

Stochastic Modeling and Analysis for Environmentally Powered Wireless Sensor Nodes

Alexandru E. Şuşu*, Andrea Acquaviva[†], David Atienza*, Giovanni De Micheli*,

*LSI/EPFL, EPFL-IC-ISIM-LSI Station 14, 1015 Lausanne, Switzerland

{alex.susu, david.atienza, giovanni.demicheli}@epfl.ch,

[†]DI/University of Verona, Strada le Grazie 15, Verona, Italy, andrea.acquaviva@univr.it

Abstract—Environmental energy is becoming a feasible alternative for many low-power systems, such as wireless sensor nodes. Designing an environmentally powered device faces several challenges: choosing the exact type of the energy harvester, the energy storage elements and determining the duty cycle of the application. With harvesting, the design process becomes even more difficult because it also has to take into account the unpredictability of the energy source.

The contribution of this paper is a methodology that facilitates the analysis of energy harvesting nodes. The existing modeling strategies for battery powered systems are not suitable because they do not capture the uncertainty of the power source. Also, the metrics of interest for battery powered devices are different, as opposed to the harvesting powered ones: in the former case we search to maximize the system lifetime, while in the latter case a more expressive goal is to increase the system availability.

Keywords: Energy harvesting, Wireless Sensor Nodes, Discrete Time Markov Chains, stochastic modeling, system metrics

I. INTRODUCTION

Energy harvesting has grown in the last decade as a viable technology for powering wireless sensor nodes and mobile electronics [21], [15], [10], [26], [7].

Nevertheless, environmental energy comes at a price. Even if harvesters can ensure a theoretically unlimited amount of energy over time, the power they provide is unpredictable. For this we use power storage elements, such as rechargeable batteries or supercapacitors, in order to have energy available for later use. However, the buffers are finite, and therefore they cannot completely hide the unreliability of the energy source, for example, when the harvester is not generating energy for a long period of time.

These intrinsic characteristics of ambient energy introduce several challenges. The nodes have to be able to recover from blackout periods caused by the unavailability of energy, from both the energy harvester and the energy buffer. Also, the design of the system becomes more complex than for battery powered systems. Besides this, the focus of the analysis and the optimization criteria for environmentally powered systems is different from the one of battery powered devices: while in the latter case we search to maximize the lifetime of the system, in the former case a good design objective is to increase the availability of the device for a given long period of time.

Because of all of the aforementioned reasons, hardware and software designers can benefit from a characterization of the system behavior under environmental power conditions with the following goals: i) the selection of components such as the harvester, energy storage elements, DC/DC converter, microcontroller, radio, sensors; ii) the tuning of the runtime policies for power management.

Designers can have as an optimization goal the maximization of the operation duty cycle (the duty cycle is the ratio between the length of the time the device spends in active mode and the duration of the entire period), which increases the data rate provided by the network, but also decreases the lifetime. As such, a suitable trade-off has to be found. For example, [12], [11] determines a maximum duty cycle (in order to maximize the quality of service, QoS), which ensures an unlimited lifetime, or, as it is called, an *energy neutral operation* point. Similarly, [16] proposes a design method for a controller that maximizes the duty cycle, while respecting, for example, the amount of available energy. We argue that using our methodology we can explore and quantify the effectiveness of the various policies introduced in these papers. In fact, we have already performed the performance evaluation of dynamic policies implemented on a video sensor node in [5].

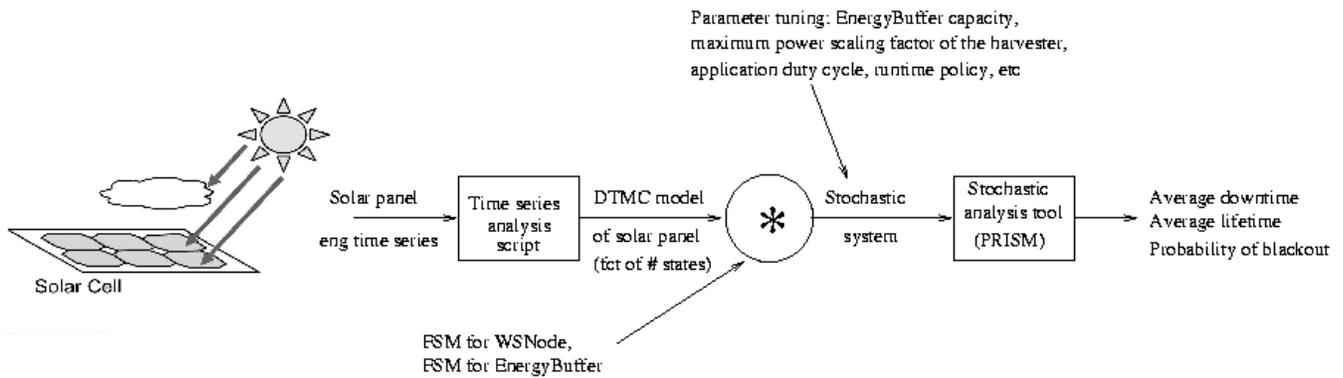


Fig. 1. Block diagram of the proposed methodology.

The remainder of this paper is organized as follows. In Section II we present the foundations used in our modeling approach and the differences with state-of-the-art models for sensor nodes. In Section IV and V we describe how we use this approach to model a real-life wireless sensor node application and we describe the results obtained during our design space exploration (see Figure 1 for an overview).

II. MODELING BACKGROUND

In this work we propose a behavioral black-box modeling technique similar to the one introduced in [3]. Such modeling contains no information on the internal structure of the electronic components, but just tries to mimic the I/O characteristics (see Figure 3). The system can be described at various levels of behavioral abstraction (determined essentially by the length of the time step and by the energy discretization step), allowing us to trade accuracy for the effort of computation.

The block diagram we introduce in Section IV can be considered a power-flow model. It is represented by a set of blocks that provide, store and consume power. The harvester device produces energy, which is stored in an energy buffer. The sensor node acts as a power sink, which takes the energy from the buffer and uses this energy to execute an application.

In general, other levels of abstractions can be used to analyze systems [3]. Transistor-level models contain the complete switch-level structure of the electronic components. Such models (for example, defined in HSpice) are too complex to be analyzed for long periods in reasonable time. Another abstraction level, the behavioral white-box model, employs simplified behavioral component models. Even the simulation of simple white-box behavioral models (using behavioral SystemC, VHDL, Matlab, etc.) requires excessive time, because the switching period of typical electronic components is of a few microseconds, and thus it is several orders of

magnitude smaller than the required time, of the order of months, for the analysis of downtime (i.e., the sum of all the blackout periods over a given time interval) or lifetime (i.e., the time until the system runs out of power). We therefore trade the accuracy for the sake of a tractable analysis.

An accurate model of the energetic sources is essential for evaluating the system’s average productivity (e.g., availability), and, in principle, to ensure the system’s command and management. Simulation will provide results only for the period over which weather data is available. Since the results are different if we use other series with the same statistical properties, we are interested to know the range of these results. Therefore, finding a representative model able to capture the uncertainty of the energy source requires attentive thinking. For example, the performance of solar energetic systems is dependent on the variable levels of solar radiation, which are neither completely random, nor fully deterministic [23]. For the stochastic energy model, no classical probability law could be suitably fitted to solar radiation empirical probabilities. On the other hand, the literature on climatology and renewable energy is already well established, using historical data, descriptive Markov chain models for various forms of environmental energy, such as solar radiation [9], [18], [23], wind speed [17] and ambient temperature, or autoregressive process models [22]. More exactly, [23] proposes a first-order stationary discrete time Markov chain model for each month of the year, due to the big monthly variations, built from traces taken over a period of 20 years.

Similarly, in this paper we build a first-order Discrete Time Markov Chain (DTMC) model of a solar energy harvester using current-voltage traces obtained from the SensorScope project [2] from EPFL. The stochastic source feeds the deterministic Finite State Machine (FSM) component of the energy storage element and, indirectly, the wireless sensor node used for weather

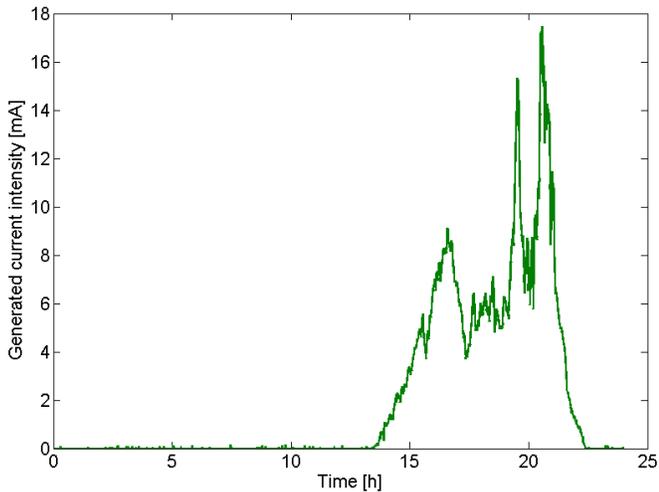


Fig. 2. Trace of the intensity of the generated current of a $5.5 \times 15 \text{ cm}^2$ solar panel, starting from Nov 14th, 2005 at 18:28, for 24 hours.

application. The goal of our methodology is to study the feasibility of the usage of the environmental power for the given target application.

We use the defined stochastic model to perform the design exploration of harvesting sensor nodes in order to achieve the previously mentioned goals. For this we use a stochastic analysis tool, namely the probabilistic model checking tool PRISM [13], to explore the impact of the design parameters and the environmental conditions such as battery size, activity duty cycle and power profile of the ambient energy, on the various metrics of the model. The reason we use a tool to perform the analysis is that the system model has a very big number of states. Also, the nature of the model makes it hard to analyze it manually. An argument is that the runtime policies can adapt, for example, the duty cycle, when various parameters change such as the energy level in the battery, network requests or sensor events. This makes analytical methods for calculating properties such as the average lifetime for the stochastic system intractable. For a similar reason, [4] mentions the fact that properties for complex Markov systems are difficult to evaluate analytically, and therefore propose automating the process using the tool NANOPRISM, a spin-off of PRISM.

Models with similar abstraction levels are given in [12], [10]. Yet, they use a different, less descriptive modeling technique, which allows the development of specific analytical methods, such that they can reason, for example, about the energy neutral operation point.

As anticipated, the stochastic analysis we perform allows us to establish various properties such as the expected system downtime or lifetime by automated analytical means, without performing expensive simulations. These properties are relevant for the hardware and

software designers for selecting the components, such as the energy buffer elements and the harvester, as well as designing the software application. For instance, the average lifetime can help in tuning the checkpointing interval, when using backward recovery in order to cope with power blackouts [8].

In Section IV we describe how we use this approach to model our target application. We present in Figure 1 a schema of the methodology of the paper, which is detailed in Sections IV and V.

The PRISM tool is able to infer many properties of the stochastic model through exhaustive exploration, many of them being non-trivial and amenable only by computer analysis. PRISM has been successfully employed to investigate the performance of stochastic power management policies [28], [20]. There, the *service requester* and the *service provider* are modeled in a probabilistic way, and also, the *power manager* implements a stochastic policy. This is in contrast with our system, in which we model only the power source as a stochastic process, while the other components are deterministic. For that system, properties such as the expected power consumption, expected number of requests in the queue or the probability that the system is able to serve requests by a given time are studied. Also, in [5] we use PRISM to model an energy harvesting wireless sensor node, on which we run a dynamic reconfiguration policy based on the inputs of the system.

In [19] a stochastic model for a solar powered wireless sensor node is used to analyze the following QoS measures for several stochastic policies proposed: the average queue length, the average battery capacity, the sleeping probability, the average delay, etc. However, their modeling technique is different than ours: they construct analytically a Markov chain model for solar radiation, by considering the wind speed, the cloud size, and the probability of having a cloudy period.

III. ENERGY HARVESTING SENSOR NODE

The target system we consider is the TinyNode [7] wireless sensor node used for weather monitoring, in the context of the SensorScope project [2]. The weather station can measure key environmental data such as air temperature and humidity, surface temperature, incoming solar radiation, wind speed and direction, precipitation, and soil moisture and pressure. The nodes communicate the acquired data via the radio directly to a base station (which implies the sensor network has a star topology), using the Berkeley Medium Access Control (BMAC) protocol [24]. For simplicity, in this paper we consider only the soil-water content sensor.

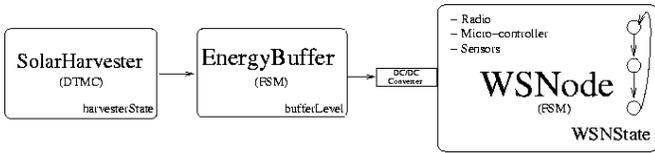


Fig. 3. Behavioral black-box model of the system.

We consider that the base station is not energy constrained, since it is connected to the electric grid.

A. Energy Harvester

The harvester device we consider is a solar panel. We use traces of the intensity value of the current generated by a real solar cell, over a given period of time [2], [6]. In Figure 2 we present an example of the type of trace that can be obtained during one complete day.

The voltage of the harvester is almost constant at a value of 5 V. The behavior of the harvester is periodic with respect to the day cycle, with variations caused by clouds, terrain obstacles from the sun and, in the longer term, caused by seasonal changes. We use the time series corresponding to Figure 2 to build the statistical model of the harvester, as explained in Section IV.

We mention that the system does not use a maximum power point tracker (MPPT) module, which helps attaining an optimal electrical load for a solar panel, because the MPPT device is not meant for low power devices [25], such as sensor nodes.

B. Energy Storage Element

We consider as energy buffer two supercapacitors of 11F or 22F and 2.5 V put in series in order to attain a maximum voltage of 5V, which implies a maximum energy capacity of 68,750 or 137,500 mJ, respectively (since $E_{cap} = \frac{CV^2}{2}$). We opt to use supercapacitors since they have a bigger maximum number of recharge cycles than batteries, making them more suitable for environmentally powered devices. However, they have higher leakage current and smaller energy density than batteries [3]. Also, supercapacitors are easier to model than electrochemical batteries [3], [29].

C. Wireless Sensor Node

The components of the node that we consider due to their big contribution for the power consumption are the DC/DC converter, the microcontroller, the radio interface and the soil-moisture sensor.

The microcontroller is a TI MSP430 featuring low active power (0.6 nJ/instruction), low standby power

(2 μ W), fast wakeup from standby to active mode (6 μ s) and on-chip 12-bit analog-to-digital converter.

The TinyNode uses the XE1205 radio transceiver from Semtech (31 nJ/bit) [27].

Also, the node uses the ECH2O EC5 soil-water content sensor, which needs 10 ms to perform one measurement and consumes 10 mA at 3.3 V during this time.

The DC/DC converter, in our case the LT1615 pulse-width modulation converter [30], is used as a step-down converter, i.e. it can generate an output DC voltage smaller than its input DC voltage. This voltage regulation is required since the SensorScope platform presents a variable voltage in the energy buffer, while the electronics requires a constant voltage of 3.3V. To determine the efficiency of the converter for the various points of operation of the load, the sensor node, while sleeping, sensing, receiving and transmitting we consult the datasheet of the LT1615.

The node acts in a simple deterministic way: every 30 seconds it senses, it processes the acquired data and it sends the results to the base station. Every node communicates only with the base station. As already stated, sensing the soil humidity takes 10 ms. Sending the data through radio takes 70 ms. The radio listens for 2.25 seconds - the BMAC protocol requires this in order to allow the various nodes in the sensor network to synchronize. (Depending on the number of nodes we want to allow to synchronize, the length of the radio listening task can be adjusted proportionally to this number.) After this, the microcontroller and the radio are put in a low power state for 27.67 seconds, which makes the duty cycle of the application be $2.33 / 30 = 7.76\%$.

IV. STOCHASTIC BEHAVIORAL MODEL OF THE SYSTEM

The behavioral model employed in this paper focuses on the energy generation and consumption.

We are interested in assessing the properties of a single node and not of the whole network running the weather application, and therefore we model only the individual sensor node. We consider negligible the overhead the nodes spend in avoiding and recovering from collisions. Since the network uses a star topology, there is no multi-hop routing overhead. Because of these last two arguments, each node consumes a deterministic amount of energy for network communication. This allows decoupling the analysis of each node from the behavior of the other nodes in the network.

The target system previously described is modeled as the parallel composition of the following modules: i) the harvesting device (*SolarHarvester*); ii) an energy storage

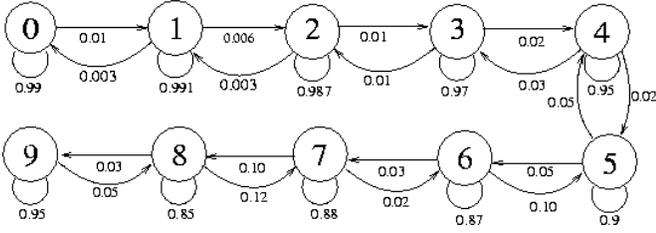


Fig. 4. Discrete Time Markov Chain with 10 states (the number of states affects the accuracy and the processing time) for the SolarHarvester model, built from the trace in Figure 2.

element (be it a supercapacitor or a rechargeable battery), called *EnergyBuffer*; iii) the node (*WSNode*), which acts as the power sink of the system, which incorporates the radio transceiver, the microcontroller and the sensor. The structure of the resulting system, a complex finite state system, can be seen in Figure 3.

Each module has a certain number of states, among which transitions are defined. Transitions can be either probabilistic or deterministic. They are specified through the use of arithmetic expressions with the standard arithmetic operators, with the exception of divisions, due to the fact the variables of the modules are restricted to be integers.

Choosing the physical length of the execution time step affects the accuracy and the time spent to perform the analysis. To increase the speed of the analysis of the model checker, we use the time scaling technique described in [14]. We choose a time step of 10 seconds, according also to the solar panel sampling resolution of the traces we used. This allows us to perform, for example, availability analysis for a given period in the order of months.

The behavior of the model is the following. Initially the energy buffer is full. In every time step we add to the buffer energy level (variable *bufferLevel*) the contribution from the harvester, taking into consideration also the efficiency of the energy transfer, and subtract the energy consumed by the node. We consider the power consumption and generation to be constant over an execution step, which allows a simpler analysis when the system is running out of power in a time step.

When the energy buffer is full, the system uses directly the energy from the harvester. If that energy is not enough for the given time step, it consumes energy from the energy buffer as well. If the energy from the harvester is bigger than what the platform requires, then the surplus is wasted. If the system experiences a blackout it constantly attempts to restart the application once every time step.

Each module is described as a state diagram, where

transitions can be either deterministic (for FSM) or probabilistic (for DTMC), as explained in Section II. For the rest of this section, we give a description of each module.

SolarHarvester. Using a solar panel trace, such as the one in Figure 2, we build a DTMC model. For this we need to discretize the energy generated by the harvester. We accomplish this by equally dividing the interval of possible energy quantities generated in the (predefined) time step of 10 seconds and associate each of the 10 subintervals, in our case, to one of the states of the Markov chain. We build the transitional frequency matrix and the transitional probability matrix the standard way, as in [23], by using a simple home built script that analyzes the collected time series, as depicted in Figure 1. More exactly, to assign probabilities on the transitions originating in a state of the DTMC, we go over the solar panel current trace and count the occurrences of each transition from that state (by transition in the trace we understand the jump from one energy level to the next, 10 seconds later). Then, we normalize these frequencies in order to have the sum of probabilities on the transitions from that state equal to 1.

The choice of the number of states of the DTMC is a tradeoff between the accuracy of the module abstraction with respect to the real harvester and the exploration speed determined by the number of states of the entire system. The statistical model can be considered representative for the days of the month of November in the geographic region of the experiments. Using the same technique, statistical models for different periods of the year can be generated. We advocate the creation of a library of different models associated, for example, with each month, from which the designer can choose, as in [23]. Therefore, the model in Figure 4 is a possible item from the library.

In Figure 4 we show an example of DTMC model, built from the trace in Figure 2. This figure shows that the cascade nature of the DTMC: the transitions in the Markov chain are only allowed between consecutive states. This comes from the fact the energy generated in a time step by a solar panel follows an almost continuous trend: jumping from a lower energy level to a much higher one is very improbable.

EnergyBuffer. The energy buffer is modeled as an FSM, where the states represent the energy levels of the buffer. The deterministic transitions are triggered only by the state of consumption of energy by the *WSNode* and by the contribution with energy by the *SolarHarvester*.

The model of the energy buffer is very simplistic and does not take into consideration the aging or leakage. We

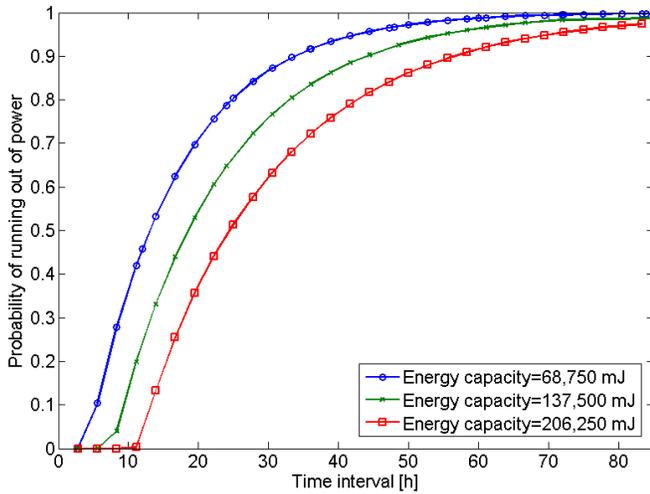


Fig. 5. Variation of the probability that the system runs out of power with respect to the time interval for various energy storage capacities.

consider the energy storage element to have a constant voltage, no matter what energy charge it has. Therefore, for different levels of charge of the buffer, if the current load is the same then the same amount of energy will be taken from the buffer.

WSNode. The node is modeled as an FSM using three states corresponding to every 10 seconds of the period of 30 seconds of operation, as described in Section III. In the first step of the three time-steps long cycle, all the important activity happens: sensing, processing, radio transmitting and synchronization, which take together about 2.33 seconds; the rest of the time, the node sleeps. The following two time-steps, the node continues to be in standby mode, until the 30 seconds long cycle concludes. We model the node at a higher level of abstraction, i.e., the radio, processor and sensor components are collapsed in one module, which is reasonable since in our model we focus on the power supply chain, with emphasis on the uncertainty of the solar panel.

We characterize the energy consumption of the WSNode module in the three states. For this we consider the most power consuming components of the TinyNode platform: the DC/DC converter, the radio interface, the microcontroller and the sensors.

In the WSNode module we also take into consideration the energy lost due to the sub unitary efficiency of the conversion, for the various current loads that the node exercises. In order to calculate the energy consumption of the wireless sensor node we take into consideration the power consumption of the radio transceiver in transmit and receive modes. Also, for the CPU, we consider the power consumption in stand-by and active mode. We obtain an energy consumption of the node for the 30 seconds period of 159.18 mJ, out of which 158.30 mJ

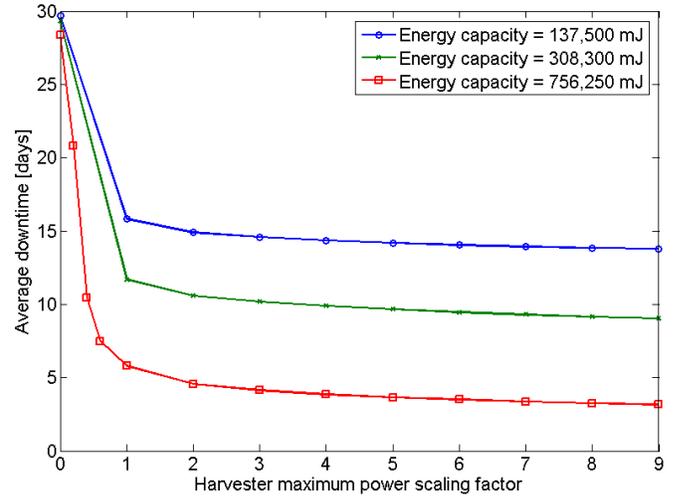


Fig. 6. Variation of the average downtime in a given time interval of 30 days with respect to the maximum power scaling of the harvester, for three different energy storage capacities.

are consumed in the 2.33 seconds of activity and the rest of 0.88 mJ in the other 27.67 seconds. These energy consumption values are obtained for an operation duty cycle of 7.76%.

Since the model can handle only integer values, we need to discretize the contributions of the consumed and the generated energy. For this we choose as energy unit 10 mJ. Choosing an energy step of 1 mJ, for example, would result in a more accurate, yet in a longer computation time with PRISM, due to the fact that the state space becomes one hundred times bigger, since both the variable that expresses the maximum capacity of the energy buffer element and the number of states of the harvester gets affected by this discretization. One execution step of the model corresponds to 10 seconds of the physical time length.

V. DESIGN SPACE EXPLORATION

A. Definition

Using our modeling framework we are able to infer two types of properties: probabilities of certain events, such as the probability of running out of power, and reward-based properties, to use the PRISM terminology, where we get expected values of various quantities, such as the average downtime.

The parameters a designer can tune are:

- i) the capacity of the energy buffer element.
- ii) the harvester maximum power scaling factor (or simply called, the harvester scaling factor) with respect to the baseline harvester with an area of $5.5 \times 15 \text{ cm}^2$ from Figure 2, which generates at most 90 mW. For example, a panel with a scaling factor

of two, has a maximum attainable power twice bigger than the baseline. A scaling factor of zero means there is no panel mounted on the system. Once the designer determines the maximum power of the panel, he can decide what should be the dimensions of the solar harvester. It is important to note that since the solar panel can be considered a current source [1], the voltage of the panel, and thus the power, is proportional to the area of the harvester, assuming the solar panel is composed of identical photovoltaic cells, and the light conditions are homogeneous.

- iii) the initial state of the harvester, `initHarvesterState`, at the beginning of the operation of the system. This parameter is an integer number between 0 (for 0 mJ) and 9 (for 90 mJ). For example, the energy level generated in a time step by a panel with `initHarvesterState` set to 4 corresponds to 40 mJ, which can be the level obtained at 11AM during the day.
- iv) the operation duty cycle, represents the fraction of time the node is in active state over the total period (which is by default, 30 seconds). As already stated, the duty cycle impacts on the power consumption of the node. To adjust the duty cycle in our modeling framework, we change the duration of the task in which the radio listens to perform synchronization.

The properties we explore with PRISM are:

- i) the probability that the system runs out of power;
- ii) the downtime of the node for a given interval of time;
- iii) the lifetime of the system.

Unless otherwise specified, the parameters of the model have the following values: the energy buffer has a capacity of 137,500 mJ (which corresponds to two 22F, 2.5V super-capacitors put in series), the operation duty cycle is 7.76% (corresponding to a 2.25 long seconds radio listening task), the harvester we use is the baseline panel (harvester scaling factor = 1) and `initHarvesterState=4` (i.e., the harvester generates initially 40 mJ, in a time step).

B. Exploration Results

We determine some of the aforementioned properties for several system models with different parameters.

In Figure 5 we show the variation of the probability of running out of power at least once for various given periods of time. We do this for three different storage capacities and we can observe the larger the energy capacity the smaller the probability of running out of power for a fixed time interval. For small time intervals

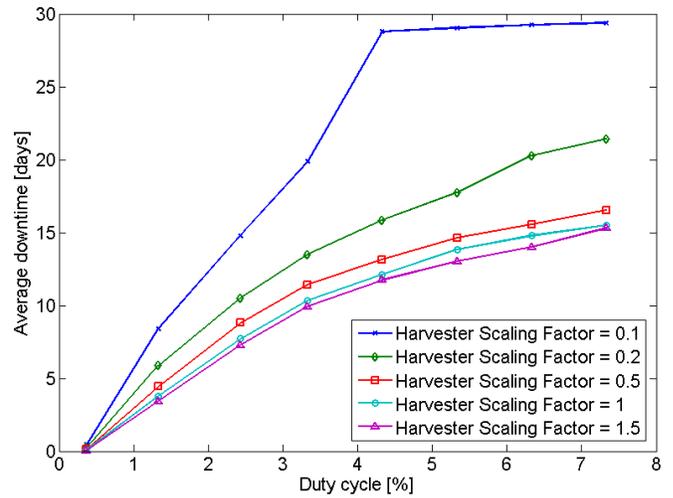


Fig. 7. Variation of the average downtime for a 30 days interval with respect to the operation duty cycle, for several maximum power scalings of the harvester.

(3 to 11 hours, depending on the capacity of the buffer) we obtain a zero probability of running out of power, because, even if we assume the worst case scenario (experiencing night during the whole time interval, thus leading to zero energy produced by the solar panel), there is enough energy in the storage element to sustain the uninterrupted operation of the node. After that, the probability tends asymptotically to 1, and we can see that for a period of 83.3 hours it has very similar values of 0.998, 0.99 or 0.974, for energy storage capacities of 68,750, 137,500 and 206,250 mJ, respectively. The designer can use this graph in order to determine which capacity to choose for the energy storage. For example, if we want to attain 10 hours of uninterrupted operation with a probability of at least 99%, then, according to the graph, we can choose a capacity of 206,250 mJ (which corresponds to three 22F, 2.5V super-capacitors put in series).

When considering an energy harvesting node, designers are interested in the downtime (from which we can directly compute the availability) of the node, depending on the environmental conditions. That is why we explore in Figure 6 the expected downtime of the node for a given time period of 30 days as a function of the harvester scaling. We can see that when the harvester scaling factor is zero (the node acts like a simple battery-powered system), the node is down for more than 29 days. Adding the baseline harvester to the system produces a significant decrease (46%, 60% and 80% for the storage capacities of 137,500 mJ, 308,300 mJ and 75625 mJ, respectively) in the downtime. After this, the downtime reduces significantly less when increasing the scaling factor. For example, for an energy buffer

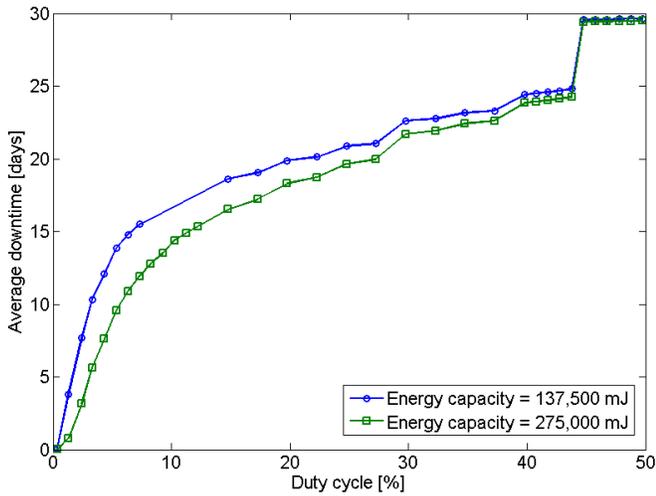


Fig. 8. Variation of the average downtime with respect to the duty cycle, for two different energy storage capacities.

capacity of 308,300 mJ, we reduce the downtime by 62% when we put a panel three times more powerful than the baseline. From there on, the decrease in downtime is smaller, about 1% with respect to the downtime of the baseline, for each increment in the scaling factor. Therefore, the designer should consider to use a solar panel at most three times bigger than the baseline, since further increase in the size is not improving significantly the availability of the system. The reason the curves tend asymptotically to a non-zero value is that the system can always experience blackout, due to the fact we have a finite energy storage capacity and there is always a non-null probability to have a long night period that can completely discharge the energy buffer (for this experiment, with the exception of the night, the energy consumed is smaller than the energy produced per time step, for a solar harvester equal or bigger than the baseline). The designer can use this graph in a straight form: for example, if he seeks to attain, on average, an autonomy of at least 85% within 30 days, for example, (meaning that the downtime is at most 4.6 days out of the 30 days), while minimizing the cost for the solar panel, with a node with a buffer capacity of 756,250 mJ, then he can find that the smallest required panel is twice bigger than the baseline, i.e., it has an area of 165 cm^2 .

Another exploration of the average downtime is with respect to the duty cycle, for various maximum power scalings of the harvester, presented in the Figure 7. Note that the curve for a harvester scaling factor of 0.1 starts saturating from a duty cycle of 4.33% onwards, because the average downtime tends asymptotically to 30 days, since it is calculated for a 30 days period, which means

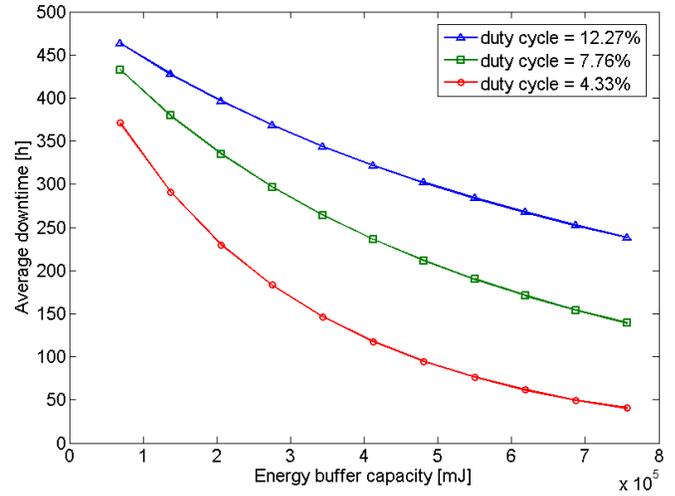


Fig. 9. Variation of the average downtime for a 30 days interval with respect to the energy buffer capacity, for three different duty cycles.

it cannot be bigger than this value. Otherwise, we can observe the non-linear, hyperbolic-like evolution of the downtime. This dependency is related to the fact that each blackout period of a downtime (and, thus, for the whole downtime) is getting smaller with a larger buffer capacity, and is inversely proportional to the duty cycle (we omit the detailed arguments, due to the lack of space).

Figure 8 presents another graph with downtime with respect to the duty-cycle, for various energy storage capacities. Note that as the duty cycle gets bigger, from 25% onwards, the shape of the curves is no longer hyperbolic, as it was in the previous graph, and the downtime increases more abruptly, since now the energy buffer can get depleted also during the day (below a duty cycle of 14.77% during the day, the energy consumed is smaller than the energy produced per time step, for a solar harvester equal or bigger than the baseline, which basically means that we can have a blackout only during the night; for a duty cycle bigger than 14.77%, we can actually have blackout also during the day). These tendencies of the curves are not intuitive, and, thus, makes our methodology indispensable for computing the expected values for the downtime.

In Figure 9 we present the variation of the downtime with respect to the increase of the energy buffer capacity. We can see the downtime decreases more rapidly with respect to the energy buffer capacity, for smaller duty cycles. This is due to the dependency already introduced two paragraphs above, which states the downtime is getting smaller with a larger buffer capacity, and is inversely proportional to the duty cycle.

Using the Figures 7, 8 and 9, the designer can choose

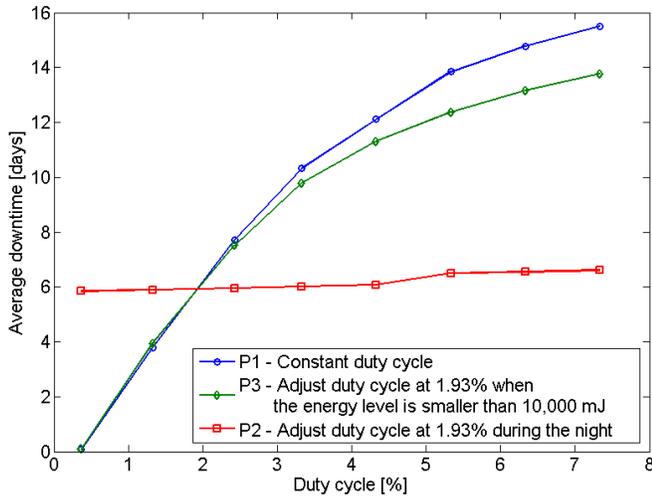


Fig. 10. Variation of the downtime with respect to the duty cycle for policies P1, P2 and P3.

a specific maximum duty cycle or minimum energy storage capacity in order to reach a desired downtime. For example, in order to have an availability of at least 94%, by looking at Figure 9, we conclude that we need to choose a buffer with a capacity of at least 756,250 mJ and an operation duty cycle of 4.33%.

As already stated, we can use the proposed methodology to compare various runtime policies, which can adapt at the observed parameters. To give an example, we propose three different policies for evaluation:

- P1 - the normal, constant duty cycle policy
- P2 - a policy that changes the duty cycle depending if it is night (i.e., no energy generated by the solar panel) or day
- P3 - a policy that adapts the duty cycle based on the state of charge of the storage element.

In Figure 10 we compare these three policies using the downtime metric. We can observe that policy P3 improves the availability of the system with respect to policy P1 (e.g., for a duty cycle of 7.33% we have a decrease in downtime of 11%), and P2 outperforms policy P3 (e.g., for a duty cycle of 7.33% we have a decrease in downtime of 52%), since it lowers the duty cycle during the whole night, which is better than lowering it only for the last 10,000 mJ of stored energy.

We emphasize again the advantage of our stochastic model by comparing it, for example, with a simple analytical method, which considers the weighted average energy generated per time step by the solar panel (if the weight coefficients are the steady state distribution of the Markov chain associated to the panel with the baseline size, then the average energy is 27.93 mJ), and the average energy consumed per time step (which, for

the normal duty cycle is 5.33 mJ). This means that at any point in time the node consumes less energy than it receives, which means the system will never run out of power. This analytic method is not accurate, since in reality we can encounter periods of night when the buffer has little energy stored, and the node can actually run out of power, for the same environmental conditions that are being modeled.

VI. CONCLUSIONS

Thanks to the harvesting technologies, designing autonomous sensor nodes and consumer embedded electronics that exploit the energy coming from the environment is becoming a feasible option. However, the design of such devices requires the careful selection of the components, such as the energy storage elements and the harvester device, according to the working environment and the features of the application (e.g., duty cycle or bandwidth) run on the system.

In this paper we presented a stochastic characterization framework for energy harvesting wireless sensor nodes. This allows designers to assess statistical properties of the system, such as the probability of achieving a given operation time, or the expected downtime or lifetime for various system configurations. In our experimental results we have shown how to use the proposed framework to tune various system parameters (e.g., energy buffer capacity and harvester size, operation duty cycle) of a solar powered wireless sensor node running a weather monitoring application, according to the concrete power characteristics of the node and of the solar harvester device.

REFERENCES

- [1] A. Aziz, K. Kassmi, F. Olivi, and A. Martinez. Symbolization of the electric diagram of the marketed solar panels in the Orcad-Pspice environment. 2006.
- [2] G. Barrenetxea, H. Dubois-Ferriere, R. Meier, and J. Selker. A weather station for SensorScope. In *Demo Session, In Information Processing in Sensor Networks (IPSN 2006)*, 2006.
- [3] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi. Discrete-time battery models for system-level low-power design. *IEEE Trans. Very Large Scale Integr. Syst.*, 9(5):630–640, 2001.
- [4] D. Bhaduri and S. K. Shukla. Nanoprism: a tool for evaluating granularity vs. reliability trade-offs in nano architectures. In *GLSVLSI '04: Proceedings of the 14th ACM Great Lakes symposium on VLSI*, pages 109–112, New York, NY, USA, 2004. ACM Press.
- [5] A. E. Şuşu, M. Magno, A. Acquaviva, D. Atienza, and G. De Micheli. Reconfiguration Strategies for Environmentally Powered Devices : Theoretical Analysis and Experimental Validation. *Computer Science Transactions on High-Performance Embedded Architectures and Compilers (HiPEAC)*, 2007.
- [6] H. Dubois-Ferriere. Sensorscope presentation at NCCR-MICS WG2, 2005.

- [7] H. Dubois-Ferrière, R. Meier, L. Fabre, and P. Metrailler. TinyNode: A Comprehensive Platform for Wireless Sensor Network Applications. In *Information Processing in Sensor Networks (IPSN 2006)*, 2006.
- [8] E. N. M. Elnozahy, L. Alvisi, Y.-M. Wang, and D. B. Johnson. A survey of rollback-recovery protocols in message-passing systems. *ACM Comput. Surv.*, 34(3):375–408, 2002.
- [9] V. A. Graham, K. G. T. Hollands, and T. E. Unny. A time series model for kt with application to global synthetic weather generation. *Solar Energy Vol. 40. No. 2. pp. 83-92*, 1988.
- [10] X. Jiang, J. Polastre, and D. E. Culler. Perpetual environmentally powered sensor networks. In *IPSN*, pages 463–468, 2005.
- [11] A. Kansal, J. Hsu, M. Srivastava, and V. Raghunathan. Harvesting aware power management for sensor networks. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 651–656, New York, NY, USA, 2006. ACM Press.
- [12] A. Kansal, D. Potter, and M. B. Srivastava. Performance aware tasking for environmentally powered sensor networks. *SIGMETRICS Perform. Eval. Rev.*, 32(1):223–234, 2004.
- [13] M. Kwiatkowska, G. Norman, and D. Parker. Prism 2.0: A tool for probabilistic model checking. *QEST*, 00:322–323, 2004.
- [14] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *Formal Methods in System Design*, 29:33–78, 2006.
- [15] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava. Heliomote: enabling long-lived sensor networks through solar energy harvesting. In *SenSys*, page 309, 2005.
- [16] C. Moser, L. Thiele, D. Brunelli, and L. Benini. Adaptive power management in energy harvesting systems. In *DATE '07: Proceedings of the conference on Design, automation and test in Europe*, pages 773–778, New York, NY, USA, 2007. ACM Press.
- [17] H. Nfaoui, H. Essiarab, and A. Sayigh. A stochastic markov chain model for simulating wind speed time series at tangiers, morocco. *Renewable Energy* 29 1407-1418, 2004.
- [18] B. T. Nguyen and T. L. Pryor. A computer model to estimate solar radiation in Vietnam. *World Renewable Energy Congress, Pages 1274 - 1278*, 1996.
- [19] D. Niyato, E. Hossain, and A. Fallahi. Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: Performance analysis and optimization. *IEEE Transactions on Mobile Computing*, 6(2):221–236, 2007.
- [20] G. Norman, D. Parker, M. Z. Kwiatkowska, S. K. Shukla, and R. Gupta. Using probabilistic model checking for dynamic power management. *Formal Asp. Comput.*, 17(2):160–176, 2005.
- [21] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *Pervasive Computing, IEEE*, 4(1):18–27, 2005.
- [22] P. Poggi, M. Muselli, G. Notton, and A. Louche. Forecasting and simulating wind speed in Corsica by using an autoregressive model. *Energy Conversion and Management, Volume 44, Pages 3177 - 3196*, 2003.
- [23] P. Poggi, G. Notton, M. Muselli, and A. Louche. Stochastic study of hourly total solar radiation in Corsica using a Markov model. *International Journal of Climatology, Volume 20, Issue 14, Pages 1843 - 1860*, 2000.
- [24] J. Polastre, J. Hill, and D. Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM Press.
- [25] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. Design considerations for solar energy harvesting wireless embedded systems. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 64, Piscataway, NJ, USA, 2005. IEEE Press.
- [26] S. Roundy, E. S. Leland, J. Baker, E. Carleton, E. Reilly, E. Lai, B. Otis, J. M. Rabaey, P. K. Wright, and V. Sundararajan. Improving power output for vibration-based energy scavengers. *Pervasive Computing, IEEE*, 4(1):28–36, 2005.
- [27] Semtech. Semtech XE1205 transceiver - <http://www.semtech.com/pc/downloadDocument.do?id=769>.
- [28] S. K. Shukla and R. K. Gupta. A model checking approach to evaluating system level dynamic power management policies for embedded systems. In *HLDVT '01: Proceedings of the Sixth IEEE International High-Level Design Validation and Test Workshop (HLDVT'01)*, page 53, Washington, DC, USA, 2001. IEEE Computer Society.
- [29] R. M. Spotnitz. Battery modeling. *The Electrochemical Society's Interface*, 14(4):39–42, Winter 2005.
- [30] L. Technology. LT1615/LT1615-1 Micropower Step-Up DC/DC converters in ThinSOT.