RAPPORT DE RECHERCHE

L R I

A RECURSIVE MODEL FOR BATTERY LIFETIME ESTIMATION IN WIRELESS SENSOR NETWORKS

RAHME J / AL AGHA K

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

09/2009

Rapport de Recherche N° 1525

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

A Recursive Model for Battery Lifetime Estimation in Wireless Sensor Networks

Joseph Rahmé[†] * and Khaldoun Al Agha [†] *
[†] LRI, University of Paris-Sud, CNRS
91405 Orsay Cedex, France
* INRIA
Email: {rahme,alagha}@lri.fr

***Abstract*—Since sensor nodes use batteries as their source of energy, energy-based routing becomes an important requirement to extend network lifetime. This routing is done using the nodes remaining energy information. Nonetheless, this information cannot be estimated using built-in primitives in nodes powered by a lithium-ion battery. The reason is that primitives rely on the decreasing battery voltage to estimate the node residual energy while a lithium-ion battery maintains an almost constant voltage during its lifetime. In this paper, we introduce an efficient battery model that estimates the remaining energy of a node independently of its battery chemistry. Our model is based on an existing battery model that uses the current consumption during a state and its duration to estimate the battery remaining energy. This model is very accurate, however, it cannot be implemented in a sensor node as it requires complex computations and large lookup tables. Therefore, we present a recursive approximation of this model that requires low memory and simple computation, while maintaining the original model accuracy. We show by simulation the validity of our model.**

I. Introduction

The recent advance in wireless communications allowed the development of tiny sensor nodes, available at low prices, capable of communicating in short distances. Sensor nodes are designed to last many years without any human intervention while using batteries. For this reason, the nodes must have a low duty cycle and energy expenditure. Moreover, sensor nodes use a built-in algorithm to discover their neighbors in order to form a network. Once the network is formed, each node senses its environment to collect a particular information (temperature, vibration,...) and relays it to a central node called *the sink*. Hence, we need an energy efficient routing protocol. Energy-based routing are very popular and these protocols rely on the nodes remaining energy information to route packets towards the sink. Consequently, the battery energy information becomes crucial for extending the lifetime of sensor networks. Having this in mind, we address the issue of estimating a node remaining energy. We consider that sensor nodes are powered by lithium-ion batteries, since a lithium-ion battery capacity is almost constant for different discharge power and temperature [1].

In practice, a sensor node is equipped with a primitive that can read the battery voltage, allowing the prediction of the battery remaining energy. This technique is valid only for a node powered by an alkaline battery, because the battery voltage varies proportionally to its remaining energy. However, this technique is not useful in a node powered by a lithium-ion battery, as the lithium battery maintains an almost constant voltage during its lifetime. To deal with this issue, we introduce a battery model that accurately estimates the remaining energy of a node battery, independently of the battery chemistry. Our battery model is based on the model presented in [2] which considers the following effects:

- *Rate capacity*: if a battery is discharged with a current of a magnitude greater than the battery rated current, then its lifetime decreases [3], [4].
- *Recovery effect*: if a battery is discharged with small currents separated by idle periods, it is possible to recover some of its energy [5].

The original model allows a node to predict the time-to-failure of its battery, given a current load profile. It relates the battery lifetime to the load and needs two parameters to be estimated, α and β. The α parameter represents the total charge in the battery when it is fully charged and β the rate at which the active charges are filled at the electrode surface.

Even so, this model cannot be implemented in a sensor node in its original form for the following reasons:

- it requires complex computations and large pre-computed lookup tables
- it needs large memory size to save the history of all node states (loads magnitude and their duration)

The contribution of the present work consists in an recursive version of the model presented in [2], that needs low memory and is simple to compute. The recursive model approximates the nodes remaining energy based on the previously computed value and updates it periodically.

In the rest of this paper, we provide an overview of the related work in section II. Section III presents the notations used to describe our model. The original model is simplified to obtain a recursive model in section IV. Section V describes the simulation results and section VI presents the conclusion.

II. Related Work

Models describing the behavior of the battery were presented in previous work. These models were divided into four

This work is supported by the ANR-OCARI project.

categories: physical, empirical, abstract and mixed [6].

The physical models [7] are the most accurate with a predicted battery lifetime very close to the experimental data. They involve a high computational complexity and require an in-depth knowledge of the battery chemistry. These two properties make them difficult to be implemented in a sensor node, with limited memory and computational ability.

The empirical models require low computational complexity and do not need any information about the battery chemistry. The work in [8] achieves a medium accuracy in estimating the battery remaining energy and it needs calibration for each temperature. This model could not be a good solution to our problem as it lacks of precision. A statistical method was presented in [9]. It uses experimental data to express the battery voltage in terms of the delivered capacity. The model gives quick predictions for the battery lifetime with low precision. We discard these models as we need to estimate the remaining energy accurately for energy-based routing.

The abstract models of the battery behavior use mathematical models related to the physical characteristics of the battery. In [4], the battery behavior is represented using a discrete time transient stochastic process. The process begins in state N (containing N charges) and proceeds to the next state, *N-i*, depending on the number *i* of charges that has been used. It considers both the rate capacity and recovery effects, and represents the recovery as a decreasing exponential. However, this model concentrates only on charge recovery and does not account for other battery non linearities.

The mixed models combine a high-level representation of the battery and provide analytical expressions based on physical laws as in [2], [10]. These models predict the lifetime of a battery with high precision and complexity. We have chosen to simplify the model in [2], as it requires only two parameters to be estimated. Other existing work estimates the battery energy in sensor networks precisely, i.e. [11], with a long computing time. In fact, if we need a high precision battery model, the computational complexity will be high. The model in [2] combines the load profile $i(\tau)$ and the parameter β to compute $\sigma(t)$, which is the charge consumed by the battery at time t. It has the following form:

$$\sigma(t) = \int_0^t i(\tau)\, d\tau + 2\sum_{m=1}^{\infty} \int_0^t i(\tau) e^{-\beta^2 m^2 (t-\tau)} d\tau \quad (1)$$

where β is the rate at which the active charge carriers are replenished at the electrode surface.

The first term in (1) determines the consumption of the load having a current magnitude $i(\tau)$ during a period [0,t]. The second term represents the non-linear behavior of the battery. To resume, $\sigma(t)$ calculates the apparent charge lost by the battery for a given load and the battery is considered depleted when $\alpha - \sigma(t) = 0$. Still there is two issues that makes the model not implementable in a sensor node: its *complex computation* and *large memory requirement*. We deal with these issues in section IV.

III. Model Description

In this section, we define the different assumptions and notations needed to introduce our model.

In general, a sensor node has a low duty cycle: it is active for a small duration and goes to sleep the rest of the time. It is important to determine a node state, so we can have an estimate about the current magnitude in this particular state. In general, a node could be in one of the following states:

1) Transmission: the node has all its components activated (antenna, micro-controller). It modulates the information to send it over the medium
2) Reception: The radio interface and micro-controller are active. The node listens to the medium for decoding the information
3) Idle: The node has all its components active and is idle
4) Sleep: The application (micro-controller) is active and all the other components are inactive
5) Deep sleep: All the node components are in the sleep mode, including the micro-controller

In our recursive model, we divide the time into intervals of duration Δ (see figure 1). During an interval, a node can be active for a duration δ_k, i.e., the duration of the current I_k ($\delta_k \in]0, \Delta]$). I_k represents the constant current value during active state δ_k and t_k is the time at which the current I_k is applied. The current is not considered null during the sleep state and δ_{kr} is the sleep duration within Δ. Moreover, the recursive model is Δ-based, which means that the energy of the battery is updated each Δ. To define our model, we use

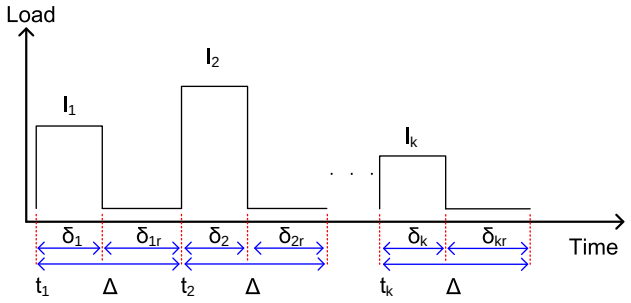

Fig. 1: Current profile

the following notations:

- Δ, is the period for energy updates,
- $\sigma(n\Delta)$ is the apparent charge consumed by node battery at $n\Delta$ in $mA\text{-}min$,
- α is the theoretical capacity of the battery in $mA\text{-}min$, and β is the same as in the original model ($min^{-\frac{1}{2}}$). α and β are estimated using the lifetime of the battery measured with continuous current discharge. We will detail the estimation method in section V,
- $C(n\Delta)$ is the battery charge level at $n\Delta$ in $mA\text{-}min$,
- $L_n = n\Delta$ is the time in terms of Δ

Implemented at the MAC layer, our model can easily deduce

the duration of a state by using a simple timer that is reinitialized at each change in the node state.

IV. Battery Model for Sensor Nodes

In this section, we first simplify the form of the summation in the original model and optimize the memory usage, in order to introduce later the recursive form of our battery model.

A. Simplifying the original model

In this section, we take advantage of some properties available in sensor nodes to simplify the computation of the original model. In general, a sensor node consumes a small current during a state and the current magnitude varies slightly during any state. Figure (2) shows the current needed by a

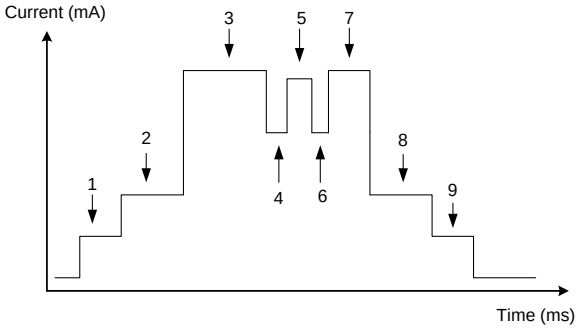

Fig. 2: Current consumption of a sensor node

CC2430 node to operate [12]. The node start up (goes to sleep) in intervals 1-2 (8-9), transmit and receive a packet from 3 to 7. Therefore without loss of generality, we consider that the current can be approximated by the average of the current magnitudes during a state. This means that the current consumption in any state is constant. Moreover, a sensor node in a beacon mode is active every beacon period to listen, receive or transmit data. Consequently, our choice to consider a constant current value during δ_k is justified.

Using this assumption, we let $i(\tau) = I_k$ in (1) and the constant I_k can be brought out of the integrals. Thus (1) reads,

$$\sigma(L_n) = \sum_{k=1}^{n} I_k \ \delta_k + 2 \ \sum_{k=1}^{n} I_k \ A(L_n, t_k + \delta_k, t_k) \quad (2)$$

where the function A is defined as

$$A(L_n, x, y) = \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (L_n - x)} - e^{-\beta^2 m^2 (L_n - y)}}{\beta^2 m^2}.$$

The evaluation of the second term in (2) requires a long computation time and a large lookup table for the exponential calculation. The reason is that a sensor node can only perform basic math operations (addition, subtraction, multiplication and division) and he needs to access the memory very often to store intermediate results. In fact, for an acceptable accuracy in computing A, m can go from 1 to 10. In this case, the A function alone requires 10 subtraction operations between two exponential terms, in addition to 10 divisions with the square of β and m. This without mentioning the product in the power of the exponential and the summation of all the terms. Furthermore, at time L_n, we need to save all the states k prior to L_n with information I_k, Δ_k, t_k related to every state. Then, the model becomes impossible to implement in a sensor node and this what motivates us to find a simplified form of A (section IV-B) in addition to resolving the memory usage problem (section IV-C).

B. Simplification of the sum

As mentioned above, $A(L_n, x, y)$ is very complex to compute and requires a large memory space. We need to find a simple and accurate mathematical approximation of $A(L_n, x, y)$, to be able to implement the model in a sensor node.
In the battery model, β is known a priori (estimated offline) and it can be brought out the sum. We also separate the two terms of the difference in A, to obtain the following form:

$$f(t) = \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t}}{m^2} = \sum_{m=1}^{\infty} \frac{1}{m^2 (e^{\beta^2 t})^{m^2}}, \quad (3)$$

which approximation can be used for both terms of A. Now, we begin to find the approximation for $f(t)$.
It is known that,

$$\sum_{m=0}^{\infty} x^m = \frac{1}{1-x} \quad (4)$$

We want m to be in the denominator, so we integrate both sides of (4) with respect to x (see [13])

$$\sum_{m=1}^{\infty} \frac{x^m}{m} = \int_0^x \frac{1}{1-y} dy = log(\frac{1}{1-x}). \quad (5)$$

If we replace x by $\frac{1}{x}$ in (5), we obtain:

$$\sum_{m=1}^{\infty} \frac{1}{m x^m} = log(\frac{x}{x-1}), |x| > 1. \quad (6)$$

By developing $\sum_{m=1}^{\infty} \frac{1}{m x^m}$, (6) becomes

$$log(\frac{x}{x-1}) = \frac{1}{x} + \frac{1}{2x^2} + \frac{1}{3x^3} + \cdots + \frac{1}{m x^m}. \quad (7)$$

We may write (7) to make $\sum_{m=1}^{\infty} \frac{1}{m^2 x^{m^2}}$ appear

$$log(\frac{x}{x-1}) = \sum_{m=1}^{\infty} \frac{1}{m^2 x^{m^2}} + \frac{1}{2x^2} + \frac{1}{3x^3} + \frac{1}{5x^5} + \cdots,$$

then the value of the sum $\sum_{m=1}^{\infty} \frac{1}{m^2 x^{m^2}}$ will be

$$\sum_{m=1}^{\infty} \frac{1}{m^2 x^{m^2}} = log(\frac{x}{x-1}) - \frac{1}{2x^2} - \frac{1}{3x^3} - \cdots. \quad (8)$$

Moreover, we know that $|e^{\beta^2 t}| > 1$, since $\beta > 0$ (model parameter) and $t > 0$ (time). Using (8), (3) reads

$$\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t}}{m^2} = log(\frac{e^{\beta^2 t}}{e^{\beta^2 t} - 1}) - \frac{1}{2 \times (e^{\beta^2 t})^2} - \cdots. \quad (9)$$

Since $\beta^2 \cdot t > 0$,

$$\forall n \in \mathbb{N}^*, n \cdot e^{n\beta^2 t} > n,$$

and the reciprocal of the inequality is

$$\frac{1}{ne^{n\beta^2 t}} < \frac{1}{n}.$$

For any t, the ratio $\frac{1}{ne^{n\beta^2 t}}$ has $\frac{1}{n}$ as an upper bound. In addition, following a certain t, the term $e^{\beta^2 t}$ becomes significant and $\frac{1}{ne^{n\beta^2 t}} \ll \frac{1}{n}$. Thus, the approximation $log(\frac{e^{\beta^2 t}}{e^{\beta^2 t}-1})$ is a good one for $f(t)$ starting from a certain value of t. We will call this approximation $f_2(t)$. In what follows, we find the sum approximation for small t that we call $f_1(t)$.

We know that the Jacobi theta function is defined as:

$$\vartheta(z;\tau) = \sum_{m\in\mathbb{Z}} e^{\pi i m^2 \tau + 2\pi i n z}. \tag{10}$$

For $z = 0$ and $\tau = it$, (10) becomes

$$\vartheta(0;it) = \Theta(t) = \sum_{m\in\mathbb{Z}} e^{-\pi m^2 t}, \tag{11}$$

where $\Theta(t)$ is called the theta function. We need to find an approximation of $f(t)$ for small t, for this reason let us remind the Poisson summation formula:

$$\sum_{m\in\mathbb{Z}} \hat{f}(m) = 2\pi \sum_{m\in Z} f(2\pi m), \tag{12}$$

where $\hat{f}$ is the Fourier transform of the function $e^{-ax^2}, (a > 0)$ defined over $\mathbb{R}$
Using (12), we can obtain the following expression [14]:

$$\forall\ t > 0, \Theta(t) = \frac{1}{\sqrt{t}} \cdot \Theta(\frac{1}{t}) \tag{13}$$

Having (11) and (13), we can write $\sum_{m\in\mathbb{Z}} e^{-\beta^2 m^2 t}$ in terms of Θ. The main reason for this is to estimate the sum from which we can deduce $f(t)$

$$\sum_{m\in\mathbb{Z}} e^{-\beta^2 m^2 t} = \theta(\frac{t\beta^2}{\pi}) = \sqrt{\frac{\pi}{t\beta^2}} \cdot \Theta(\frac{\pi}{t\beta^2}) \tag{14}$$

We develop (14) to find the form $\sum_{m=1}^{\infty} e^{-\beta^2 m^2 t}$:

$$1 + 2 \cdot \sum_{m=1}^{\infty} e^{-\beta^2 m^2 t} = \sqrt{\frac{\pi}{t\beta^2}} \cdot \left(1 + 2 \cdot \sum_{m=1}^{\infty} e^{-\frac{\pi^2 m^2}{t\beta^2}}\right)$$

We obtain the final form of our sum, which is:

$$\sum_{m=1}^{\infty} e^{-\beta^2 m^2 t} = \frac{1}{2} \cdot \left(\sqrt{\frac{\pi}{t\beta^2}} - 1\right) + \sqrt{\frac{\pi}{t\beta^2}} \cdot \sum_{m=1}^{\infty} e^{-\frac{\pi^2 m^2}{t\beta^2}} \tag{15}$$

We can notice that for small values of t, the first term $\frac{1}{2} \cdot \left(\sqrt{\frac{\pi}{t\beta^2}} - 1\right)$ is a good approximation of the original sum. The reason is that the sum $\sum_{m=1}^{\infty} e^{-\frac{\pi^2 m^2}{t\beta^2}}$ is negligible for small values of t and as a consequence the second term of (15). However, when t increases significantly the approximation becomes inaccurate since the first term $\frac{1}{2} \cdot \left(\sqrt{\frac{\pi}{t\beta^2}} - 1\right)$ alters to negative as $\sqrt{\frac{\pi}{t\beta^2}} \ll 1$.

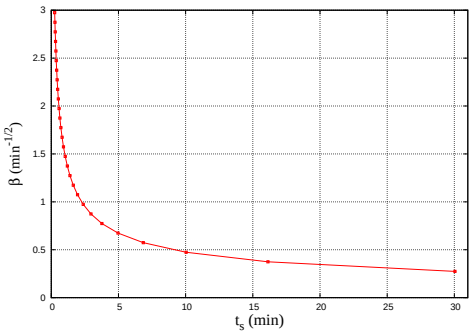

Fig. 3: t_s for different value of the parameter β

Therefore, for small values of t, we will have

$$\sum_{m=1}^{\infty} e^{-\beta^2 m^2 t} \approx \frac{1}{2} \cdot \left(\sqrt{\frac{\pi}{t\beta^2}} - 1\right). \tag{16}$$

In our case, we need to find the approximation of $f(t)$ for small values of t. By integrating (16) with respect to t, we obtain,

$$\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t}}{-\beta^2 m^2} + C \approx \sqrt{\frac{\pi}{\beta^2}} \cdot \sqrt{t} - \frac{t}{2} \tag{17}$$

For $t = 0$ in (17), we get, $C = \frac{\pi^2}{6\beta^2}$.
After simplification, (17) reads,

$$\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t}}{m^2} \approx \frac{t\beta^2}{2} - \beta\sqrt{\pi t} + \frac{\pi^2}{6} \tag{18}$$

For the rest of this paper, we will use $f_1(t)$, $f_2(t)$ to designate $\frac{t\beta^2}{2} - \beta\sqrt{\pi t} + \frac{\pi^2}{6}$ and $\beta^2 \cdot t - log(e^{\beta^2 t} - 1)$ respectively. Additionally, we still have to define for a value of the β parameter, when to use $f_1(t)$ and $f_2(t)$. We will call t_s, the time at which the approximation of $f(t)$ switches from $f_1(t)$ to $f_2(t)$.

$$f(t) = \begin{cases} f_1(t) & = \frac{t\beta^2}{2} - \beta\sqrt{\pi t} + \frac{\pi^2}{6} \quad , \text{if } t \in [0, t_s[\\ f_2(t) & = \beta^2 \cdot t - log(e^{\beta^2 t} - 1) \quad , \text{if } t \in [t_s, \infty[\end{cases}$$

To calculate t_s, we find the time t for which the difference between $f_1(t)$ and $f_2(t)$ is minimal, i.e., the time where the two functions are too close. It can be computed by calculating the derivative $d'(t)$ of $f_1(t) - f_2(t)$ with respect to t and finding its root, i.e. the value of t_s. $d'(t)$ has the following form:

$$d'(t) = \frac{-\beta^2 e^{\beta^2 t}}{e^{2\beta^2 t} - 2e^{\beta^2 t} + 1} + \frac{\sqrt{\pi}}{4\beta\sqrt{t^3}} \tag{19}$$

For a given β, we can calculate t_s numerically using the Newton-Raphson method. Figure (3) shows t_s for different values of β.

TABLE I: Number of operations

Operation	Original form	Approximated form
+	9	2
-	20	3
*	40	5
/	10	1

Table (I) shows the number of operations for the original form of A and our approximation. One can clearly see the simplification brought to the original form of A, in terms of computation and memory usage. Also, half of the subtraction operations in A are done between exponential terms requiring the use of a lookup table. Yet, in our approximation we use only one lookup table for the logarithm function because the square root can be approximated by a linear function. The linear approximation involves one multiplication and an addition.

Despite our approximation of the sum A, we still have the memory issue to handle in order to implement the model in a sensor node. We deal with this problem in the next section.

C. Reduce the memory usage

To estimate the remaining energy at L_n, the model needs the information on all previous states prior to L_n (see equaton 2). In this case, the information (I_k, t_k, δ_k) regarding the previous states must be saved in memory permanently. The former condition makes the model unimplementable in a sensor node with a limited memory. Thus, we introduce a recursive version of the original model to optimize the memory usage and computation. We also define the two following sets

- $S_\delta = \{\delta_k \mid k = \{1, \cdots, n\}\}$, is the set of all the possible states duration sorted in an ascending order,
- $S_\xi = \{\xi_k(n) \mid k = \{1, \cdots, n\}\}$, is the set of all the ratios $\xi_k(n)$ of the state durations δ_k,

where,

$$\xi_k(n) = \frac{A(L_{n+1}, t_k + \delta_k, t_k)}{A(L_n, t_k + \delta_k, t_k)}. \tag{20}$$

1) Introduce the recursive model: Using the original model, we use the current magnitude I_1 and the pulse duration δ_1 to calculate the energy consumed at Δ,

$$I_1\ \delta_1 + 2\ I_1\ A(\Delta, \delta_1, 0). \tag{21}$$

At time 2Δ, the remaining energy estimation needs the previous state history, in this case states 1 and 2:

$$I_1 \cdot \big(\delta_1 + 2\ A(2\Delta, \delta_1, 0)\big) + I_2 \cdot \big(\delta_2 + 2\ A(2\Delta, \Delta + \delta_2, \Delta)\big) \tag{22}$$

In equation (22), the time L_n in function A related to I_1, is incremented by Δ at each energy update period. For instance, at $L_2 = 2\Delta$, $I_1 \cdot \big(\delta_1 + 2\ A(2\Delta, \delta_1, 0)\big)$ and at $L_3 = 3\Delta$, $I_1 \cdot \big(\delta_1 + 2\ A(3\Delta, \delta_1, 0)\big)$ and so on. Therefore, the following question raises now: is it possible to avoid computing $I_1 A(n\Delta, \delta_1, 0)$ other than in the first state Δ ? To answer this question, we try find a relation between $A(L_{n+1}, \delta_1, 0)$ and $A(L_n, \delta_1, 0)$. We begin first by expressing $A(L_{n+1}, \delta_1, 0)$ as follows:

$$A(L_{n+1}, \delta_1, 0) = \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 L_{n+1}} \left(e^{-\beta^2 m^2 \delta_1} - 1\right)}{\beta^2 m^2}. \tag{23}$$

Then, we take $e^{-\beta^2 m^2 \Delta}$ out of the sum to retrieve the form of $A(L_n, \delta_1, 0)$,

$$A(L_{n+1}, \delta_1, 0) \leq e^{-\beta^2 \Delta} \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 L_n} \left(e^{-\beta^2 m^2 \delta_1} - 1\right)}{\beta^2 m^2}$$

$$A(L_{n+1}, \delta_1, 0) \leq e^{-\beta^2 \Delta} \cdot A(L_n, \delta_1, 0)$$

which implies

$$\xi_1(n) = \frac{A(L_{n+1}, \delta_1, 0)}{A(L_n, \delta_1, 0)} \leq e^{-\beta^2 \Delta} \tag{24}$$

$\xi_1(n)$ is the ratio performed for the current of duration δ_1. Since β is the parameter of the model and Δ is a constant, the upper bound of $\xi_1(n)$ is a constant. This propriety is logic as our A function is a sum of the terms from 1 to ∞ and from a certain t, the new terms added to the first sum terms becomes negligible. We can also calculate a constant υ (using a linear regression) such that $e^{-\beta^2 \upsilon \Delta}$ is the upper bound of $\xi_1(n)$,

$$\sup(\xi_1(n)) \leq e^{-\beta^2 \upsilon \Delta}. \tag{25}$$

After simplification, we will obtain the following form:

$$\frac{log(\sup(\xi_1(n)))}{-\beta^2 \Delta} \leq \upsilon \tag{26}$$

Now, we plot $\xi_1(n)$ for different values of β (fig.4) to visualize the value of n for which the ratio becomes a constant. On the

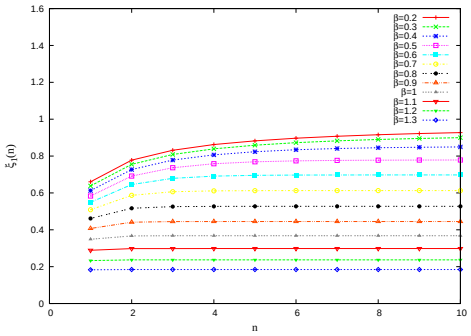

Fig. 4: $\Delta = 1$ min, $\delta_1 = 0.1$ min

one hand, we notice that for $\beta \geq 1.1$ and an appropriate Δ value, the ratio $\xi_1(n)$ is constant for all n. This ratio permits to compute the A function related to a current at any time step $n\Delta$ without the history of previous states information. On the other hand, the ratio is not constant for small values of β. With $\beta = 0.2$ for instance, we need to save nine values in memory $\{\xi_1(1), \cdots, \xi_1(9)\}$, to be able to use the recursive model. However, in practice the batteries possess a minimum β of 0.6

$min^{-\frac{1}{2}}$. $\xi_1(n)$ can be used to represent the variations of $\xi_k(n)$, because the variation among different δ_k is not significant. We will develop how to compute the value of Δ in the next section but now we want to obtain the recursive form of our battery model. We begin by computing the energy consumed using (2) at L_1 (Δ), $L_2(2\Delta)$. The apparent charge consumed is given by the following two equations:

$$\begin{aligned}\sigma(\Delta) &= I_1\ \delta_1 + 2\ I_1\ A(\Delta, \delta_1, 0)\\ \sigma(2\Delta) &= I_1\ \delta_1 + I_2\ \delta_2 + 2\ I_2\ A(2\Delta, \Delta+\delta_2, \Delta)\\ &\quad + 2\ I_1\ A(2\Delta, \delta_1, 0)\\ &= I_1\ \delta_1 + I_2\ \delta_2 + 2\ I_2\ A(2\Delta, \Delta+\delta_2, \Delta)\\ &\quad + 2\ \xi_1(1)\ I_1\ A(\Delta, \delta_1, 0)\end{aligned}$$

As we can notice, the computation for the charge consumed by the current I_1 can be done once at Δ and can be concluded using $\xi_1(n)$ in the next state. The same property applies for all the current $I_2, \cdots, I_n$ and allows us to avoid storing states history. Therefore, the size of the memory needed to save the states history can be freed. We introduce the recursive model solution by computing the energy consumed at 2Δ function of $\sigma(\Delta)$:

$$\begin{aligned}\sigma(2\Delta) &= I_1\ \delta_1 + I_2\ \delta_2 + 2\ I_2\ A(2\Delta, \Delta+\delta_2, \Delta)\\ &\quad + 2\ I_1\ \xi_1(1)\ A(\Delta, \delta_1, 0)\end{aligned} \tag{27}$$

In (27), $\xi_1(1) = \frac{A(2\Delta,\delta_1,0)}{A(\Delta,\delta_1,0)}$ is known a priori. We can also conclude from (27) that $2\ I_1\ A(\Delta, \delta_1, 0) = \sigma(\Delta) - I_1\ \delta_1$. Thus, (27) reads

$$\begin{aligned}\sigma(2\Delta) &= I_1\ \delta_1 + I_2\ \delta_2 + 2\ I_2\ A(2\Delta, \Delta+\delta_2, \Delta)\\ &\quad + \xi_1(1)\ \left(\sigma(\Delta) - I_1\ \delta_1\right).\end{aligned}$$

The general recursive form of the our battery recursive model is

$$\begin{aligned}\sigma(L_n) &= \sum_{k=1}^{n} I_k\ \delta_k + \lambda \cdot \left(\sigma(L_{n-1}) - \sum_{k=1}^{n-1} I_k\ \delta_k\right)\\ &\quad + 2 \cdot I_n \cdot A(L_n, L_{n-1}+\delta_n, L_{n-1})\end{aligned} \tag{28}$$

where $\sigma(L_1) = I_1\ \delta_1 + 2\ I_1\ A(\Delta, \delta_1, 0)$ is the model initial value and $\lambda = \xi_1(n) \approx \xi_2(n) \approx \xi_3(n) \approx \cdots \approx \xi_n(n), \forall n \in N^*$ with an appropriate Δ value. We study the complexity of equation (28) in terms of the operations needed to compute the remaining energy. The first term in (28) is the sum $\sum_{k=1}^{n} I_k\ \delta_k$. The evaluation of this term requires one multiplication during n ($I_n\ \delta_n$), considering the previous multiplication $\sum_{k=1}^{n-1} I_k\ \delta_k$ are done progressively during previous states. We need to save at each energy update the multiplication of the present current to the sum of previous states.

The second term in (28), $\lambda \cdot \left(\sigma(L_{n-1}) - \sum_{k=1}^{n-1} I_k\ \delta_k\right)$ is composed of the charge consumed during previous states $\sigma(L_{n-1})$ and the sum $\sum_{k=1}^{n-1} I_k\ \delta_k$. These terms are saved in memory to calculate their subtraction which is multiplied by λ. The last term is the most complex compared to the first ones. This term is composed of two terms with the form $\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 t}}{\beta^2 m^2}$ with $t > 0$. Recall $f(t)$ from (3), we can write function A in terms of $f(t)$ and we obtain:

$$A(L_n, L_{n-1}+\delta_n, L_{n-1}) = \frac{f(\Delta-\delta_n)}{\beta^2} - \frac{f(\Delta)}{\beta^2}. \tag{29}$$

β and Δ are two parameters of the model, which means that $\frac{f(\Delta)}{\beta^2}$ can be computed offline and it equal to a constant c_0

$$\frac{f(\Delta)}{\beta^2} = \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 \Delta}}{\beta^2 m^2} = c_0.$$

The first term of (29) can be computed using the approximation of $f(t)$

$$\frac{f(\gamma)}{\beta^2} = \begin{cases} \frac{\gamma}{2} - \frac{\sqrt{\pi}}{\beta} \cdot \sqrt{\gamma} + \frac{\pi^2}{6\beta^2} & ,if\ \gamma \in [0, t_s[\\ \gamma - \frac{log(e^{\beta^2\gamma}-1)}{\beta^2} & ,if\ \gamma \in [t_s, \infty[\end{cases}$$

where $\gamma = \Delta - \delta_n$. The ratio $\frac{\pi^2}{6\beta^2}$ is also a constant (c_1) that is kept in memory and needs no computation to estimate its value. The other ratio $\frac{\sqrt{\pi}}{\beta}$ is a constant c_2. We still have the square root $\sqrt{\gamma}$ that is easy to compute using a linear approximation in the vicinity of a value a. In fact, if γ varies around constant a, the approximation of the square root is $\sqrt{\gamma} \approx \sqrt{a} + (\gamma - a)\frac{1}{2\sqrt{a}}$ where $\sqrt{a}$, $\frac{1}{2\sqrt{a}}$ are computed offline. This assumption is realistic because the duration of a state does not vary a lot between different states.

D. Optimal period Δ

The Δ parameter is important in our recursive model. It is the time at which a node updates its energy. If the duration Δ is too short, the energy will be updated very often in vain because between two successive states the energy consumed is not significant. Our conclusion results from the fact that nodes have low duty cycle and use small current. For this reason, we want to find the optimal period Δ that maintains the model precision while considering the maximum variation in state duration. In this case, the condition in (30) should be minimal:

$$\max_{i=\{2,\cdots,n\}} \left(\xi_1(n) - \xi_i(n)\right) \tag{30}$$

We consider that the current durations in S_δ are sorted in an ascending order and that δ_1 (δ_n) is the smallest (highest) pulse. This makes $\xi_1(n) - \xi_i(n) \geq 0, \forall i \neq 1$. We also notice that $\xi_i(n)$ converges to a constant value at high n values. In addition, the difference $\xi_1(n) - \xi_i(n)$ is maximal for $n = 1$, which allows us to replace $\xi_i(n)$ by $\xi_i(1)$ and $\xi_1(n)$ by $\xi_1(1)$ in (30) to obtain

$$\max_{i=2,\cdots,n} \left(\xi_1(1) - \xi_i(1)\right). \tag{31}$$

We conclude from the above that the maximum of the difference between $\xi_1(n) - \xi_i(n)$ is equal to $\xi_1(1) - \xi_n(1)$. Thus, the optimal parameter Δ to update the energy is the value that minimizes the difference $\xi_1(1) - \xi_n(1)$.

For instance, if $\Delta = 1\ min$ and the maximum pulse duration δ_n is $0.7\ min$, the difference $\xi_1(1) - \xi_n(1)$ is negligible for high β values. However, for small β values, the difference becomes significant (see figure 5).

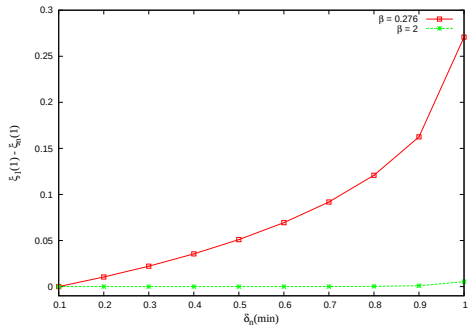

Fig. 5: $\Delta = 1\ min$

E. Error in the approximation

To prove the accuracy of our recursive model, we need to know what is the maximum error we introduce when replacing the original model by our approximation. Besides, it is very important to see the impact of an error in the first energy computations and its propagation in our estimation during the node operation.

We define the error between the original model and our approximation as

$$\epsilon_k(L_n) = |E_r(L_n, t_k + \delta_k, t_k) - E_a(L_n, t_k + \delta_k, t_k)|$$

where the $|.|$ denote the absolute value and

- E_r, is the energy consumed computed using the original model (2),
- E_a, is the approximated value of E_r, i.e. our recursive model (28),
- $E(x)$, is the error at time x .

Our model is recursive and to study the propagation of the error in our energy estimation, we compute the energy consumed from Δ to $n\Delta$ to find the form of the error,

$$\begin{aligned}
\sigma(L_1) &= I_1\ \delta_1 + 2\ I_1\ A(L_1, \delta_1, 0) \\
E(L_1) &= 2\ I_1\ \epsilon_1(\Delta) \\
\sigma(L_2) &= I_1\ \delta_1 + I_2\ \delta_2 + 2\ I_2\ A(L_2, L_1 + \delta_2, L_1) \\
&\quad + 2\ I_1\ A(L_2, \delta_1, 0) \\
E(L_2) &= 2\ I_2\ \epsilon_2(L_2) + 2\ I_1\ \xi_1(1)\ \epsilon_1(\Delta)
\end{aligned}$$

The expression of the error at L_n is:

$$E(L_n) = \eta \cdot \left(2 \cdot \sum_{k=1}^{n-1} I_k \cdot \epsilon_k(L_{n-1}) \right) + 2 \cdot I_n \cdot \epsilon_n(L_n) \quad (32)$$

where η is the upper bound of $\xi_1(n)$. We will see in the next section that the approximation error is almost negligible, even with a significant variation in the current magnitude and pulse duration.

V. SIMULATION

The quality of our model is evaluated with respect to results obtained by the low-level simulator DUALFOIL [7], [15]. The high quality of this simulator has been demonstrated in [16], [17]. DUALFOIL numerically simulates a set of partial differential equations that mimic the behavior of a lithium-ion cell. Over 50 parameters must be supplied as an input in order to have a simulated battery. We have modified these parameters to have the simulated battery in [6] which was demonstrated to be realistic. In the rest of this section, we will use this battery to perform our lifetime measurements and it will called dualfoil battery.

A. Determine α and β parameters

As mentioned earlier, we need to estimate two parameters α *mA-min* and β $(min^{-\frac{1}{2}})$ to implement our model. These parameters are determined using the lifetime information gathered by discharging the battery using constant currents (see table II).

Under constant-current discharge ($i(t) = I$), equation (1) reduces to the following form:

$$\alpha = \sigma(L) = I \cdot \left(L + 2 \sum_{m=1}^{\infty} \frac{1 - e^{-\beta^2 m^2 L}}{\beta^2 m^2} \right) \quad (33)$$

where L is the lifetime duration of the battery under a continuous current load of magnitude I and α is the total charge in the battery. Using (33), we fit the current values I for the set of the measured lifetime L in table II and we estimate the parameters (α, β) using least square estimation.

TABLE II: Battery simulated lifetime

Load current, mA	Lifetime, min	Load current, mA	Lifetime, min
2	21202.039	40	1052.170
5	8475.242	50	840.021
10	4234.405	60	698.589
15	2820.079	70	597.871
20	2112.915	80	522.869
28	1506.469	100	415.723

To have a good estimate of α and β, it is necessary to discharge the dualfoil battery with a wide range of load currents. We begin from small current values (current in the sleep state) to very high currents (while all the components are active). Our simulated battery has an $\alpha = 40027$ *mA-min* and $\beta = 0.276\ min^{-\frac{1}{2}}$ with an open-circuit voltage of $4.3V$ and the cutoff voltage is $3.2V$. The low value of β means that the battery exhibits strong non-linearity in contradiction to an ideal battery model. We introduce this model in the next section.

B. Ideal model

The ideal model considers that if a battery has a capacity of $640\ mAH$, it can deliver $640mA$ for one hour. The energy estimation using the ideal model is not accurate and we will prove this by comparing the ideal model lifetime estimation to the simulator predictions.

In order to compare the dualfoil battery lifetime with the ideal model, we must know the capacity of the our battery. Many techniques are used to estimate the capacity of the battery, we will use the one that determines the battery capacity when its voltage reaches $3.2V$ (cutoff voltage for

our dualfoil battery). For this reason, the dualfoil battery is discharged with a constant current (100 mA for example) until the battery voltage reaches $3.2V$, and we measure its lifetime. The battery lifetime is 6.928 hours and the battery estimated capacity is $100 \times 6.928 = 692.87\ mAH$. We apply the same method with all the currents in table II and we calculate the average to get the best battery capacity estimation. The dualfoil battery has a capacity of $701.55\ mAH$.

C. Performance evaluation

To prove the accuracy of our recursive model, we need to compare its lifetime estimation with the one given using the dualfoil simulator. Therefore, we have given a series of current profiles as an input to both the simulator and the recursive model, and we have computed the battery lifetime using both models. We will consider that Δ is the same in all our scenarios and it is equal to 60 seconds.

In the first scenario, we consider that a constant current load is applied to the battery for the first 10% of the interval Δ and that a 0.1 mA load is applied in the rest of this interval. The current profile is repeated every Δ until the battery is depleted. The current remains constant during all the battery duration and we repeat the same scenario for the currents ranging from 20 to 100. Using the recursive model, the battery is considered depleted when $\alpha - \sigma(L_n)$ is null, however, in the dualfoil simulator the battery is depleted when its voltage reaches the cutoff voltage. The results of our simulation can be seen in table III. In this table E represent the percentage of estimation error of every method compared to the result given by the dualfoil simulator. We can clearly notice that the error using the recursive model is approximately half of the one appearing when considering the battery as an ideal model. And we can prove that the ideal model is not very accurate to quantify the remaing energy of a battery. Despite this fact, many papers consider the battery as an ideal source when estimating the remaining energy of a sensor node. They simply calculate the energy consumed per node state (transmissionn reception, ...) using consumption measurement and update the battery remaining energy accordinly. If those estimations are used in energy-based routing, the routing protocol can use sensor nodes with low remaining energy, since the energy information is overestimated. Consequently, the sensor network lifetime will be shortened.

The second scenario is the same as the first one, except that the current $0.1\ mA$ is replaced by $0.0001\ mA$. The result of our simulation is shown in table IV. The main goal of this study is to find the effect of the current consumed in a node sleep state on the battery lifetime. In fact, we need to show that the assumption that current consumed in the sleep state can be neglected is not realistic. Figure 6 shows the percentage of error a battery model can make if it did not account the current consumed in sleep state. It can be clearly seen that for sensor nodes with low current consumption, neglecting the current consumed in a sleep state can affect considerably the energy estimation of the node. In addition, the error percentage decreases as the current increases since at high current the current in the sleep mode becomes negligible compared to the current in the active states.

TABLE III: Battery lifetime estimation (min)

Load, mA	DUALFOIL	Ideal model	E%	Recursive model	E%
20	18156.1	20140.1	10.92	19116	5.28
40	9249.1	10291.1	11.26	9751	5.42
60	6203	6911.8	11.41	6537	5.38
80	4664.1	5203	11.55	4912	5.31
100	3737.1	4171.7	11.62	3932	5.21

TABLE IV: Battery lifetime estimation (min)

Load, mA	DUALFOIL	Ideal model	E [1] %	Recursive model	E%
20	18866	21037.1	11.50	19978	5.89
40	9430	10520.9	11.56	9968	5.70
60	6283.1	7014.4	11.63	6636	5.61
80	4710	5261	11.69	4968	5.47
100	3766	4208.9	11.76	3967	5.33

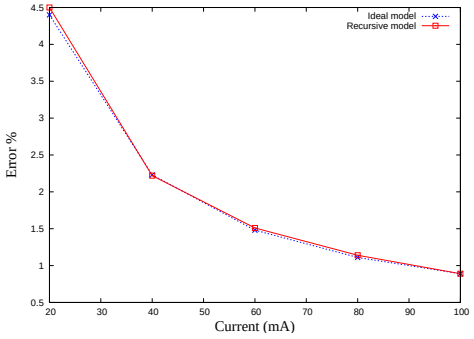

Fig. 6: Percentage of estimation error

The last scenario is the same as previous scenarios but with a random current ranging from $2\ mA$ to $100\ mA$ and with also a random duration that varies from 0.1 to 0.4 min during the interval Δ. Other than during the duration of the active pulse, the current is very low. The estimation of the lifetime using dualfoil simulator is $3203.240\ min$ and $3072\ min$ using our recursive model. The estimation error in the battery lifetime is 3.8%. The mean error between the original model and our approximation is 0.08% with a maximum peak of $10\ mA$-min.

VI. Conclusion

In this paper, we proposed a recursive battery model that estimates the remaining energy of a node battery. Although our model estimates the remaining energy in lithium-ion batteries, it can be used in any node independently of its battery chemistry. We have showed that our model can be implemented in a sensor node since it requires a simple computation and needs a small memory. In addition, the accuracy of our model is showed by a good estimation of the battery remaining energy information with different scenarios.

We have also proved in this paper that the assumption of considering the battery as an ideal model is not realistic. In fact, if a node consumes $c\ mAH$ in a state, it is not realistic

to update the battery energy after this state by subtracting c from the original battery capacity.

ACKNOWLEDGMENT

I would like to thank Professor Daler Rakhmatov for helping me in finding the right parameters for the dualfoil battery. A special thank also to Professor Jean-Christophe Leger for his remark that putted me on the right track to find the sum approximation.

REFERENCES

[1] "Energizer battery," http://data.energizer.com/PDFs/L91ULT_EU.pdf.

[2] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *ACM transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 277–324, August 2003.

[3] M. Doyle and J. Newman, "Analysis of capacity rate data for lithium batteries using simplified models of the discharge process," *J. Applied Electrochem.*, vol. 27, no. 7, pp. 846–856, July 1997.

[4] D. Panigrahi, C. Chiasserini, S. Dey, R. Rao, A. Raghunathan, and K. Lahiri, "Battery life estimation of mobile embedded systems," *Proceedings of the The 14th International Conference on VLSI Design*, pp. 57–63, 2001.

[5] T. Fuller, M. Doyle, and J.S.Newman, "Relaxation phenomena in lithium-ion-insertion cells," *J. Electrochem.*, vol. 141, no. 4, pp. 982–990, April 1994.

[6] R. Rao, S. Vrudhula, and D. Rakhmatov, "Battery modeling for energy-aware system design," *Computer*, pp. 77–87, December 2003.

[7] M. Doyle, T. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *J. Applied Electrochemical Soc.*, vol. 140, no. 6, pp. 1526–1533, 1993.

[8] M. Pedram and Q. Wu, "Design considerations for battery-powered electronics," *Proc. 36th ACM/IEEE Design Automation Conf.,ACM Press*, pp. 861–866, 1999.

[9] K. Syracuse and W. Clark, "A statistical approach to domain performance modeling for oxyhalideprimary lithium batteries," *Proc. 12th Ann. Battery Conf.Applications and advances, IEEE Press*, pp. 163–170, 1997.

[10] P. Rong and M. Pedram, "An analytical model for predicting the remaining capacity of lithium-ion batteries," *Proc. 2003 Design, Automation and Test in Europe conf. and Exposition, IEEE CS Press*, pp. 1148–1149, 2003.

[11] C. Ma, Z. Zang, and Y. Yang, "Battery-aware scheduling in wireless mesh networks," *Mobile Networks and Applications*, vol. 13, no. 1-2, pp. 228–241, April 2008.

[12] B. Selvig, "Measuring power consumption with cc2430 and z-stack," 2007.

[13] Graham, Knuth, and Patashnik, *Concrete mathematics*. Addison-Wesley, 1989.

[14] "http://www.dynamaths.com/docs/lecons/developpement_analyse_130.pdf."

[15] T. Fuller, M.Doyle, and J. Newman, "Simulation and optimization of the dual lithium ion insertion cell," *J. Applied Electrochemical Soc.*, vol. 141, no. 1, 1994.

[16] M. Doyle and J. Newman, "Comparison for modeling predictions with experimental data from plastic lithium ion cells," *J. Applied Electrochemical Soc.*, vol. 143, no. 6, 1996.

[17] P.Arora, M. Doyle, A.Gozdz, R.White, and J. Newman, "Comparison between computer simulations and experimental data for high-rate discharges of plastic-ion batteries," *J. Power Sources*, vol. 88, 2000.