</sup>. Each  $I$ - $V$  curve, is converted to a  $P$ - $V$  (power-voltage) curve, where the maximum indicates the maximum power point. Figure 10 shows the different  $P$ - $V$  curves collected.

With the  $P$ - $V$  curves, we reach 10 Watt with 930 W/m<sup>2</sup> solar irradiation. We observed that the maximum power point propagates nearly 17 Volts for this solar power range. It corresponds to the  $V_{mp} = 17.20$  V and the  $I_{mp} = 0.58$  A given by the datasheet. This is because module performance is rated under Standard Test Conditions (STC): irradiance of 1000 W/m<sup>2</sup>, solar spectrum of AM 1.5, and module temperature at 25°C. Then, we modelled the Suntech maximum power point by a linear equation. Thus, we confirmed that the solar microsource harvested power range vary from 0 to 10 W.

**6.5. AS512/2 Battery Cell.** The AS512/2 cell is a sealed lead-calcium battery of 12 Volts for a 2 A h capacity. The battery

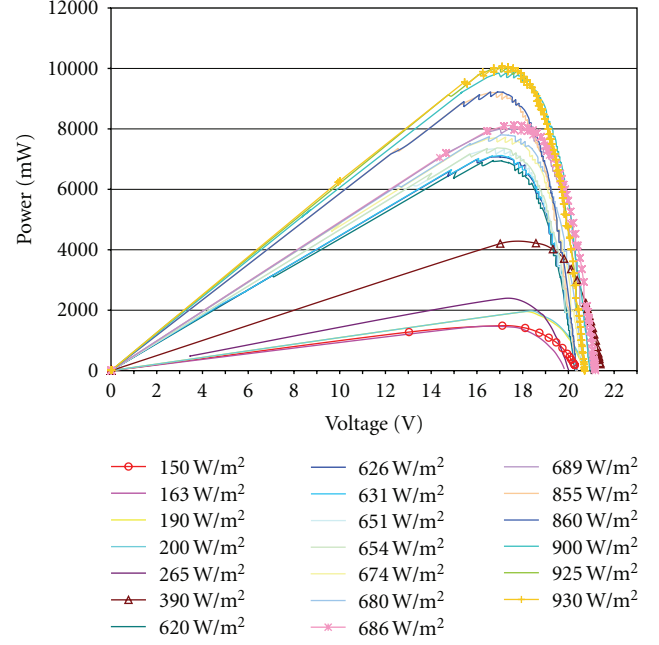


FIGURE 10: Suntech solar panel characterization.

uses lead-calcium grids with gas recombination and offers good recovery from deep discharge. For this model, we use a similar method of work as that developed by [48] to model a battery with simultaneous charge and discharge. Method: By using the FLPA methodology, we define the high-level parameters from an electrical point of view. We use parameters defined in electricity: the electromotive force, the capacity, the impedance, the efficiency, and the self-discharge as the base of the model criteria. The electronic model is defined by (4) as

$$V_{out} = V_{emf} - R_{int} * I, \quad (4)$$

where  $V_{out}$  is the battery tension (V),  $V_{emf}$  the battery electromotive force (V),  $R_{int}$  the battery internal resistance (Ohms), and  $I$  the provided current (A).

**bench description:** to take the measurements, we used a battery bench. The bench is composed of a power supply which provides the recharge current to the reference battery. The battery is placed in an incubator to measure behaviour as a function of temperature. The bench integrates an electronic load which varies according to a measurement scenario.

**Analysis:** to experiment a scenario, we must calibrate the power supply to deliver a constant recharge current. In this bench, a constant recharge current of 200 mA is applied while a variable discharge current is applied with a repeated pattern. The battery voltage is measured over time until the threshold tension limit has been reached.

**Characterization:** we calculate the  $R_{int}$  and the emf as a function of capacity. To do this, we take a couple of points on the data curve at each state transition. For example, one point on a peak period and one on another peak



(with a nearest delta). As we have a couple of points, we can solve (4) by the double equation system (5)

$$\begin{aligned} V_{out1} &= V_{emf} - R_{int} * I_1, \\ V_{out2} &= V_{emf} - R_{int} * I_2. \end{aligned} \quad (5)$$

By solving (5), we determine the evolution of the internal resistance and the electromotive force depending on the evolution of the battery capacity.

Model determination: then, we model the electromotive force and internal resistance parameters. We notice that  $V_{emf}$  can be modelled as a linear equation (6)

$$V_{emf} = -a \cdot q + b, \quad \text{with } q \text{ the capacity (A.h).} \quad (6)$$

The internal resistance has an exponential aspect modelled by (7)

$$R_{int} = a \cdot e^{(b \cdot q)} + c, \quad \text{with } q \text{ the capacity (A.h).} \quad (7)$$

Coefficients  $a, b$  in (6), and  $a, b, c$  in (7) are determined by regression approach. They are tightly coupled with the battery and the technology chosen.

Finally, the battery model is evaluated and compared to the real measurements in Figure 11. The absolute maximum measurement error is 8%.

As the battery model handles simultaneous charge and discharge, it can be used to model a self-recharging WSN. Thus, it can predict the battery autonomy of a WSN node according to its activity and environment.

Finally, by repeating the process for all other components, we can build a power map of the different node equipment. If components or circuit design change, models are rebuilt to reconstruct the power consumption map. Table 5 shows an example of the power consumption map for the required equipment of the CAPNET project.

## 7. Experiment

**7.1. Experiment Setup.** We measure the relevance of the DPM presented in Figure 3, by measuring its autonomy. This WF-DPM includes weather forecast enhancements.

The PEE is prepared and initialized with configuration files. The simulator needs:

- (i) to configure the hardware node components and set associated model parameters,
- (ii) to select the DPM used to simulate the node behaviours,
- (iii) to provide the Meteo environment for the scenario,
- (iv) to provide the RF Events to estimate the radio activity,
- (v) to provide the scenario file,
- (vi) to provide the global simulation parameters,
- (vii) to provide the destination of the results.

With the scenario builder, we recreate the area protection scenario developed in Section 2. A gas measurement is taken every 5 minutes for 15 seconds duration, permanently during

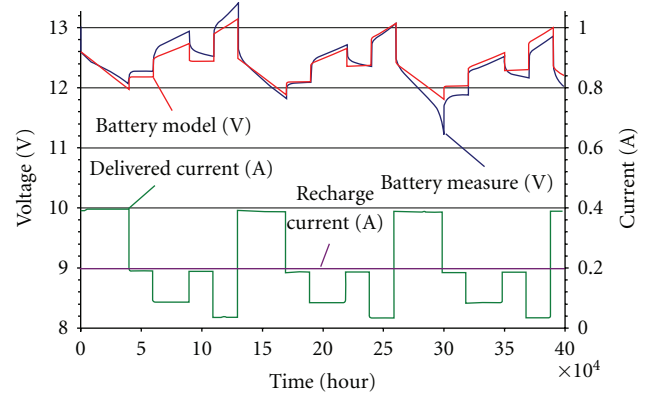


FIGURE 11: AS512 battery model results.

the entire scenario. A first detection with sensor 1 occurs, but is not yet confirmed by sensor 2. The second detection is confirmed by sensor 2 which sends an alarm to the supervisor. The supervisor has WF predictions and knows how long the system can take videos. After ten seconds of video, the supervisor has identified the threat and decides to send the intervention unit. The video is deactivated. The DPM waits for sensor 1 and 2 to disarm and goes back to Sleep mode.

**Simulation assumptions.** The weather environment has been acquired by recording real weather coefficients over time, and extracted for the weather scenario. DPM is built by translating the FSM into an external dynamic Labview module. Here, we consider just one node, thus we do not take external RF Events into account. Moreover, global simulation resolution is set to one second and we run the simulation for 529200 seconds (6.125 days).

We start the simulation and at the end, the simulator produces the results files in the results destination. Results are detailed current, voltage, power, and state values for all discrete time simulations. Global information is minimum power, maximum power, average power, and total energy consumption per components. The simulator provides the final global system average power and total energy consumption.

**7.2. Simulation Results.** Table 6 describes the WF-DPM energy and power consumption results. Results are grouped by components which callback the minimum power (in mW), the maximum power (in mW), the average power (in mW), the total energy (in Joules), and the total ratio (in %).

By examining the results of Tables 5 and 6, we can extract some conclusions. When we study the system equipment, we can see that the battery system has a capacity of 2 Ah. The power consumption in Sleep state is mainly given by the sensor 1, it is near 300 mW average in Sleep. It seems that this sensor have the highest power impact on the system (i).

We replay the same scenario without any harvesters to simulate the worst case/failure condition. In this case, the scenario duration is 44.4 hours which means nearly

TABLE 5: Node power consumption map.

Type	Component/chip	Mode 1	Mode 2	Mode 3
CPU	PIC24F256	Full:	Idle:	Sleep:
	@3.3 Volts	16 mA	2.5 $\mu$ A	100 nA
		53 mW	8.5 $\mu$ W	330 nW
SRAM (x2)	A623308A	Op R/W:	StandBy:	OFF:
	@5 Volts	35 mA	10 $\mu$ A	0 mA
		175 mW	50 $\mu$ W	0 mW
RF1	MRF24J40	TX:	RX:	Sleep:
	@3.3 Volts	130 mA	25 mA	2 $\mu$ A
AES128		430 mW	82.5 mW	6.6 $\mu$ W
RF2	CC1101	TX:	RX:	Idle:
	@3.3 Volts	75 mA	25 mA	200 nA
WakeOn		248 mW	82,5 mW	660 nW
Sensor1	PRO-45 <sup>1</sup> /S912	Operating:	/	/
	@10–13 Volts	300 mW	/	/
Sensor2	Mygale 4S	Operating:	/	/
	@10–13 Volts	600 mW	/	/
Video	MicroCam	Capture:	StandBy:	/
	@5 Volts	64 mA	61 mA	/
		320 mW	305 mW	/
Gas1	OLCT 50	(18–30 V)	/	/
	@18 Volts	1080 mW	/	/
Gas2	OLCT 80 @18 Volts	(12–28 V) 900 mW	/	/
			/	/
Gas3	TX-12	(6.5–9 V)	/	/
	@9 V	108 mW	/	/
MPPT	LT3652	Operating:	StandBy:	Idle:
	$\eta = 90\text{--}95\%$	2.5 mA	85 $\mu$ A	15 $\mu$ A
	(in = 4.95–32 V)	Variable	Variable	Variable
DC/DC (Down)	LM3100	Operating:	StandBy:	/
	$\eta = 85\text{--}92\%$	0.7 mA	17 $\mu$ A	/
	@(10.5–13 V)	Variable	Variable	/
DC/DC (Up)	Power stages:	(1–3x) 3.3 V	(1x) 5 V	(0–1x) 9 V
	MAX618	Operating:	Shutdown:	/
	$\eta = 93\%$	2.5 mA	3 $\mu$ A	/
MS1 Solar	@(10.5–13 V)	Variable	Variable	/
	Power stages:	(0–2x) 18 V		
	SunTech 10 W	1000 W/m <sup>2</sup>	500 W/m <sup>2</sup>	200 W/m <sup>2</sup>
MS2 Wind	PMax	10 W	5 W	2 W
	Rutland 504	75 Km/h	20 Km/h	15 Km/h
	PMax	60 W	4.3 W	1.5 W
MS3 Thermal	MPG604	$dT = 70^\circ\text{C}$	/	/
		80 mW	/	/
CELL Battery	AS512/2	12 V	/	/
	(10.5–13 V)	2 Ah	/	/

1.85 days. Thus, the initial battery capacity of 2 Ah will be too low to cover the scenario duration of 6 days in the failure condition. Consequently, the project team decided to extend the battery capacity to a 7 Ah battery. With further

verification, the battery level never drops down; the power supply is now wellscaled for the system demand (ii).

Moreover, the WF-DPM always switches to the potential maximal energy microsource (iii). The benefit has not been

TABLE 6: Simulation results.

Component	Min	Max	Avg	Energy	Ratio
S912	351.26	354.348	352.732	186666	70.35%
Mygale 4S	0	1619.73	38.8471	20557.9	7.75%
MicroCam	0	305	5.09774	2697.73	1.02%
OLCT 50	0	1048.8	0.0812	43.001	0.016%
OLCT 80	0	12.3993	0.00096	0.508	1.9E-4%
PIC24F256	1E-5	52.8	1.393	737.199	0.278%
A623308A	0	160	4.2213	2233.92	0.842%
MRF24J40	82.5	429	82.5956	43709.6	16.47%
CC1101	0	165	0.0452	23.925	0.009%
LM3100	0	74.70	14.58	7717.66	2.9%
LM3100	0	5.866	0.155	81.91	0.031%
LM3100	0	82.0588	1.644	870.29	0.328%
MAX618	0	187.271	0.0145	7.678	0.0029%
LT3652	0	140.992	41.4749	21948.5	8.272%
SunTech	0	2819.84	829.497	438970	/
Others	0	0	0	0	/
Total sensors	–		396.759	209965	79.13%
Total radios	–		82.6408	43733.5	16.48%
Total DC/DC	–		16.3975	8677.54	3.27%
Total CPU/RAM	–		5.6143	2969.96	1.12%
Total harvested	+		788.022	417021	/
Total cons.			501.412	265346	100%

quantified here, but it does offer a power switcher solution like the one in Duranode system [45].

WF-DPM can help a supervisor user by automatically computing the maximal duration of each service. Predicting potential future environment energy permits us to present lifetime autonomy indicators for real-time purposes (iv).

Finally, the prototype has helped to identify three power bottleneck types and explore one new technology:

- (i) heavy sensor consumption,
- (ii) battery total autonomy and harvester scaling,
- (iii) power switcher for automatic source selection.,
- (iv) experiment a new WF estimating the remaining time duration for a service.

WF-DPM presents interesting information to help a supervisor activate the adapted service. Also, WF-DPM can be used for automatic DPM decisions depending on the application context. For example, there are some particular deep-sea beacons which need to come back to the surface for battery recharging. Here, WF-DPM can predict the potential best period to emerge to maximize energy harvesting. We think that WF has many applications that can be exploited by WSN.

## 8. Conclusion

In this paper, we proposed to develop a power and energy estimator to determine the node autonomy duration of

WSNs. We present the CAPNET project with a representative scenario of fire brigade intervention. We combine the modelling approach with a problem decomposition approach to realize the estimator. The simulator is built upon a two-layer representation, handling the node logical part and the node hardware part. By using the FLPA methodology, we built models with successive incremental refinements to adjust granularity to project duration, cost, and needs. The power consumption engine exploits these models to manage the hardware part and the DPM strategy at runtime.

The most important outcomes of this paper is that: the simulator ability to handle recharging WSNs to predict node autonomy in a given scenario. This way, it can identify power bottlenecks such as states with heavy sensor consumption. Thus, the tool enables the calibration of the total autonomy of the battery and the harvester scale accordingly. Moreover, by integrating WF into DPM, we can automatically select the potentially best ambient power source. We validate that a WF-DPM can be used to calculate the remaining time duration for their services.

From this work, we learn to exploit component cartography consumption, allowing for the quick identification of abnormal power peak consumption. We observed that the DPM must be easily writable because during adjustments, lot of changes impact the DPM policy.

The FLPA methodology can be exploited for energy characterization of hardware components, not just CPU ones. This approach is representative about the way tools must be based on real experimentation behaviours, to determine its own particular models. By this way, it does not restrict to a particular WSN, but can include lot of type of WSNs. Now, we want to explore WF-DPM for automatic DPM decisions, such as for particular deep-sea beacons which need to surface for battery recharging. Here, WF-DPM can predict the potential best period to emerge to maximize the energy harvesting.

By the end of the CAPNET project, the goal is to build a demonstrator including final sensors, harvesters, and others system components. Now, we expect to measure the real difference between the simulator and the demonstrator. This will permit us to enhance the power/energy estimator to better adapt tool accuracy to reality.

## Acknowledgments

The authors would like to thank the “CAPNET” project partners: fire service partners: SDIS29 (Service Départemental d’Incendie et de Secours du Finistère), industrial partners: ERYMA Security Systems (Leader), DeltaDore, and Atlanti-crE, academics partners: I.E.T.R Rennes and Lab-STICC Lorient. Financial pole: “Pôle Image et Réseaux”, “Conseil Général du Morbihan”, “Région Bretagne”, and “Ministère Français de l’Economie, des Finances et de l’Industrie”, spellchecker: Joanna Ropers.

## References

- [1] Wikipedia, “Wireless sensor network,” 2010, [http://en.wikipedia.org/wiki/Wireless\\_sensor\\_network](http://en.wikipedia.org/wiki/Wireless_sensor_network).

- [2] P. H. Chou and C. Park, "Energy-efficient platform designs for real-world wireless sensing applications," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '05)*, pp. 912–919, Irvine, Calif, USA, November 2005.
- [3] M. Y. I. Idris, E. M. Tamil, N. M. Noor, Z. Razak, and K. W. Fong, "Parking guidance system utilizing wireless sensor network and ultrasonic sensor," *Information Technology Journal*, vol. 8, no. 2, pp. 138–146, 2009.
- [4] F. J. Rincón, M. Paselli, J. Recas et al., "OS-based sensor node platform and energy estimation model for health-care wireless sensor networks," in *Proceedings of Design, Automation and Test in Europe (DATE '08)*, pp. 1027–1032, Munich, Germany, 2008.
- [5] Y.-H. Chee, M. Koplow, M. Mark et al., "PicoCube: a 1cm<sup>3</sup> sensor node powered by harvested energy," in *Proceedings of the 45th annual Design Automation Conference*, pp. 114–119, Anaheim, Calif, USA, 2008.
- [6] M. A. Guerrero, E. Romero, F. Barrero, M. I. Milanés, and E. González, "Overview of medium scale energy storage systems," in *Proceedings of the 6th International Conference on Computability and Power Electronics (CPE '09)*, pp. 93–100, May 2009.
- [7] 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, 2005.
- [8] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 457–462, 2005.
- [9] M. Chen and G. A. Rincón-Mora, "Single inductor, multiple input, multiple output (SIMIMO) power mixer-charger-supply system," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '07)*, pp. 310–315, August 2007.
- [10] C. Park and P. H. Chou, "AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor nodes," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Adhoc Communications and Networks (Secon '06)*, vol. 1, pp. 168–177, 2006.
- [11] F. Simjee and P. H. Chou, "Everlast: long-life, supercapacitor-operated wireless sensor node," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 197–202, 2006.
- [12] M. A. Green, *Third Generation Photovoltaic: Advanced Solar Energy Conversion*, Springer, New York, NY, USA, 2005.
- [13] S. Roundy, P. K. Wright, and J. Rabaey, "A study of low level vibrations as a power source for wireless sensor nodes," *Computer Communications*, vol. 26, no. 11, pp. 1131–1144, 2003.
- [14] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 299–316, 2000.
- [15] R. J. M. Vullers, R. van Schaijk, I. Doms, C. Van Hoof, and R. Mertens, "Micropower energy harvesting," *Solid-State Electronics*, vol. 53, no. 7, pp. 684–693, 2009.
- [16] D. Brunelli, L. Benini, C. Moser, and L. Thiele, "An efficient solar energy harvester for wireless sensor nodes," in *Proceedings of Design, Automation and Test in Europe (DATE '08)*, pp. 104–109, Munich, Germany, 2008.
- [17] V. Ekanayake, C. Kelly IV, and R. Manohar, "An ultra low-power processor for sensor networks," *ACM SIGARCH Computer Architecture News*, vol. 32, no. 5, pp. 27–36, 2004.
- [18] P. Guitton-Ouhamou, H. Ben Fradj, C. Belleudy, and M. Auguin, "Low power co-design tool and power optimization of schedules and memory system," in *Proceedings of the 14th International Workshop on Integrated Circuit and System Design Power and Timing Modeling, Optimization and Simulation (PATMOS '04)*, vol. 3254 of *Lecture Notes in Computer Science*, pp. 603–612, Santorini, Greece, 2004.
- [19] J. Laurent, N. Julien, E. Senn, and E. Martin, "Functional level power analysis: an efficient approach for modeling the power consumption of complex processors," in *Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE '04)*, vol. 1, pp. 666–667, 2004.
- [20] M. Pedram and J. Rabaey, *Power-Aware Design Methodologies*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.
- [21] L. Benini and G. De Micheli, "System-level power optimization: techniques and tools," *ACM Transactions on Design Automation of Electronic Systems*, vol. 5, no. 2, pp. 115–192, 2000.
- [22] S. Irani, S. Shukla, and R. Gupta, "Algorithms for power savings," in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 37–46, Baltimore, Md, USA, 2003.
- [23] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy-aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, 2003.
- [24] G. Kremer, J. Osmont, and B. Pottier, "A process oriented development flow for wireless sensor networks," in *Proceedings of the International Workshop on Smalltalk Technologies*, Brest, France, 2009.
- [25] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [26] Wikipedia, "List of wireless sensor nodes," 2010, [http://en.wikipedia.org/wiki/List\\_of\\_wireless\\_sensor\\_nodes](http://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes).
- [27] S. Douhib, *Operating systems power and energy consumption characterization for critical embedded systems described in AADL*, Thesis financement, ITEA, projet SPICES, Université de Bretagne Sud, Lorient, 2009.
- [28] T. Quarles, "The SPICE3 implementation guide," Rapport techniqueUCB/ERL M89/44, EECS Department, University of California, Berkeley, Calif, USA, 1989.
- [29] K. Baynes, C. Collins, E. Fiterman et al., "The performance and energy consumption of embedded real-time operating systems," *IEEE Transactions on Computers*, vol. 52, no. 11, pp. 1454–1469, 2003.
- [30] S. Kang, H. Wang, Y. Chen, X. Wang, and Y. Dai, "Skyeye: an instruction simulator with energy awareness," in *Proceedings of the 1st International Conference on Embedded Software and Systems (ICESS '05)*, vol. 3605 of *Lecture Notes in Computer Science*, pp. 456–461, Springer, 2005.
- [31] N. Fournel, A. Fraboulet, and P. Feautrier, "eSimu: a fast and accurate energy consumption simulator for real embedded system," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, pp. 1–6, June 2007.
- [32] T. K. Tan, A. Raghunathan, and N. K. Jha, "EMSIM: an energy simulation framework for an embedded operating system," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 2, pp. 464–467, Phoenix, Ariz, USA, 2002.



- [33] S. Gurumurthi, A. Sivasubramaniam, M. J. Irwin et al., "Using complete machine simulation for software power estimation: the SoftWatt approach," in *Proceedings of the 8th International Symposium on High-Performance Computer Architecture (HPCA '02)*, pp. 141–150, IEEE Computer Society, Boston, Mass, USA, 2002.
- [34] V. Shnayder, M. Hempstead, B. Chen, and M. Welsh, "PowerTOSSIM: efficient power simulation for TinyOS applications," in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.
- [35] ChipVision Inc., "PowerOpt<sup>tm</sup>," low-power synthesis tool, ANSIC/System C code into VerilogRTL, (2003–2009).
- [36] E. Senn, J. Laurent, N. Julien, and E. Martin, "SoftExplorer: estimation, characterization, and optimization of the power and energy consumption at the algorithmic level," in *Proceedings of the International Workshop on Power and Timing Modeling Optimization and Simulation (PATMOS '04)*, vol. 3254 of *Lecture Notes in Computer Science*, pp. 342–351, 2004.
- [37] A. Courtay, O. Sentieys, J. Laurent, and N. Julien, "High-level interconnect delay and power estimation," *Journal of Low Power Electronics*, vol. 4, no. 1, pp. 21–33, 2008.
- [38] D. Saadia, S. Eric, D. Jean-Philippe, L. Johann, and B. Dominique, "Model driven high-level power estimation of embedded operating systems communication services," in *Proceedings of International Conference on Embedded Software and Systems (ICCESS '09)*, pp. 475–481, Hangzhu, China, 2009.
- [39] M. Rafiee, M. B. Ghaznavi-Ghoushchi, and B. Seyfe, "Processing power estimation of simple wireless sensor network nodes by power macro-modeling," in *Proceedings of the International Conference on Computer Engineering and Technology (ICCET '09)*, pp. 428–431, January 2009.
- [40] J. Glaser, D. Weber, S. A. Madani, and S. Mahlknecht, "Power aware simulation framework for wireless sensor networks and nodes," *EURASIP Journal on Embedded Systems*, vol. 2008, Article ID 369178, 16 pages, 2008.
- [41] N. Julien, J. Laurent, E. Senn, and E. Martin, "Power consumption modeling and characterization of the TI C6201," *IEEE Micro*, vol. 23, no. 5, pp. 40–49, 2003.
- [42] D. Elléouet, Y. Savary, and N. Julien, "An FPGA power aware design flow," in *Proceedings of the 16th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS '06)*, vol. 4148 of *Lecture Notes in Computer Science*, pp. 415–424, Montpellier, France, 2006.
- [43] Meteociel, <http://www.meteociel.fr/>.
- [44] Meteo-France, <http://france.meteofrance.com/>.
- [45] C. Park and P. H. Chou, "Power utility maximization for multiple-supply systems by a load-matching switch," in *Proceedings of the International Symposium on Lower Power Electronics and Design (ISLPED '04)*, pp. 168–173, Newport Beach, Calif, USA, 2004.
- [46] S. Irani, S. Shukla, and R. Gupta, "Online strategies for dynamic power management in systems with multiple power-saving states," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 325–346, 2003.
- [47] L. Benini, A. Macii, E. Macii, and M. Poncino, "Discharge current steering for battery lifetime optimization," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 118–123, August 2002.
- [48] O. Erdinc, B. Vural, and M. Uzunoglu, "A dynamic lithium-ion battery model considering the effects of temperature and capacity fading," in *Proceedings of the International Conference on Clean Electrical Power (ICCEP '09)*, pp. 383–386, June 2009.
- [49] C. Renner, J. Jessen, and V. Turau, "Lifetime prediction for Supercapacitor-powered wireless sensor nodes," in *Proceedings of the 8th GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze" (FGSN '09)*, Hamburg, Germany, 2009.
- [50] National instruments, "NI LabVIEW," 2010, <http://www.ni.com/labview/>.
- [51] L. Gao, S. Liu, and R. A. Dougal, "Dynamic lithium-ion battery model for system simulation," *IEEE Transactions on Components and Packaging Technologies*, vol. 25, no. 3, pp. 495–505, 2002.