

## Research Article

# A Hybrid Model for Accurate Energy Analysis of WSN Nodes

**Muhammad Mahtab Alam, Olivier Berder, Daniel Menard, Thomas Anger,  
and Olivier Sentieys**

*IRISA, University of Rennes 1, 6 Rue de Kerampont, BP80518, 22300 Lannion, Cedex, France*

Correspondence should be addressed to Muhammad Mahtab Alam, mahtab.alam@irisa.fr

Received 10 June 2010; Revised 30 October 2010; Accepted 10 January 2011

Academic Editor: Xiaorui Wang

Copyright © 2011 Muhammad Mahtab Alam 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.

Energy modeling is an important issue for designing and dimensioning low power wireless sensor networks (WSN). In order to help the developers to optimize the energy spent by WSN nodes, a pragmatic and precise hybrid energy model is proposed. This model considers different scenarios that occur during the communication and evaluates their energy consumption based on software profiling as well as the hardware components power profiles. The proposed model is a combination of analytical derivations and real time measurements. Firstly, the analytical model provides a global view of various elements of the link and MAC layers and shows their impact on the energy consumption. Secondly, the real-time measurements provide an accurate estimate of the power consumption of the software as well as the hardware platform. These experiments are particularly useful to understand the MAC layer mechanisms, such as wake up or data collisions for the preamble sampling category, and the energy wasted by collisions is evaluated. The presented model is validated under a specific setup with three different test cases. The results verify that the relative error is between 1 to 8%.

## 1. Introduction

Wireless sensor networks (WSNs) are characterized by collaborative information transmissions from multiple sensor nodes observing a physical phenomenon [1]. Battery-driven sensor nodes are easy to deploy, but they are severely constrained on physical size, cost, and energy efficiency. Since the most important constraint in WSN remains the energy consumption, it is of great concern to know that how to increase the lifetime of the sensor nodes. In this regard, it is prerequisite to know precisely the energy consumption of the sensor node. Since the energy consumed is not only due to the components at the individual layers [2], different partial cross-layer energy models such as [3–5], proposed limited interaction between lower layers such as DLL (Data Link layer), MAC (Medium Access Control) and PHY (Physical Layer).

The driving parameters of these layers in terms of energy consumption are idle listening, amount of collisions, retransmissions, overhearing, overheads, and the associated power consumption of the hardware components is the main

focus of the global energy evaluation. In the context of global energy consumption, the MAC layer plays a pivotal role in optimizing the WSN lifetime, because the dominant energy consumption is due to the radio chip, whose activity is controlled by the MAC layer [6]. Another fact is that the actual payload at the MAC layer is negligible with regards to the associated headers. As a consequence in typical WSN applications such as temperature monitoring, surveillance, and so forth, the radio wakes up at more frequent rate with regards to what it actually transmits, and that leads to a significant proportion of energy waste.

In this paper, we propose a scenario-based hybrid and precise energy model with the aim to optimize the global energy of a WSN. The energy model relies on the interaction of both analytical modeling and real-time measurements. Firstly, the analytical model provides a global view of various elements of the DLL and MAC layers and shows their impact on the energy consumption. Secondly, the real-time measurements provide an accurate estimate of the power consumption of the software as well as the hardware components of the platform [7].

The energy consumption is critical to the occurrence of the events and more precisely to their timings. For example, in distributed asynchronous sensor networks (DASN), a sensor node can wake-up at a particular instant and communicate with another node without any collision, but the same node can have a collision when it wakes up at another time instant. Consequently, the energy consumption for the same node is completely different. Therefore, in order to have deeper and realistic evaluation of energy consumption, it is important to identify different executing scenarios. The energy model explores the details of the energy consumption based on different scenarios that can occur during the communication between sensor nodes. Each scenario is executed with regard to particular instantaneous events and the energy consumed by these scenarios is calculated, which helps to evaluate precise and detailed energy consumption of the software and hardware. The presented model focuses on a specific MAC protocol of the preamble sampling category but it can be easily extendable in larger context to MAC protocols of this category.

The rest of this paper is organized as follows. In Section 2, we present the state of the art, followed by the analytical model of the DLL and MAC layers in Section 3. Further, in Section 4, we present the protocol stack implementation and the real-time power measurements of our platform. Section 5 is devoted to our scenario-based hybrid energy consumption modeling and its validation, while applications are proposed in Section 6. The paper ends with the conclusion and future work.

## 2. Related Work

Understanding power consumption is crucial for the design of autonomous nodes with the highest lifetime. During the last several years different power models have been presented and most of them are either only state-based with no or limited interaction of cross-layers, or only based on hardware components power consumption. Works in [8, 9] present energy models based on the consumption at the transmit and receive states and limited to radio consumption. Reference [10] presents an energy model which only takes the consumption of the hardware components into account, whereas [11] presents the software energy model. Also, couple of companies, for example, [12, 13], have presented similar power consumption models. The state-based model considers constant current levels in different states, which does not represent a precise measure, as they lack the inside details of the variations in the current levels.

Now, with regard to the global energy model, [14] presents the MAC And Physical LAYER Power (MAPLAP) model taking some cross-layer impacts into account, but which is limited and again based on the different states. Some other works propose new methods in [15, 16], to give accurate consumption models based on microprocessor platforms running on the widely used TinyOS event-driven operating system. These tools capture the detailed energy requirements of the CPU, radio, and every other peripherals, but the model presents a very general overview and lacks real-time measurements. The authors in [3] propose a very

interesting framework for an integrated layer design of the link and MAC layers of sensor networks in order to consider a global system for energy optimization of different layers. This is a solution to the fact that some components are not selected/designed knowing the impact of cross-layer interactions, especially concerning the power consumption evaluation.

The power and energy models in [8–10] are based on limited aspects and propose very general overview. The work proposed in [3] is in fact a framework for partial cross-layer integrated energy model, but it does not give a real measure of the energy. So, the bottom line is that there exists a deficiency for realistic energy model which could provide the details of the energy consumption and explain all the possible scenarios that can occur during the communication in sensor networks, which eventually leads us to the inside details of the state-based models.

## 3. Analytical Modeling

In this section the analytical characteristics of different low level layers with special focus on the MAC layer are presented. Different parameters such as *wake-up interval* for a node, *collision probability* and *amount of data transmitted from node  $i$  to node  $j$*  during a given time interval, have a direct influence on the presented energy model. For the context of this paper, the parameters considered fixed are the topology and geographic configuration, while other parameters such as data error rate, wake-up intervals, packet collisions, packet size, transmission power level, associated with the low level layers such as DLL and MAC layers, will be investigated through the model.

**3.1. Data Link Layer Model.** The link layer model is based on the error control strategy which can be applied mainly through three different strategies; automatic repeat request (ARQ), forward error correction (FEC) and hybrid automatic repeat request (HARQ). The objective of our link layer model is to manage the automatic repeat request (ARQ) and the forward error coding (FEC) to ensure a reliable link. The quality criterion considered is the tolerance to binary errors. Now, based on the bit error probability  $p_b$  [17, 18], the packet error rate (PER) for the ARQ and FEC schemes are presented in the following.

The ARQ scheme is based on cyclic redundancy check for the error detection. Considering that all the errors can be detected, the PER for the payload  $pl$  (which is the actual data in the complete data packet) can be represented as

$$PER_{ARQ}(pl) = 1 - (1 - p_b)^{pl}. \quad (1)$$

A detection-only strategy will not tolerate any binary error, while a correction strategy associated with a simple checksum for detection will tolerate one error. In the case of forward error correction coding, the link layer can be designed using several techniques such as block and convolutional codes. For the case of block codes, considering that the payload is  $pl$  and if the tolerance (i.e., the number of errors that can be corrected by FEC) is noted as  $tol$ , then

the retransmission rate (RT) for the communication between node  $i$  and node  $j$  for a tolerance  $\text{tol}$  is calculated as

$$\text{RT}_{i,j}(\text{tol}) = 1 - \left( \sum_{k=0}^{\text{tol}} C_{\text{pl}}^k \cdot p_b^k \cdot (1 - p_b)^{\text{pl}-k} \right), \quad (2)$$

where  $C_{\text{pl}}^k = \binom{\text{pl}}{k}$  is the combination between the  $\text{pl}$  and  $k$ .

Let us consider that the bit error probability ( $p_b$ ) (for the binary convolutional code using the hard-decision decoding) can be represented with the upper performance bound as [19]

$$p_b \leq \frac{d}{dz} T(\text{Dist}, Z) \Big|_{Z=1, \text{Dist}=2\sqrt{p_s(1-p_s)}}, \quad (3)$$

where  $p_s$  is the probability of symbol error,  $d$  is the minimum distance between all pairs of code word sequence,  $Z$  is the number of branch transitions for bit 1 that is equal to 1 for binary convolutional code and  $T(\text{Dist}, Z)$  is the generating function of the code. The energy per bit consumption for various coding schemes against the minimum transmission power can be seen for the details in [20].

**3.2. Medium Access Control Model.** Medium Access Control (MAC) is the ability of a node to efficiently share the wireless medium with the other nodes in the network. The main objective of the MAC layer is to keep the energy consumption low by turning off the radio module as often as possible. In the design of energy aware MAC protocols, the main cause of energy consumption is idle listening, overheads, overhearing and collisions. Therefore, in order to achieve the energy efficiency these factors need to be minimized, but there exists a tradeoff for the optimal design. For example, the protocol with the aim to reduce idle monitoring and collisions demands extra synchronizations and overheads, whereas, reducing the overheads and synchronizations results in an increase in energy waste due to collisions.

In the context of energy efficient WSN protocols, preamble sampling methods are attractive options for light WSN traffic [6]. Preamble sampling MAC protocols are based on a nonscheduled mechanism without any synchronization among the nodes, which means that each node is completely independent of its own active/sleep strategy. Preamble sampling category includes energy efficient MAC protocols such as RICER (Receiver Initiated Cycled Receiver) [14], TICER (Transmitter Initiated Cycled Receiver) [14], LPL (Low Power Listening) [21], BMAC (Berkeley MAC) [21], XMAC [22] and WiseMAC [23]. These protocols reduce the cost of extra overheads (in comparison with scheduled based protocols) and synchronizations by having single or multiple preambles. In addition a protocol such as WiseMAC can adjust the duty cycle efficiently based on the 'wake-up time' of the neighbor nodes, which results in a great reduction of idle monitoring, preamble size and probability of collisions.

The general mechanism of a preamble sampling protocol is shown in Figure 1. As the initialization of the communication between sensors can be initiated by either a transmit or a receive node (depending upon the specific protocol being used), both cases are depicted in Figure 1.

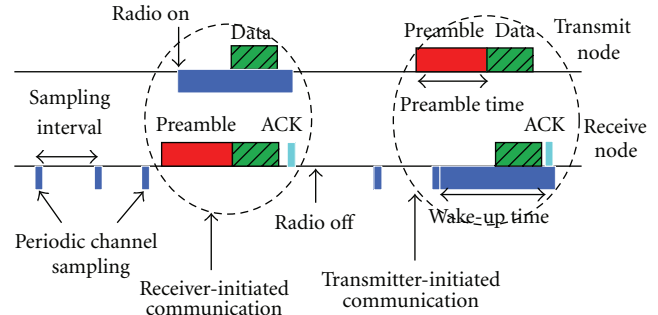


FIGURE 1: General mechanism of preamble sampling protocols. Every receive node wakes up with a specific sampling interval  $S_{\text{Int}}$ , and samples the channel. After channel sensing, if a node finds the preamble, it remains awake otherwise the node goes to sleep. The communication can be initiated by either a receive or a transmit node. If the communication is initiated by the receiver, then the receive node will send the preamble to the transmit node and the transmit node will respond with the data packet, whereas if it is initiated by the transmitter then the preamble will be sent by the transmit node followed by the data.

If the communication is initiated by the receiver then the receive node will send the preamble to the transmit node and the transmit node will respond with the data packet, whereas if it is initiated by the transmitter then the preamble will be sent by the transmit node followed by the data packet. The receive node wakes up periodically at a regular sampling interval to sense the channel activity and if it does not find any preamble it goes to sleep mode immediately. It is to be noted that preambles have to be long enough such that the intending receive/transmit node can be able to receive the preamble on the wake-up and further to keep the radio on for receiving the subsequent data packet.

In preamble sampling protocols, the wake-up time of one node is independent of the other nodes in the network, which means that two nodes can wake-up and send their preambles and subsequent data packets at the same time, which results in a *preamble collision*. In addition to *preamble collisions*, *data collisions* can occur due to randomly distributed sensing and also due to the hidden terminals, even though in different variants of preamble protocols, such as XMAC, WiseMAC, TICER and RICER, the data packet is being sent after channel sensing and initial rendez vous between the nodes. So, in order to derive the mathematical expressions for the probability of collisions, some parameters have to be defined. Let us consider  $n_r$  being the number of nodes in the radio range,  $S_{\text{Int}}$  the rate at which a receive node wakes up and samples the channel,  $P_{\text{Time}}$  the time of the preamble, and ( $W_{\text{UTime}}$ ) the time during which the receive node remains awake for the data as shown by the transmitter-initiated communication in Figure 1. In this paper, we consider that the network traffic is approximated as a Poisson distribution with a mean value of data generated per second per node denoted as  $\lambda$  and  $T_{\text{Obs}}$  is the observation time to observe the traffic. Then,

the total number of successful rendez vous  $n_{rdv}$  during  $T_{obs}$  is

$$n_{rdv} = n_r \frac{T_{obs}}{WUTime}. \quad (4)$$

Then, the occupation rate of the channel is

$$OR = n \frac{WUTime}{SInt}, \quad (5)$$

where  $n$  is the number of nodes in the network. OR must be inferior to 1 in the general situation or equal to 1 when the channel is saturated. So for a particular node, the probability of *preamble collision* is equal to

$$Pr_{PC} = (n_r - 1) \frac{WUTime}{SInt}. \quad (6)$$

After the general overview of the energy efficient preamble sampling category of MAC protocols, we will focus on cycled receiver protocols for a detailed MAC energy analysis. The chosen protocol is based on the principle of Receiver Initiated Cycled Receivers (RICER) [24, 25]. RICER is a pseudoasynchronous technique (also called cycled receiver) to realize rendez vous between wireless nodes. It means that nodes establish rendez vous on demand, but underlying there exists a periodic wake-up scheme. RICER MAC scheme can be used with very simple communication hardware with the following key features:

- (i) reduced preamble size and signals, Beacon, Data, and Acknowledgment Frames,
- (ii) low power pseudoasynchronous mechanism ideally suited for low traffic, that is, WSN,
- (iii) only the beacon frame is broadcasted, therefore it results in limited overhearing,
- (iv) reduced idle monitoring and collision with underlying self-periodicity,
- (v) RICER performance is better than its counter-part, Transmitter Initiated Cycled Receiver (TICER) under strong fading conditions [14].

Several variants of RICER and their comparison with Transmitter Initiated Cycled Receiver (TICER) have been presented in [14]. We have chosen RICER3 (3-way handshake), which is shown in Figure 2, because it contains less preambles (control signals) and its performance is comparable with other variants. RICER3 is a power optimized protocol for low traffic WSN application [14], but it can be also very effective for adaptive traffic load providing that the wake-up interval is adaptively adjusted.

**3.2.1. Collision Probability Model.** In order to evaluate the collision probability for this pseudoasynchronous protocol, the MAC layer has to be accurately described. In RICER3, as shown in Figure 2,  $N$ -data slots are placed to avoid the collision which happens when two nodes send the data to the same destination node at the same time in response to

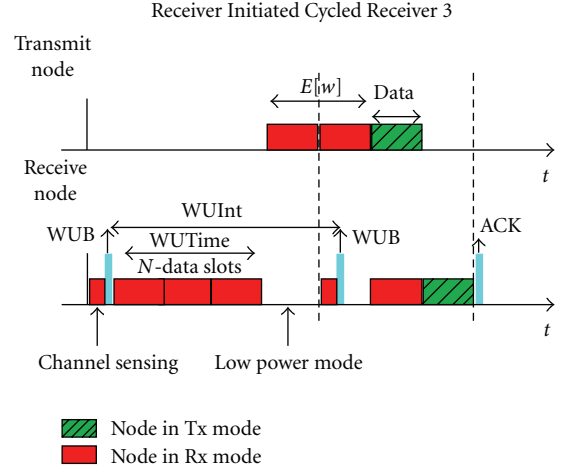


FIGURE 2: RICER3 is a 3-way handshake receiver initiated cycled receiver scheme. The  $N$ -data slots are placed to avoid the collision which happens when two nodes send the data to the same destination node at the same time in response to the wake-up beacon.  $E[w]$  is the expected waiting time,  $WUTime$  is the total wake-up time of the receiver,  $WUInt$  is the wake-up interval of each node,  $WUB$  is the wake-up beacon signal and  $ACK$  is the acknowledgment signal.

the wake-up beacon. The destination node wakes up with its own periodic interval and senses the channel for a very short time before broadcasting the wake-up beacon to avoid the control packet collisions on the channel, whereas the source node on reception of wake-up beacon also senses the channel for a short duration before transmitting the data to prevent the *data collision*. The *data collision* can occur when two or more source nodes select the same data slot of the destination node. It is being observed that the channel sensing time needs to be at least as long as twice the packet propagation time [25]. Now, to reduce the *data collision*, it is necessary to choose an appropriate number of data slots in RICER3. The more slots there are, the lower the *data collision* rate and thus the fewer retransmissions, but the more monitoring power wasted per wake-up (in the case of limited data transmission). These factors lead to a tradeoff in both power consumption and latency. It is observed that on a low traffic wireless sensor network application, the probability of three nodes to transmit at one node at the same time is very rare [7, 25]. So we have assumed in our collision model that there can be maximum two nodes that can transmit the data to the same destination node at the same time.

Two kinds of collision are possible in this pseudoasynchronous scheme.

- (i) The control packet collision, called wake-up collision (WUC), can happen at the initial phase of the rendez vous when two nodes wake-up in the receive mode at the same time; since both nodes found the channel as idle and therefore, they send the wake-up beacon at the same time.

- (ii) The data packet collision, called data collision (DC), is possible when two source nodes are intending to transmit the data to the same destination node. As they receive the wake-up beacon from the destination node, both source nodes find the channel being idle and transmit their data in the same data slot.

(a) *Wake-Up Collision.* In the RICER protocol, a node wakes up in the receive mode at every wake-up interval (WUInt) and sends the wake-up beacon (WUB). After this phase, the receive node waits for the duration of wake-up time (WUTime) to receive the response. Let us consider  $T_{\text{obs}}$  as the observation time to observe the traffic and  $n_r$  the number of nodes which are in the same radio range. Then, the probability of the wake-up collision is equal to the sum of collisions due to the  $n_r - 1$  nodes and due to the node which is hidden for a particular node and the expression can be calculated as

$$\Pr_{\text{WUC}} = (n_r - 1) \frac{\text{WUTime}}{\text{WUInt}} + \frac{\text{WUTime}}{\text{WUInt}} = n_r \frac{\text{WUTime}}{\text{WUInt}}. \quad (7)$$

(b) *Data Collision.* Following the Poisson law, the probability of  $i$  source nodes (out of  $n_r$  nodes which are within the radio range) intending to transmit to the same destination node at the same time is

$$\Pr(X = i) = \frac{(n_r \lambda \text{WUInt})^i}{i!} \exp(-n_r \lambda \text{WUInt}). \quad (8)$$

Focusing on one of these  $i$  source nodes, with  $N$  data slots, the probability that the data of the node does not collide with any others is

$$\Pr_{\text{success}} = \frac{N(N-1)^{i-1}}{N^i} = \left(\frac{N-1}{N}\right)^{i-1}. \quad (9)$$

Accordingly, the total probability of *data collision*  $\Pr_{\text{DC}}$  for this particular node in  $T_{\text{obs}}$  is

$$\Pr_{\text{DC}} = \sum_{i=2}^{n_r} \frac{(n_r \lambda \text{WUInt}/T_{\text{obs}})^i}{i!} \times \exp(n_r \lambda \text{WUInt}/T_{\text{obs}}) \left(1 - \left(\frac{N-1}{N}\right)^{i-1}\right). \quad (10)$$

One should note that  $\Pr_{\text{DC}}$  is expected to be higher than the actual *data collision rate*, because nodes sense the channel for a random time before transmitting a data. It also considers the collision due to the hidden terminal. So in this way, as long as the attempting source nodes are in the radio range, source nodes that sense later will detect the channel as busy and can back off.

3.2.2. *Analytical Analysis of Power Consumption in RICER3.* Let  $N$  be the number of data slots followed by each wake-up beacon (WUB), then the power consumption for RICER3 can be calculated as [25] follows:

$$P_{\text{tot}} = \Delta_{\text{tx}} P_{\text{tx}} + \Delta_{\text{rx}} P_{\text{rx}} + \Delta_{\text{mn}} P_{\text{mn}} + \Delta_{\text{aq}} P_{\text{aq}} + \Delta_{\text{sp}} P_{\text{sp}} + \sum E[N_{\text{turn}}] P_{\text{turn}} \quad (11)$$

with

$$\begin{aligned} \Delta_{\text{tx}} &= E[N_{\text{WUB}}] \text{WUInt} + \lambda' (T_{\text{DATA}} + T_{\text{ACK}}), \\ \Delta_{\text{rx}} &= \lambda' (\text{WUInt} + T_{\text{DATA}} + T_{\text{ACK}} - 3T_a), \\ \Delta_{\text{aq}} &= \lambda' (3T_a), \\ \Delta_{\text{mn}} &= E[N_{\text{WUB}}] N T_{\text{DATA}} + \lambda' E[W], \\ \Delta_{\text{sp}} &= 1 - \Delta_{\text{tx}} - \Delta_{\text{rx}} - \Delta_{\text{aq}} - \Delta_{\text{mn}}, \end{aligned} \quad (12)$$

where  $\Delta_{\text{tx}}$ ,  $\Delta_{\text{rx}}$ ,  $\Delta_{\text{aq}}$ ,  $\Delta_{\text{mn}}$ , and  $\Delta_{\text{sp}}$  are the percentage of the time each node spends in the transmit, receive, acquisition, monitoring and sleep states, respectively. These are the five different states represented in the MAPLAP model [24]. The term  $E[N_{\text{turn}}]$  is the expectation of the transition from one state to another state of a node within one second.

$P_{\text{tx}}$ ,  $P_{\text{rx}}$ ,  $P_{\text{aq}}$ ,  $P_{\text{mn}}$ ,  $P_{\text{sp}}$ , and  $P_{\text{turn}}$  are the power consumption levels in the transmit, receive, acquisition, monitoring, sleep and transition states, respectively.  $E[N_{\text{WUB}}]$  is the expected number of WUB transmitted during  $T_{\text{obs}}$ , and is equal to

$$E[N_{\text{WUB}}] = (1 - \Pr_{\text{busy}}) \frac{(1 - \lambda' (E[W] + 2T_{\text{DATA}} + 2T_{\text{ACK}}))}{T_{\text{obs}}}. \quad (13)$$

$\Pr_{\text{busy}}$  is the probability that a node detects the channel to be busy before transmitting the WUB, DATA or ACK frame and is expressed as

$$\Pr_{\text{busy}} = 1 - (1 - (E[N_{\text{WUB}}] \text{WUInt} + \lambda (T_{\text{DATA}} + T_{\text{ACK}})))^{n_r}. \quad (14)$$

In the case of channel being busy, the node backs off for one or multiples of WUInt. The channel is detected to be busy if any other packets are already transmitted on the network. After taking account of collisions, the actual traffic load becomes  $\lambda'$ , and can be calculated as

$$\lambda' = \frac{\lambda}{1 - P_c} \quad (15)$$

where  $P_c$  includes both  $\Pr_{\text{WUC}}$  and  $\Pr_{\text{DC}}$ .

$E[W]$  is the expected monitoring time (or expected waiting time) of the source node until it receives the WUB from the destination node. This monitoring time can increase in two cases. First, if the destination node finds the channel being busy before transmitting its WUB, in this case the transmitting node has to wait for the multiples of WUInt (but the transmitter can send its own WUB to avoid the longer latency for the whole network). Second is due to the collision of packets over the channel.

The power consumed during the switch between the states depends upon the expected number of transitions from one state to another. These can be *sleep to transmit* ( $E[N_{\text{SP-TX}}]$ ), *sleep to receive* ( $E[N_{\text{SP-RX}}]$ ), *transmit to receive* ( $E[N_{\text{TX-RX}}]$ ), *receive to transmit* ( $E[N_{\text{RX-TX}}]$ ). During one

second for RICER3 the expected number of transitions can be calculated as [25] follows:

$$\begin{aligned}
 E[N_{\text{SP-TX}}] &= 0, \\
 E[N_{\text{SP-RX}}] &= \frac{(1 - \lambda'(E[W] + 2T_{\text{DATA}} + 2T_{\text{ACK}}))}{\text{WUInt}} + \lambda(1 + 1), \\
 E[N_{\text{TX-RX}}] &= \frac{(1 - \lambda'(E[W] + 2T_{\text{DATA}} + 2T_{\text{ACK}}))}{\text{WUInt}} + \lambda, \\
 E[N_{\text{RX-TX}}] &= \frac{(1 - \lambda'(E[W] + 2T_{\text{DATA}} + 2T_{\text{ACK}}))}{\text{WUInt}} + \lambda(1 + 1).
 \end{aligned} \tag{16}$$

**3.3. Summary of Analytical Model.** To summarize the analytical model, we have presented the expressions for different parameters of the lower layers that have significant impact on the global energy consumption in WSN. Data link layer supports the model for error detection and correction and the error occurs during the transmission due to non ideal channel conditions. For that reason, the amount of retransmissions for a given tolerance is presented. The MAC layer is the core of our model; we consider the energy efficient preamble sampling category and we explore the collision probability models along with the power consumption for RICER protocol. This analytical model will be used along with the real-time power measurements to obtain the proposed hybrid energy model presented in Section 5.2.

## 4. Protocol Implementation and Real-Time Power Measurements

In this section, we explain different optimizations that have been achieved at each layer and the results of the power measurements based on our low power WSN hardware platform. The architectural block diagram of our WSN platform (PowWow [7]) is shown in Figure 3. PowWow is a hardware platform associated to a software architecture designed for a complete WSN solution. The hardware platform is, like many others, based on a low-power microcontroller and a radio transceiver. However, PowWow also includes new features which improve the energy efficiency with regards to state-of-the-art platforms: dynamic voltage and frequency scaling (DVFS) of the digital processing part and also coprocessing capabilities using a low-power FPGA (Field Programmable Gate Array). The detailed description of the hardware architecture and its utilization can be found in [20].

Our sensor network software is based on the embedded system Contiki, which is built around an event-driven kernel but provides optional preemptive multithreading that can be applied to individual processes [26], and more precisely on the protothread library [27], which enables the use of extremely light-weight, stackless threads. Protothreads allow to realize event-driven systems, and it has been shown that asynchronous processing is typical of sensor networks applications which perfectly suit the event-driven programming.

The main kernel of the program is an infinite loop that sequentially gives the processor to each process. After each loop of the main kernel, it is possible to record an execution trace which depends on external events and which can be identified and classified. The approach used for the power estimation is both accurate and simple, because it is based on the analysis of real code and includes the consumption of the communication component and the microcontroller. Therefore, after having identified the power consumption of each typical execution trace, the total software power only depends on the number of these typical traces.

**4.1. Optimizations of MAC Parameters.** The MAC protocol (RICER3) has been optimized in terms of packet size, collision reduction (wake-up collision and data) and effective utilization of dual channels in comparison to [14]. The *overhead* in the data and control packets is one of the main source of energy utilization. Therefore, these overheads have been reduced significantly in the actual implementation; the size of control packets, that is, wake-up beacon and acknowledgment, is reduced to four bytes, whereas the data packet is consisting of sixteen bytes, as shown in Figure 4.

*Two channels* have been used quite effectively, the *channel 1* works for the normal mode which supports low traffic such as temperature sensing, and the *channel 2* is used when there exists a heavy data traffic such as images to be transmitted in typical surveillance monitoring or software code update.

*Collision avoidance* and *idle monitoring* are the primary concerns for power reduction in pseudo asynchronous rendez vous schemes. In RICER3, there is significant power consumption due to wake-up collisions, as there are many WUB transmitted to have a successful rendez vous. Many nodes can wake-up at the same time and can result in collision mainly because of the clock drifts and the asynchronous nature of the protocol. To reduce these wake-up collisions and to reduce the bad situation where two or more nodes always wake-up at the same time, a *random delay* that is completely autonomous to each node has been introduced. This *random delay* brings the small offset which is enough to avoid WUB collision. Finally, the *idle monitoring* has been significantly reduced by several ways. The data packet is always ready at the transmitter as it received the WUB and due to the fact that the time to switch between the states is very short, therefore, the receiver does not need to wait for the  $N$  data slots as shown in Figure 2. In our implementation the receiver only waits for the *first byte* of the data packet. This reduces the unnecessary wait/monitoring of the receive node for  $N$  data slots as had been done in [14, 24, 25]. At the same time, the probability of collision in the absence of  $N$  data slots will remain the same as the *data collision* is independent of data slots and it only occurs when the two nodes transmit exactly at the same time. The optimized RICER3 is shown in Figure 5, by adding these optimizations in RICER3 (MAC Protocol) along with some improvements suggested in the results will make this protocol very efficient for low power WSN applications.

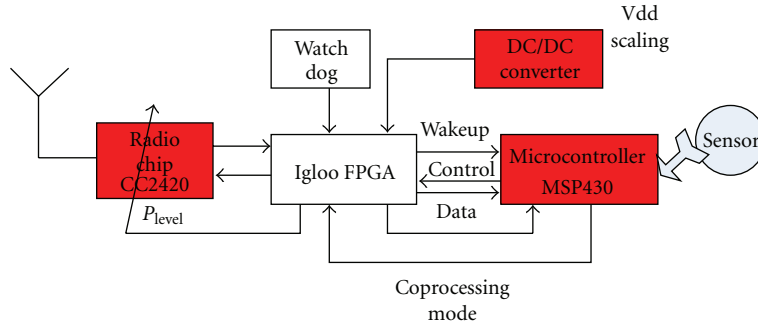


FIGURE 3: Hardware block diagram of PowWow. The key components in terms of power consumption are processing units (T.I MSP430, Actel IGLOO FPGA), DC/DC converter (used for voltage scaling), and the radio chip (T.I. cc2420).

| X        | Y        | CRC      | Type     | Mode     |
|----------|----------|----------|----------|----------|
| 8        | 8        | 8        | 3        | 5        |
| <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> |

Control packet (bits)

| Sx       | Sy       | Dx       | Dy       | F.Hx     | F.Hy     | L.Hx     | L.Hy     | CRC 1    | CRC 2    | N.Ack    | N.T      | T        | Mode     | B        | Temp     |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| 8        | 8        | 8        | 8        | 8        | 8        | 8        | 8        | 16       | 16       | 8        | 8        | 3        | 5        | 2        | 6        |
| <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> | <i>b</i> |

Data packet (bits)

FIGURE 4: The control packet (wake-up beacon and acknowledgment) consists of the geographical network address ( $x, y$ ) of the receive node which transmits the control packets, cyclic redundancy check (CRC) and the last byte contains the type of packet (3-bits) and the mode of communication (5-bits). The data packets consists of the network address of the source and destination as well as the hop-addresses of the next and last hop-nodes, along with few bytes for acknowledgment number, temperature, battery, and  $N$ .  $T$  is a frame buffer indicator.

**4.2. Platform Power Measurements.** PowWow hardware platform has been designed with modularity. It is composed of a central printed circuit board (PCB) and of different daughter-cards as mentioned in [7]. The real-time measurements are conducted with an Agilent N6705A DC Power Analyzer which is equipped with four channels which means that at one time the power consumption of four components/devices can be measured. The platform setup for measurements is constituted of three WSN PowWow motes connected to the DC Power Analyzer. The external power supply of 3.3 volts is used. Four modules of the N6705A are connected with the cc2420 and MSP430, to measure the current consumption of each component of the connected nodes. It is noteworthy that no DC/DC conversion is required with this configuration which reduces the perturbation of the measurements. The parameters that have been described in Section 3, such as *collision avoidance*, *wake-up*, and *data collisions*, have been identified through real-time measurements. Some general parameters used in the physical platform are shown in Table 1. These parameters are data length, MAC layer timing data, antenna gains, receive and transmit power levels of the radio transceiver chip.

The current consumption for the complete communication between three sensor nodes are shown in Figure 6. In

TABLE 1: System parameters of the physical platform and the timing values of the MAC protocol.  $T_{react}$  is the reaction time to switch the states,  $T_{sync}$  is the time to synchronize,  $T_{DF}(Rx)$  is the time to receive the data frame,  $T_{DF}(Tx)$  is the time to transmit the data frame.

|                  |                             |
|------------------|-----------------------------|
| $pl = 128$ bits  | $E[w](Tx) = (1.5-15)$ ms    |
| $G_t = -1.0$ dBi | $T_{react} = 0.1$ ms        |
| $G_r = -1.0$ dBi | $T_{sync} = 0.1$ ms         |
| $P_n = -95$ dBm  | WUInt = (0.3-3) s           |
| $F_c = 2.4$ GHz  | $T_{DF}(Rx) = (1.5-4.5)$ ms |
| $C = 3.10^8$ m/s | $T_{DF}(Tx) = 1$ ms         |
| $\alpha = 4$     | $P_t(0$ dBm) = 57.2 mW      |
| $P_r = 56.9$ mW  | $P_t(-20$ dBm) = 30.5 mW    |

the MAC protocol, since the transmission is initiated by the receiver, the transmitting node after waking up has to wait for the WUB from the receiver, before it starts transmitting the data. In our implementation, the maximum expected waiting time of the transmitter is 15 blocks, where each block can be between 0.3 s and 3 s depending on the application. In Figure 6, the combined total current consumption based on software and hardware (MSP430 and cc2420) for three WSN nodes are shown. In this configuration each node is connected to a single power analyzer module. The interesting

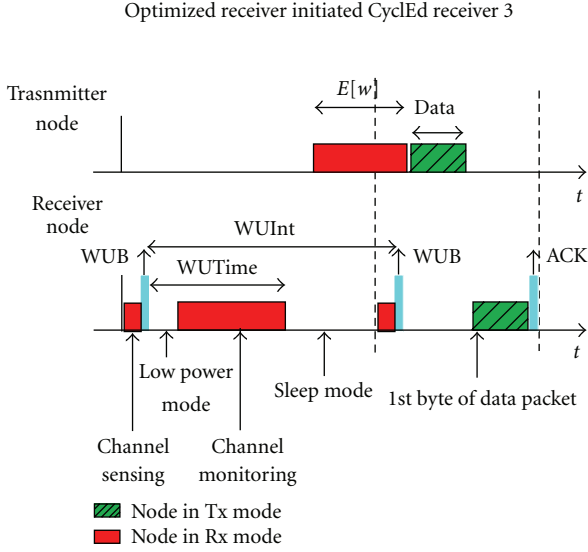


FIGURE 5: Optimized 3-way handshake receiver initiated cycled receiver scheme. The  $N$ -data slots are not necessary to avoid the collision, instead the receiver only requires the first byte of the data packet. Therefore, the idle channel monitoring is irrespective of the data slots but adjustable according to the application.

aspect of Figure 6 (coming from MAC Protocol) is that the transmitting nodes also send their own WUB at an interval of every *block*, so as to reduce the overall network latency and to improve the power consumption of idle monitoring of other neighboring nodes that want to communicate with the transmitting node. The rendez vous occurs between the nodes and after the complete communication all the nodes go to sleep mode.

The power consumption details of a rendez vous are shown in Figure 7. Two nodes are trying to transmit data to the same destination node and both are waiting for WUB in the receive mode. The receive node wakes up (according to its own wake-up interval) from the sleep mode and configures its software as well as hardware in calculation before transmission (CBT). The execution of this calculation can be different depending upon the exact mode of software and hardware. For example, if the regulator of the radio transceiver is off (sleep state), it requires complete restart. Similarly, if the microcontroller (software) is in deep sleep mode, it requires to boot from the start. When the WUB is received, the two nodes change their mode from receive to transmit and start transmit their data packet. Since both nodes sense the channel through clear channel assessment (CCA) before transmitting their data, in this particular example, the Tx node 2 finds the channel already occupied by Tx node 1 and hence it backs off. Therefore, the *data collision* is being avoided through channel sensing as shown in Figure 7. Meanwhile, the Tx node 1 continues its transmission which completes at the reception of ACK signal from the receive node.

Though Figures 6 and 7 show the total current consumption and provide the detail inside of the communication phases during rendez vous, the details of sleep, low power, active, transitions, states are not clearly identified.

For that matter, Figure 8 shows the details of software as well as hardware consumption. The software component (MSP430) consumes  $0.0 \mu\text{A}$ ,  $0.01 \mu\text{A}$ ,  $2.5 \text{ mA}$ ,  $3.7 \text{ mA}$ , in sleep, low power mode (LPM), transmit, and receive modes, respectively. Whereas the hardware component (cc2420) consumes,  $0.0 \mu\text{A}$ ,  $0.4 \text{ mA}$ ,  $17.6 \text{ mA}$ , and  $19.6 \text{ mA}$ , in sleep, LPM, transmit, and receive modes, respectively. It should be noted here that the time taken for the WUB transmission and for the data transmission is different (because of different packet size), and hence the energy consumption is different even though the level of the current drawn by the two is nearly the same.

Figure 9 shows the effect of *data collision*. In this case, two transmitting nodes intend to transmit at the same time instant to the same destination node, both nodes find the channel as idle and transmit their data, which results in *data collision* over the channel. In this specific example, both transmitters, after sending their data, switch to the receive mode and wait for the ACK signal from the receiver. Due to *data collision* over the channel, the receive node keeps waiting until 4 ms, which is three times more than the time during a normal successful communication. Similarly, both transmitters wait more than 3 ms for the ACK signal. First, the extra cost of this *data collision* at the two transmitters is found to be  $0.3699 \cdot 10^{-3} \text{ J}$  and  $0.3701 \cdot 10^{-3} \text{ J}$ , respectively, secondly at the receiver the energy is calculated as  $0.3888 \cdot 10^{-3} \text{ J}$ .

## 5. Scenario-Based Energy Consumption Modeling and Validation

The main work of this paper concerns with more precise energy consumption estimation of the communication between nodes which consists of both software (protocol stack) and hardware (physical platform), respectively. Traditionally the total energy consumption  $E_{\text{tot}}$  is the sum of three terms  $E_{\text{algo}}$ ,  $E_{\text{proc}}$  and  $E_{\text{amp}}$ .  $E_{\text{algo}}$  is the energy consumed by the microcontroller for the protocol layers and their control,  $E_{\text{proc}}$  is the part of the energy consumed by shaping of the signal (at the PHY layer) and it involves all the analog and digital signal processing including the conversion from analog to digital and vice versa. Finally,  $E_{\text{amp}}$  is the additional part that depends on the output power and it is mainly due to the power amplifier.

The evaluation of the term  $E_{\text{algo}}$  is done by measuring the processing time of the different tasks of the loop traces. The terms  $E_{\text{proc}}$  can be calculated by the power consumption of communication components in the receive or transmit modes, whereas  $E_{\text{amp}}$  depends on the transmit power level  $P_{\text{Level}}$ . Let us note  $P_r$  the power consumption of the signal processing components in receive mode,  $P_t(P_{\text{Level}})$  and  $P_t(\text{min})$  the power consumption due to signal processing components in the transmit mode for an output power of  $P_{\text{Level}}$  and for a minimal transmission power, respectively.  $T_r$  is the time spent in receive mode and  $T_t$  is the time spent in transmit mode, then  $E_{\text{proc}}$  and  $E_{\text{amp}}$  can be expressed as

$$\begin{aligned} E_{\text{proc}} &= P_r \cdot T_r + P_t(\text{min}) \cdot T_t, \\ E_{\text{amp}} &= (P_t(P_{\text{Level}}) - P_t(\text{min})) \cdot T_t. \end{aligned} \quad (17)$$



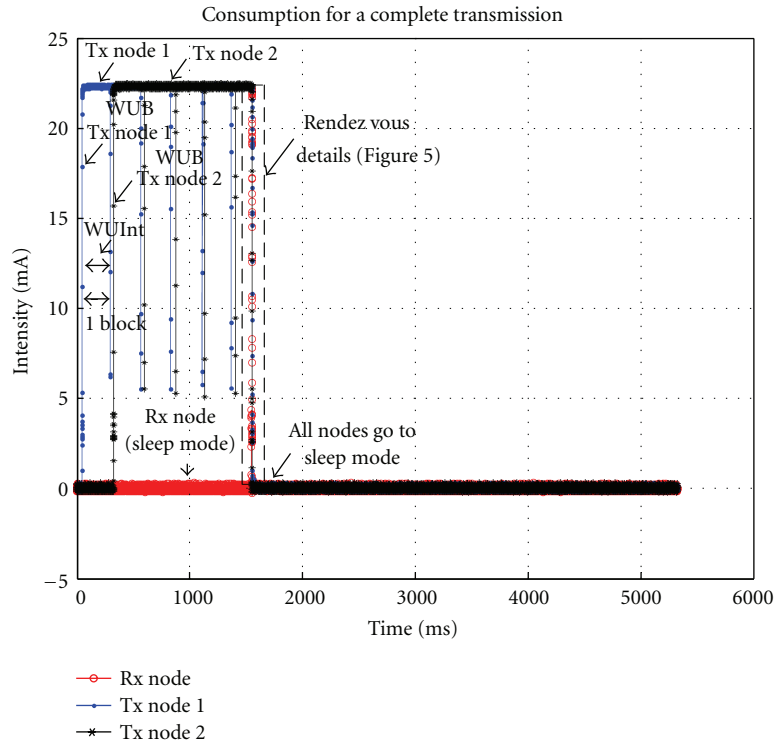


FIGURE 6: Total current consumption: two nodes are trying to transmit the data to the same destination node. In this particular case, after 5 blocks, the transmitter wakes up and sends its WUB, and a rendez vous (detailed in Figure 5) occurs between the two nodes on the single channel. The mean total current consumption during the active period including the cc2420 and MSP430 is 22.5 mA.

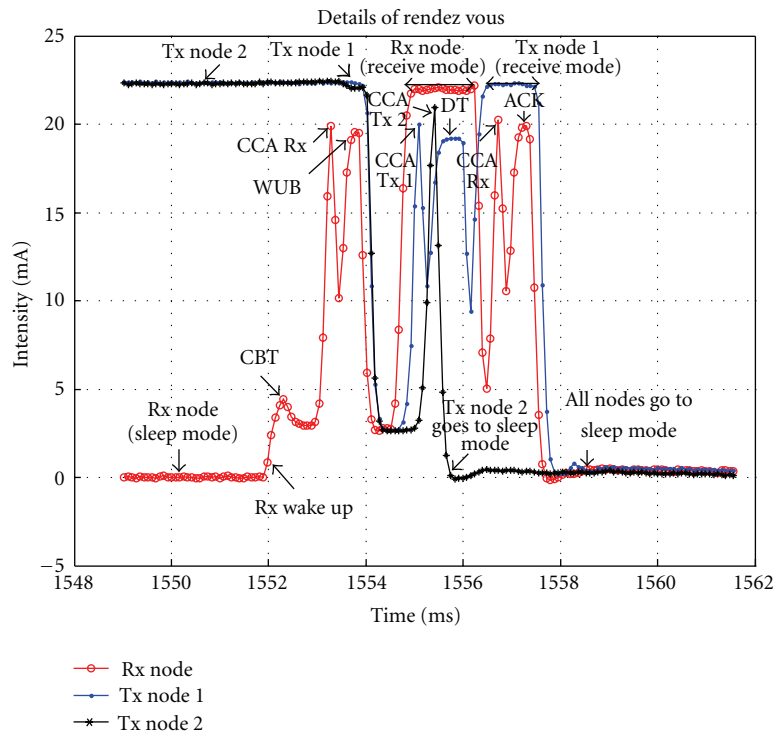


FIGURE 7: Details of rendez vous: three phases, that is, WUB, DT, and ACK are clearly shown, and all the events are described during the rendez vous. The Tx node 1 senses the channel first and hence occupies the channel before the Tx node 2. Therefore, the collision is being avoided.

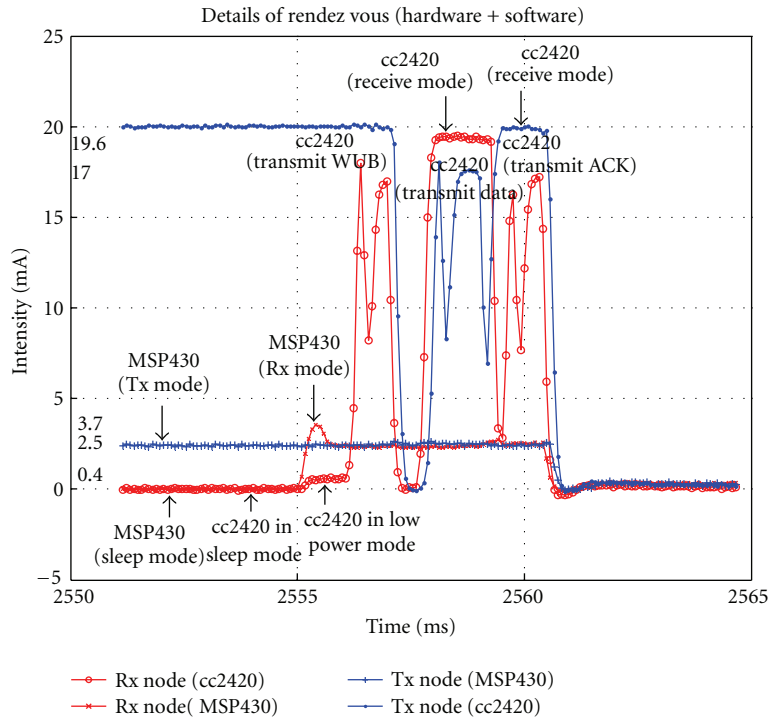


FIGURE 8: Details of software and hardware consumption: the software component, that is, MSP430 consumes,  $0.0\ \mu\text{A}$ ,  $0.01\ \mu\text{A}$ ,  $2.5\ \text{mA}$ ,  $3.7\ \text{mA}$ , in sleep, low power mode (LPM), transmitting, and receiving modes, respectively. Whereas, the hardware component, that is, cc2420 consumes,  $0.0\ \mu\text{A}$ ,  $0.4\ \text{mA}$ ,  $17.6\ \text{mA}$ , and  $19.6\ \text{mA}$ , in sleep, low power mode (LPM), transmitting, and receiving modes, respectively.

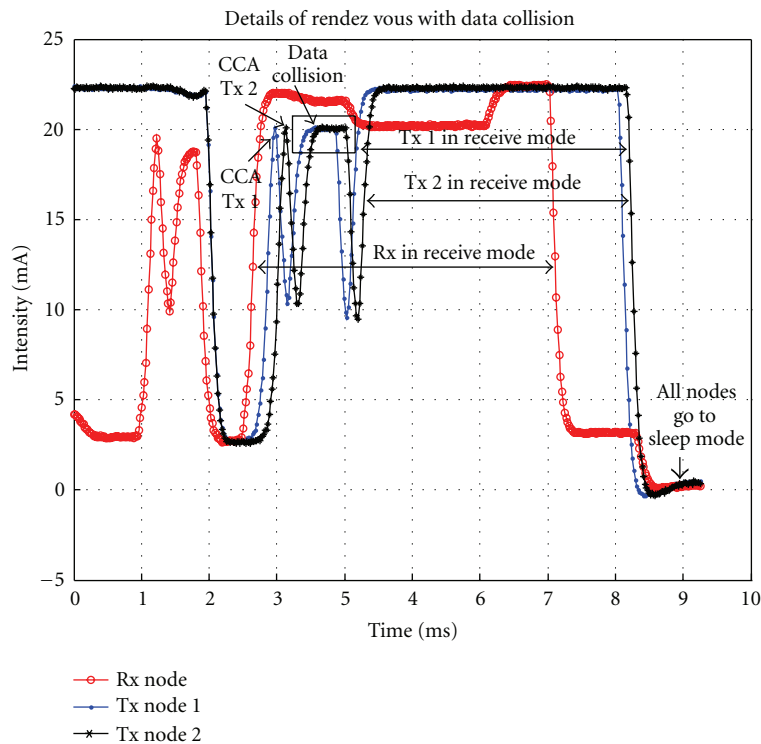


FIGURE 9: Unsuccessful communication because of *data collision*, which is one of the bad executing scenario as mentioned in the energy model. This results in extra power consumption because of the waiting of ACK for the transmit nodes and data packet for the receive node. After about 4 msec. of wait in monitoring state, all nodes go to sleep mode.

This classical energy consumption model is only based on the hardware components, similarly there are purely software energy models such as [15, 16]. These energy models do not give a precise and detailed measure of the energy because there are different scenarios which are associated with the software consumption and the hardware consumption. Therefore, to achieve accurate measure, it is necessary to identify the energy consumption of the scenarios which correspond to the occurrence of the events and more precisely to their timings.

**5.1. Scenario Descriptions and Cost Analyses.** To introduce the scenario-based energy consumption, we consider an example of distributed asynchronous sensor networks (DASN). In DASN, a sensor node can wake-up at a particular instant and communicate with another node without any collision, but the same node can have a collision when it wakes up at another time instant. Consequently, the energy consumption for the same node is completely different. Further, in the software, the energy consumption is different when being in the transmit or receive states, depending upon the software executing traces of different scenarios. The sleeping mode of the software components are linked directly with the frequency of the events and depends on WUInt and WUTime. If there are more events, it does not go to deep sleep mode or otherwise it can be in the deep sleep mode for multiples of WUInt. Similarly, for the hardware consumption, while transmitting the data or control packets, there are two distinct executions: the channel sensing (CCA) and the actual transmission, and their energy consumptions are not the same. Therefore, we have to isolate them and not evaluate them through only one state (i.e., transmit state) as in classical models. We believe that the energy consumed by the software and hardware through scenarios provides details on the energy consumption and moreover the impact of cross-layer on the energy consumption is more clear.

The executing scenarios which typically occur during a communication process are shown in Figure 10. These scenarios are identified in the form of traces and defined as: calculation before transmission (CBT), transmitter wake-up (TWU), wake-up beacon (WUB), wake-up collision (WUC), data transmission (DT), data reception (DR), acknowledgment (ACK), data received with errors (DRE) and data collision (DC). The scenarios which are shown in plain lines are common to every communication while the ones in dotted lines illustrate some bad events that may occur especially in asynchronous rendez vous schemes.

Different energy costs are calculated based on the scenarios that have been identified during the communication and are presented in what follows. The total energy cost is the sum of the costs of the main loop that records the execution traces (as explained in Section 4).

**5.1.1. Basic Cost.** The basic cost of an ideal communication, which is represented in Figure 10 with plain line style, is calculated as

$$C_B = C_{TWU} + 2 \cdot C_{CBT} + C_{DT} + C_{DR} + 2 \cdot C_{WUB} + 2 \cdot C_{ACK}. \quad (18)$$

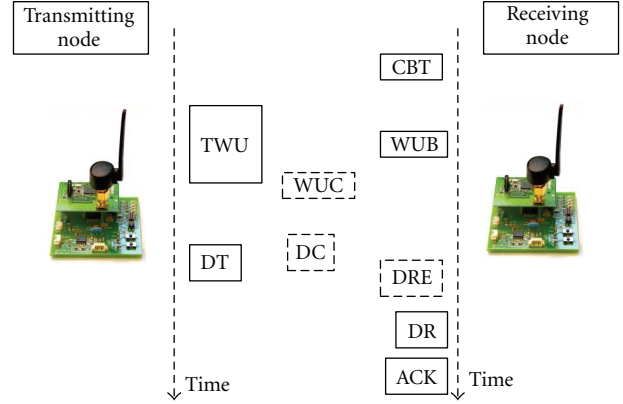


FIGURE 10: Communication phase between transmit and receive nodes: the energy model is based on execution traces of different scenarios which occur during the communication. These scenarios include: calculation before transmission (CBT), transmitter wake-up (TWU), wake-up beacon (WUB), wake-up collision (WUC), data transmission (DT), data reception (DR), data collision (DC), data received with errors (DRE) and acknowledgment of data (ACK).

**5.1.2. Retransmission Cost.** A retransmission happens to be necessary when a received data frame is still erroneous after the error correction step, forcing the transmitter to realize another complete rendez vous with the receiver. The additional cost of this event is

$$C_{RT} = C_{DRE} + C_B, \quad (19)$$

where

$$C_{DRE} = C_{TWU} + 2 \cdot C_{CBT} + 2 \cdot C_{WUB} + C_{DT} + C_{DR}. \quad (20)$$

**5.1.3. Data Collision Cost.** Two or more transmitters want to respond at the same time to the node waking up (i.e., the node that has broadcasted the wake-up beacon), which results in data collision. This bad situation wastes the energy of the transmitter and causes an additional cost of  $C_{DC}$ . It is worth mentioning that  $C_{DC}$  is not straightforward to calculate since the behavior of the nodes during collision is unpredictable. For example, according to our observations, a data collision results in the loss of packet identifiers and, in that case, the receive node keeps receiving the data packet for more than the duration of one packet. Therefore, the  $C_{DC}$  depends on the actual effect of collision that can vary from each other. As an example, one of the behavior of two transmit and one receive nodes during real-time *data collision* is shown in Figure 9 and explained in Section 4.2.

**5.1.4. Wake-Up Collision Cost.** If the two receivers wake-up exactly at the same time and sense the channel being idle, they transmit their wake-up beacon at the same time and then it results in a wake-up collision. This collision causes the waste of energy in the form of  $C_{WUC}$  at the receiver and it has a direct impact on the  $C_{TWU}$  as the transmitting node have to keep waiting for longer time and the term  $E[w]$  will increase.

5.1.5. *Total Cost.* The total cost of the communication from node  $i$  to node  $j$  is then calculated as (each term depending on  $i$  and  $j$ )

$$C_{\text{Tot}} = V_{i,j} \cdot (C_B + N_{\text{RT}} \cdot C_{\text{RT}} + N_{\text{WUC}} \cdot C_{\text{WUC}} + N_{\text{DC}} \cdot C_{\text{DC}}), \quad (21)$$

where  $V_{i,j}$  is the volume of data from node  $i$  to node  $j$ ,  $N_{\text{RT}}$  is the average number of retransmissions per communication,  $N_{\text{WUC}}$  is the average number of wake-up collisions per communication and  $N_{\text{DC}}$  is the average number of *data collisions* per communication. These values can be evaluated by counting the events on real platforms or by the following theoretical formulas (each term depending on  $i$  and  $j$ ):

$$\begin{aligned} N_{\text{RT}} &= \left( \frac{1}{1 - \text{RT}(\text{tol})} \right) - 1, \\ N_{\text{WUC}} &= \left( \frac{1}{1 - Pr_{\text{WUC}}} \right) - 1, \\ N_{\text{DC}} &= \left( \frac{1}{1 - Pr_{\text{DC}}} \right) - 1, \end{aligned} \quad (22)$$

where  $\text{RT}(\text{tol})$  is the retransmission rate,  $Pr_{\text{WUC}}$  and  $Pr_{\text{DC}}$  are probabilities of the wake-up collision and data collision, respectively. These terms can be evaluated through (2), (6) and (9).

5.2. *Hybrid Energy Model.* Our hybrid energy model is presented in Figure 11. The model has different input parameters: wake-up interval (WUInt), wake-up time (WUTime), observation time ( $T_{\text{obs}}$ ), amount of data packets ( $\lambda$ ) and number of nodes in the radio range ( $n_r$ ). The model utilizes the real-time power measurements with an identification of typical scenarios that can occur in asynchronous WSN communication and the analytical model which evaluates the expressions of  $N_{\text{RT}}$ ,  $N_{\text{WUC}}$  and  $N_{\text{DC}}$ . All these parameters together with traffic load ( $\lambda$ ), we compute the global cost and eventually the total energy consumption.

5.3. *Energy Consumption Evaluation.* The energy consumptions have been measured for the different executing scenarios (as presented in Figure 10) and the results are given in Table 2. Moreover the consumptions have been isolated into software and hardware. For the case of software, the energy is measured through profiling of the actual code of different scenarios, whereas for the hardware, it is calculated based on the real-time measurements of the current consumption (from Figures 7 and 8). The current drawn in different states of the radio is interpreted in terms of numerical sample values and the integrated sum over the time results in energy for different scenarios. The total energy consumption for one complete communication (i.e., for transmitting and receiving one packet between two nodes which are within the radio-range) includes all the energy required for scenarios that are shown in Table 2, and also includes the energy of TWU. The energy consumption of the transmitting node in Joules (J) is (0.0431, 0.0052) for (cc2420, MSP430), whereas for the receive node it is

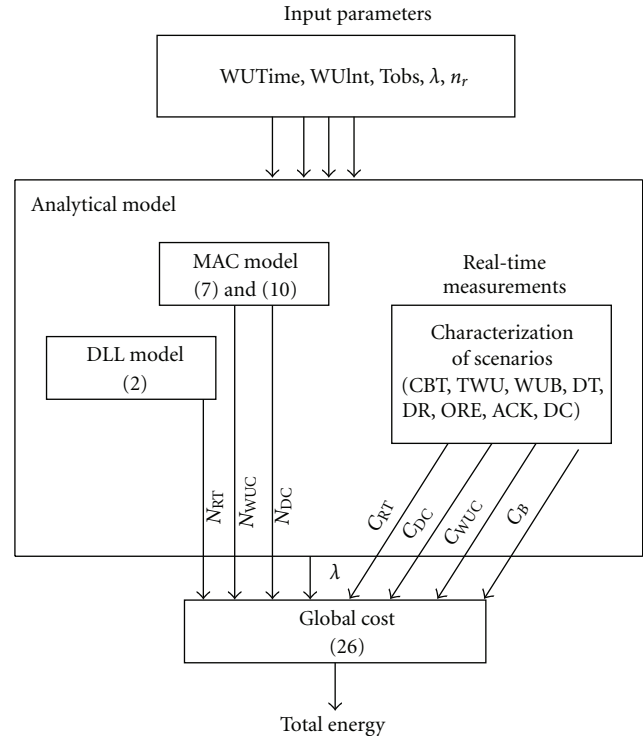


FIGURE 11: Hybrid energy model. The model combines real-time measurements of typical scenarios and analytical models to compute the energy consumption (20).

$(0.1789, 0.0043) \times 10^{-3}$ . The transmitter consumption is  $10^3$  times more in comparison to the receiver because of the penalty of long waiting time  $E[w]$  for the WUB. The energy consumption at the transmitter due to TWU is 0.0483 J, which is nearly equal to the total energy consumed by the transmitting node during one complete communication or more precisely  $10^3$  times more in comparison to the rest of the consumption for the transmitting node. This concludes that the main bottleneck (in the selected MAC protocol of preamble sampling category) lies in expected wait  $E[w]$  for the wake-up beacon. Having said that, this consumption can significantly be reduced by applying adaptive wake-up intervals at each node according to the data traffic.

5.4. *Validation and Performance Evaluation.* In order to validate the presented energy model, a test scenario as shown in Figure 12 is considered to compare the real-time energy consumption of the sensor nodes with the estimated model. Let us consider two nodes  $A$  and  $B$  which are intending to transmit the data to the node  $C$  at a specific rate, the real-time energy consumption of node  $C$  is evaluated for a certain duration of time. This scenario can be applicable to a situation where the node  $C$  acts as a first hop between the node  $A$  or  $B$ , to forward the received data to the destination nodes  $F$ , and  $G$  which can be out of range from  $A$  and  $B$ . The real-time current measurements of the monitored node  $C$  are exported as a file to the external memory of Agilent N6705A DC power analyzer to compute the average energy

TABLE 2: Energy consumption of communication phases (Joules): the software and hardware energy consumptions of the executing scenarios which are identified during communication between sensor nodes. The energy consumed in WUB and ACK is the same because of same control packet size. The energy consumed in TWU is mentioned and explained in the text.

| SW/HW  | CBT                 | WUB/ACK             | DT                  | WUC                 | DC                  |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|
| tx     | $2.6 \cdot 10^{-8}$ | $5.3 \cdot 10^{-8}$ | $1.2 \cdot 10^{-5}$ | $4.1 \cdot 10^{-8}$ | $1.0 \cdot 10^{-5}$ |
| rx     | $5.3 \cdot 10^{-8}$ | $1.2 \cdot 10^{-6}$ | $5.3 \cdot 10^{-8}$ | $1.3 \cdot 10^{-6}$ | $5.3 \cdot 10^{-8}$ |
| Timing | $5.5 \cdot 10^{-7}$ | $5.5 \cdot 10^{-7}$ | $5.5 \cdot 10^{-7}$ | $5.5 \cdot 10^{-7}$ | $5.5 \cdot 10^{-7}$ |
| Link   | $4.8 \cdot 10^{-7}$ | 0                   | 0                   | 0                   | $4.8 \cdot 10^{-7}$ |
| ntwk   | $4.8 \cdot 10^{-7}$ | 0                   | 0                   | 0                   | 0                   |
| Contr. | $3.7 \cdot 10^{-7}$ | $3.7 \cdot 10^{-7}$ | $3.7 \cdot 10^{-7}$ | $3.7 \cdot 10^{-7}$ | $3.7 \cdot 10^{-7}$ |
| cc.Tx  | 0                   | 0                   | $6.8 \cdot 10^{-5}$ | 0                   | $2.9 \cdot 10^{-4}$ |
| cc.Rx  | $7.8 \cdot 10^{-6}$ | $5.0 \cdot 10^{-5}$ | $1.0 \cdot 10^{-4}$ | $3.1 \cdot 10^{-5}$ | $2.3 \cdot 10^{-4}$ |
| Total  | $9.7 \cdot 10^{-6}$ | $5.1 \cdot 10^{-5}$ | $1.7 \cdot 10^{-4}$ | $3.2 \cdot 10^{-5}$ | $5.3 \cdot 10^{-4}$ |

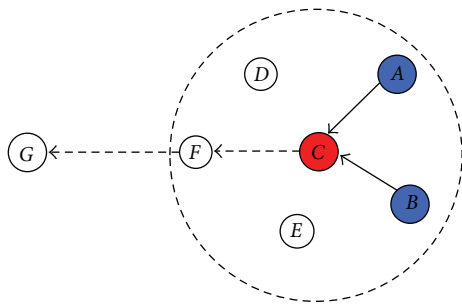


FIGURE 12: Two nodes A and B are intending to transmit the data to node C at a specific rate, and the energy consumption of node C is monitored for a certain duration of time.

consumption. The observation time is one of the constraint in the validation of our model due to limiting memory size of the power analyzer to keep the logged data, therefore the real-time measurements are not possible for several hours and for large-scale network.

Three different test results are presented in Figure 13, for the comparison of the estimation model and the real-time measurements. In order to have reliable measure of the estimation model, the real-time measurements have been repeated five time (i.e., A to E), for every test with the same setting of parameters. During these tests the parameters considered fixed are  $WU_{int} = 100$  ms,  $WU_{Time} = 5$  ms,  $T_{obs} = 300$  s, and  $TWU = 2$  s. The data transmission rate from node A and B to the monitored node C varies in three tests from 1 s, 500 ms and 100 ms for test 1, test 2 and test 3, respectively. For the case of real-time measurements, the average energy per second is computed from 300 s of observation time which contains  $2929 \times 10^3$  samples of the current levels for a sampling resolution of 0.1 ms. The results shown in Figure 13 extend for 300 s of observation time  $T_{obs}$ . For the estimated model, the energy consumption of the scenarios shown in Table 2 are injected in the model to compute the average energy per second and further for 300 s which is compared with the actual measurements as shown in Figure 13.

The energy consumption of estimated model versus real-time measurements (averaged) for test 1, test 2 and test 3

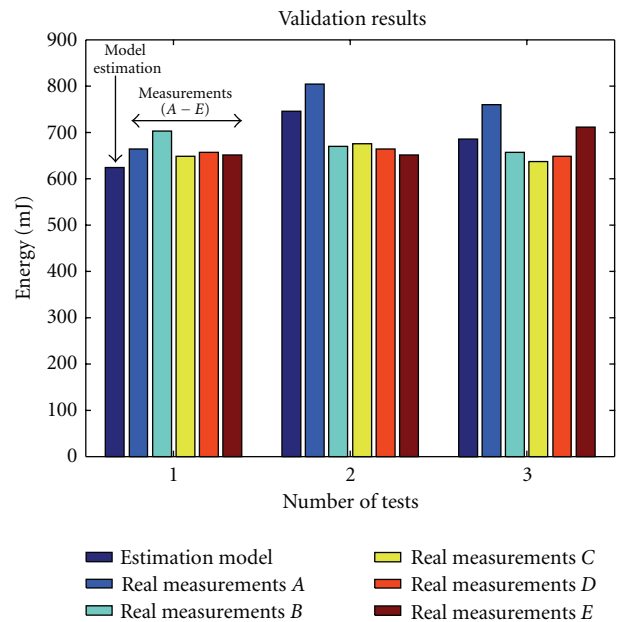


FIGURE 13: Test 1, Test 2 and Test 3 correspond to fixed parameters settings of  $WU_{int} = 100$  ms,  $WU_{Time} = 5$  ms,  $T_{obs} = 300$  s, and  $TWU = 2$  s, whereas the data reception at node C varies from 1 s, 500 ms and 100 ms. Y-Axis represents the energy consumed in mJ for the duration of 300 s.

are shown in Table 3. Results verify that the relative error of our energy model is between 1% and 8%. As the estimated model uses the energy consumption of the scenarios which are evaluated through real-time measurements, therefore we have very accurate energy comparison. The validation test is being performed at the receive node and the energy consumption of *wake-up beacon* and the reception of data packet are considered.

The deviation of the energy model depends upon the parameter *wake-up time* ( $WU_{Time}$ ) of the receive node. In the real-time experiments the receive node (which is a monitored node C of our validation scenario), has a variation in waiting for the data packet between 1.5 and 5 ms.

TABLE 3: Performance evaluation of validated results.

|                           | Test 1 | Test 2 | Test 3 |
|---------------------------|--------|--------|--------|
| Model Estimate (mJ)       | 624.2  | 745.2  | 684.6  |
| Average Measurements (mJ) | 664.2  | 692.2  | 682.2  |
| Standard Deviation        | 19.59  | 56.36  | 46.18  |
| Relative Error            | 6.0%   | 7.5%   | 0.35%  |

This variation is due to the exact wake-up of the transmit nodes, if the transmit node wakes up earlier, the waiting time at the receive node will be less and consequently less energy consumption, but 5 ms is the maximum *wake-up time* (WUTime) that the receive node waits for the data. In our estimate model, the waiting time is taken as an average value between 1.5 to 5 ms.

## 6. Applications of the Energy Model

**6.1. Effects of Collisions and Total Energy.** In order to illustrate the effects of collisions on the overall power consumption of the individual nodes and consequently of the network, a numerical analysis is presented in Table 4, which evaluates the extra waste of energy due to collisions. The amount of collisions in a network is calculated through analytical expressions presented in Sections 3.2.1(a) and 3.2.1(b), whereas the energy wasted due to collisions is calculated through the results of real-time measurements. For the numerical analysis three different WUInts and variable traffic rates of data packets are considered. The probability of WUC and DC have been obtained through (6) and (9), respectively, where the number of nodes within the radio range is taken as  $n_r = 10$ , WUTime = 5 ms is the time that the receive node waits for the data packet from transmit node after sending its WUB and  $T_{\text{obs}} = 1$  hr is the network observation time. To support various WSN applications, the data packets are generated with rate (1/3, 1/2, 1) times of WUInt (3, .3, .1) s, whereas the WUB is transmitted at every WUInt.

It is observed that the probability of collisions sharply increases when the shorter WUInt is selected to support heavy traffic and hence, there is a significant increment in the actual energy consumption. Obviously, the selection of WUInt depends upon the specific application. For typical WSN applications such as critical temperature sensing or environment monitoring, it is evaluated that the probability of WUC and DC have resulted in an increment of 21.5% of the energy consumption in comparison to the energy consumed during noncollision transmission and reception. The traffic and WUInt have a direct impact on the amount of collisions. It is worth mentioning here that, even if there is no collision (i.e., noncontention-based protocol), our energy model and the real-time evaluations are still better in terms of accuracy than the basic power models which consider constant current levels in different states such as Tx, Rx, and so forth.

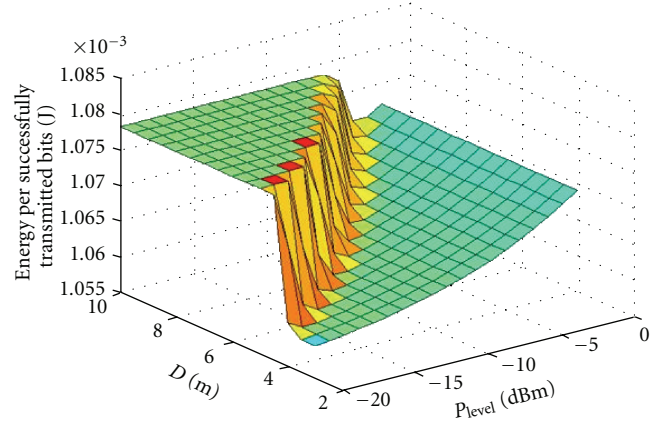


FIGURE 14: Total energy per successfully transmitted useful bit between nodes  $i$  and  $j$  as a function of the distance  $D_{i,j}$  and of the transmission power level  $P_{\text{Level}}$ .

**6.2. Optimization of the Output Transmit Power.** Figure 14 illustrates one of the potential application of our energy model. The total energy per successfully transmitted useful bit between nodes  $i$  and  $j$  is calculated as a function of the distance  $D_{i,j}$  and of the transmission power level  $P_{\text{Level}}$ . Now, for given  $i$  and  $j$ , a particular  $P_{\text{Level}}$  allows to optimize the total power. The optimal point in terms of energy consumption can be found on the curve which gives the best relationship between  $P_{\text{Level}}$  and  $D_{i,j}$ . Similar examples can be obtained, for example, for different error correcting codes we can determine the optimal point on the curve against the power consumption and the energy per useful transmitted bit.

The lifetime prediction of the system can be evaluated by considering a geographical network and by using a routing algorithm to compute the volume of traffic for all the nodes of the network and, thereafter, by using our energy model the lifetime of the network can be predicted.

## 7. Conclusion and Future Work

In this paper a hybrid and scenario-based energy model is presented for accurate energy consumption analysis of the low power WSN. Firstly, an analytical model has been described for the lower layers of protocol stack such as PHY, MAC, DLL, layers, to have a global overview of the energy utilization in the system. The MAC strategy has a particular impact on the performance of the whole system, since the consumption of the radio part is the most important. That is why, the MAC layer has been precisely modeled for preamble sampling category with special focus on receiver initiated cyclic receiver (RICER) protocol. Then the real-time measurements have been presented to understand the details of the energy consumption by isolating the hardware and software consumption. The detailed current consumption of the components in various states are evaluated. Further, the details of the rendez vous including the collision avoidance and *data collision* have been presented.

TABLE 4: The cost of collision energy (C. Energy) results in an extra power consumption of the individual nodes and consequently of the complete network. The traffic is generated for different WUInts and accordingly the probability of collisions are calculated to compute the collision energy.

| WUInt | Packets/hr | WUB/hr | WUC  | DC    | C. Energy |
|-------|------------|--------|------|-------|-----------|
| 3 s   | 400        | 1200   | 1.5% | 2.2%  | 23.8 mJ   |
| .3 s  | 6000       | 12000  | 15%  | 4.6%  | 1107 mJ   |
| .1 s  | 36000      | 36000  | 45%  | 11.2% | 17115 mJ  |

A realistic and accurate energy model for WSN has been proposed. The power evaluation is done by the analysis of the different scenarios that occur during the communication between sensor nodes. These scenarios have been isolated, characterized, and injected into the energy consumption model. Our energy model can help system architects to identify the critical regions for energy savings with a better understanding of the variations in the current levels that occur in different scenarios. Furthermore, it enables the designers to achieve more accurate and energy efficient designs than using classical approaches for energy modeling.

The presented model is validated under different test cases which show that the relative error of the estimated model is less than 8% in comparison with the real-time energy estimate. At the end few examples have been explained which are the applications of our energy model. The model presented is specific to RICER protocol, however the model gives details of the power consumption based on the scenarios which can be identified in other energy efficient protocols. Therefore, in a larger context our model can easily be extended to the other preamble sampling category of MAC protocols. In the future, *adaptive wake-up intervals* in the MAC protocol will be introduced, which will appreciably reduce the *cost of monitoring* ( $E[w]$ ) at the transmit node as well as the cost of idle channel monitoring of the receive node.

## List of Acronyms

|                     |  |
|---------------------|--|
| WUC:                | Wake-up collision                      |
| DC:                 | Data collision                         |
| DT:                 | Data transmission                      |
| DR:                 | Data reception                         |
| TWU:                | Transmitter wakeup                     |
| WUInt:              | Wake-up interval                       |
| SInt:               | Sampling interval                      |
| PTime:              | Preamble time                          |
| Pr <sub>PC</sub> :  | Preamble collision                     |
| WUB:                | Wake-up beacon                         |
| CBT:                | Calculation before transmission        |
| DRE:                | Data received with errors              |
| ACK:                | Acknowledgment                         |
| WUTime:             | Wake-up time                           |
| $T_{obs}$ :         | Observation time                       |
| $N$ :               | Number of data slots                   |
| $n_r$ :             | Number of nodes within the radio range |
| $n$ :               | Number of nodes in the network         |
| $E[w]$ :            | Expected waiting time                  |
| Pr <sub>WUC</sub> : | Probability of wake-up collision       |
| Pr <sub>WC</sub> :  | Probability of data collision.         |

## Acknowledgments

The research described in this paper is funded by European Union ITEA2 Geodes Project and the Conseil Général Côtes-d'Armor, France. The work is conducted in the research Lab IRISA, at University of Rennes 1, France. The authors would like to thank the Lab Engineer Arnaud Carer for realizing the platform measurements and developing the PowWow platform. The authors would also like to thank the reviewers for their valuable comments.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, "Cross-layer design of ad hoc networks for real-time video streaming," *IEEE Wireless Communications*, vol. 12, no. 4, pp. 59–64, 2005.
- [3] L. C. Zhong, J. M. Rabaey, and A. Wolisz, "An integrated data-link energy model for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications*, vol. 7, pp. 3777–3783, Paris, France, 2004.
- [4] M. C. Vuran and I. F. Akyildiz, "Error control in wireless sensor networks: a cross layer analysis," *IEEE Transactions on Networking*, vol. 17, no. 4, pp. 1186–1199, 2009.
- [5] S. Chouhan, R. Bose, and M. Balakrishnan, "Integrated energy analysis of error correcting codes and modulation for energy efficient wireless sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, pp. 5348–5355, 2009.
- [6] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
- [7] <http://powwow.gforge.inria.fr/>.
- [8] E. R. Sanchez, C. Chaudet, and B. Montrucchio, "An energy consumption model of variable preamble sampling MAC protocols for wireless sensor networks," in *Proceedings of the 20th IEEE International Symposium On Personal, Indoor and Mobile Radio Communications*, pp. 2285–2289, Tokyo, Japan, September 2009.
- [9] S. Kellner, M. Pink, D. Meier, and E.-O. Blass, "Towards a realistic energy model for wireless sensor networks," in *Proceedings of the 5th IEEE Annual Conference on Wireless on demand Network Systems and Services*, pp. 97–100, Garmisch-Partenkirchen, Germany, January 2008.
- [10] W. Qin, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SecoN '06)*, vol. 1, pp. 286–295, September 2006.
- [11] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets '07)*, pp. 28–32, New York, NY, USA, June 2007.

- [12] Chipcon, SmartRF CC1000 Single Chip Very Low Power RF Transceiver.
- [13] Chipcon, SmartRF CC2420, 2.4GHz IEEE 802.15.4/ZigBee-ready RF Transceiver.
- [14] E.-Y. A. Lin, J. M. Rabaey, and A. Wolisz, "Power-efficient Rendez-vous schemes for dense wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications*, vol. 7, pp. 3769–3776, Paris, France, 2004.
- [15] O. Landsiedel, K. Wehrle, and S. Gotz, "Accurate prediction of power consumption in sensor networks," in *Proceedings of the IEEE Workshop on Embedded Networked Sensors*, pp. 37–44, Sydney, Australia, 2005.
- [16] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welch, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the ACM Embedded Networked Sensor Systems*, Baltimore, Md, USA, 2004.
- [17] P. J. Lee, "Computation of the bit error rate of coherent M-ary PSK with Gray code bit mapping," *IEEE Transactions on Communications*, vol. 34, no. 5, pp. 488–491, 1986.
- [18] M. K. Simon and M. S. Alouini, *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*, John Wiley & Sons, New York, NY, USA, 1st edition, 2000.
- [19] A. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communications*, vol. 19, no. 5, pp. 751–772, 1971.
- [20] O. Berder and O. Sentieys, "PowWow: energy-efficient HW/SW techniques for Wireless Sensor Networks," in *Proceedings of the Workshop on Ultra-Low Power Sensor Networks*, pp. 229–233, Hannover, Germany, February 2010.
- [21] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, ACM Press, Baltimore, Md, USA, November 2004.
- [22] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 307–320, November 2006.
- [23] A. El-Hoiydi, J. D. Decotignie, C. Enz, and E. Le Roux, "Poster abstract: wiseMac, an ultra low power MAC protocol for the WiseNET wireless sensor networks," in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 302–303, ACM, Los Angeles, Calif, USA, November 2003.
- [24] E. Y. A. Lin, J. M. Rabaey, S. Wiethoelter, and A. Wolisz, "Receiver initiated rendezvous schemes for sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '05)*, vol. 5, pp. 3117–3122, December 2005.
- [25] E. Y. Lin, *A comprehensive study of power-efficient rendezvous schemes for wireless sensor networks*, Ph.D. thesis, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Calif, USA, 2005.
- [26] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki—a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 1st Workshop on Embedded Networked Sensors*, pp. 455–462, Tampa, Fla, USA, November 2004.
- [27] A. Dunkels, O. Schmidt, and T. Voigt, "Using protothreads for sensor node programming," in *Proceedings of the Workshop on Real-World Wireless Sensor Networks*, Stockholm, Sweden, 2005.