Energy Modeling of Multihop Energy Neutral WSN Using Wake-up Receiver

Fayçal Ait Aoudia, Matthieu Gautier and Olivier Berder
University of Rennes 1, IRISA, INRIA, Rennes, France

1 STAGE OF THE RESEARCH

Wireless sensor networks (WSN) constitute a key technology for many applications that require virtual-physical interactions, such as military surveillance, environmental monitoring, home automation and health monitoring. For many long-term applications, a severe bottleneck is the limited amount of energy provided by individual batteries commonly used to power WSN. Energy harvesting forms a promising technology to tackle this problem. Moreover, the most energy consuming task in a wireless node is usually communication. Recent improvements in wake-up receivers (WURx) hardware make them a privileged direction to reduce the consumption of radio on wireless nodes.

Using both technologies can enable WSN to achieve long term sustainability while maintaining quality of service requirements fulfilled. However, this requires careful design of both power managers (PMs) and MAC protocols to efficiently use these two emerging technologies together. The aim of our work is to design efficient PMs and MAC protocols for multihop energy harvesting wireless sensor networks (EH-WSN) in which each node is equipped with a WURx. In this paper, we present the energetic model and two PMs on which we are currently working on.

2 OUTLINE OF OBJECTIVES

Wireless nodes are commonly powered with individual batteries. For long-term applications, the limited amount of energy available in batteries becomes a severe bottleneck and the challenge of maximizing the lifetime of battery-powered WSN has received considerable attention. Two promising technologies to tackle the problem of energy management in WSN are energy harvesting and WURx. Indeed, enabling powering nodes thanks to energy harvested in their environment instead of batteries is a privileged way to achieve long term sustainability. Moreover, recent improvements in the hardware of WURx make them a promising technology to reduce the power consumption of communication in WSN, which is usually the most consuming task.

The goal of our work is to jointly design energy management systems and MAC protocols to efficiently use these emerging technologies in the context of multihop WSN. The rest of this section motivated the use of each of this technology.

2.1 Energy Harvesting Sensor Networks

With advances in energy harvesting techniques, it is now possible to build sensor networks in which each node is entirely powered by energy harvested in its environment (Le et al., 2013b; Kansal et al., 2007; Vigorito et al., 2007). In EH-WSN, each node has at least one energy harvester and one or more energy storage devices. The storage devices are used to buffer energy and thus enable the node to survive periods during which harvested energy is not enough to power it. In those platforms, the challenge is not to maximize the lifetime as in battery-powered WSN, but to ensure long-term sustainability.

Sustainability is obtained when over long periods harvested energy is greater or equal to consumed energy, an operating mode called *Energy Neutral Operation* (ENO) (Kansal et al., 2007). To achieve the best energy efficiency while satisfying the ENO condition, all the harvested energy should be used to increase the nodes performance. The harvested energy then equals the consumed energy over long periods. This operating mode is called *ENO-Max* (Vigorito et al., 2007).

2.2 Wake-up Receivers

Usually, the energy consumed for communication is dominant over all the other nodes activities, such as sensing or computing. To successfully achieve communication, when a node transmits a packet, the destination nodes should be listening the medium to receive data from the sender. This task is particularly challenging in the context of WSN because the duty

Ait Aoudia F., Gautier M. and Berder O..
Energy Modeling of Multihop Energy Neutral WSN Using Wake-up Receiver.

cycling schemes used to improve energy efficiency make this synchronization not trivial. Multiple rendezvous schemes were proposed, and they can be classify into two types (Huang et al., 2013):

Synchronous Schemes which require the nodes to be synchronized such that the wake up time of each node is known a priori. These schemes require synchronization between nodes, which consumes a lot of energy. In the context of EH-WSN, each node may have its own duty cycle, chosen to be the most appropriate to achieve sustainability. As the duty cycle changes frequently, especially in highly dynamical environments, the need of frequent synchronization among nodes often makes these schemes unpractical. Moreover, the nodes wake up even when there is no data to transmit or receive, which results in idle listening.

Asynchronous Schemes which use preamble sampling. Each node chooses its active schedule independently of other nodes and wakes up for short durations to check if there is a transmission on the channel. When a node wants to transmit a packet, it must first transmit a preamble long enough to make sure all the potential receivers detect it and get the data frame. In those schemes, preamble signaling consumes a lot of energy and nodes still need to follow a duty cycle to avoid deafness.

Recent improvements in WURx hardware make them a promising solution for WSN (Magno and Benini, 2014; Marinkovic and Popovici, 2011). The basic idea of WURx is to allow a node to be waking up from deep sleep by another node. When a node wants to transmit a packet, it first transmits a wake up beacon. The receiver WURx hardware detects this beacon and wakes up the node. Moreover, many WURx propose addressing capabilities, which allow a node to insert in the wake up beacon the unique identifier of the receiver. In this way, the WURx only wakes up the node if it reads its own address in the wake up beacon. This feature avoids a node from waking up all its neighbors when it wants to transmit data to only one of them.

3 RESEARCH PROBLEM

Considering multihop WSNs, nodes have two functions. First, as sensors, they must perform sensing operations to generate new data and send them to the sink. Second, they must be relays for other nodes to permit them to forward their packets. We refer to these tasks as the *sensing task* and the *relay task* respectively. As packets aggregation can significantly reduce energy consumption, the packets sending can be uncoupled from these two tasks and seen as a third one.

The available energy for each node must be efficiently allocated between these tasks. Note that, in the context of a node equipped with WURx, performing a relay task must follow the reception of a wake up beacon, *i.e.* an interrupt from the WURx. We call the reception of a wake up beacon a *relay request*. In the context of EH-WSN, energy allocation is guided by the following constraints:

Sustainability: each node must operate under ENO-Max condition

Quality of Service: the network must fulfill some Quality of Service (QoS) requirements that depend on the application. A QoS metric can operate at node scale (e.g. minimum sensing rate) or at network scale (e.g. minimum end-to-end delay).

Two classical examples illustrate how the QoS requirements strongly depend on the application. In an event detection application, a node only sends data when it detects an event (e.g. fire detection). Most of the time, a node does not send anything, but when an event is detected, it must be transmitted to the sink in the shortest delay possible. Therefore, an important QoS metric is the end-to-end delay that should be as small as possible. A good strategy is to send a packet as soon as it is ready. Moreover, energy allocation should favor the relay task in order for the packets to reach the sink in the shortest delays.

In monitoring applications, QoS requirements might be a minimum sensing rate for each node. If the end-to-end delay is not an important constraint, for example because the data are not going to be processed immediately anyway, a good strategy for the energy allocation policy is to favor the sensing task in order to maximize the sensing rate.

These two examples illustrate how the design of an efficient PM depends on the application and the QoS requirements. We aim to design efficient power management schemes and MAC protocols to efficiently tackle the energy allocation problem. Our work targets EH-WSN in which each node is equipped with a WURx device capable of addressing functionalities.

4 STATE OF THE ART

Research in EH-WSN is a hot area and much attention is drawn to several topics such as harvester technologies, storage devices, power management, MAC and routing protocols and others. Similarly, many efforts are devoted to WURx hardware. Yet, combining energy harvesting technologies and WURx is just emerging and, to the best of our knowledge, there has been only little research in that direction.

Magno et al. proposed a power device (Magno et al., 2012) featuring multi-source harvesting capabilities, multiple energy storage devices, fuel cells and radio trigger capabilities. This device is also equipped with a TI MSP430 microcontroller. The idea is to let the device executing the energy management policy programs. It is also responsible for waking up the node when a wake up signal arises and offers energy monitoring features to the node. Therefore, in contrast to our work, the aim of the authors is not to propose energy management policies, but a device to execute them.

In (Magno et al., 2013), the authors introduced a power device to be connected to an existing node. The device they proposed is equipped with two harvesters, one battery and one WURx. One of the harvester is use to supply the battery and power the node, while the other is a piezoelectric MEMS harvester used to supply the WURx in order to achieve “zero power” WURx.

Le et al. proposed in (Le et al., 2013a) an adaptation of the TICER MAC protocol for EH-WSN node equipped with WURx. Each node uses a PM to dynamically adapt the duty cycle to achieve ENO-Max. The authors show that by using a WURx, idle listening is drastically reduced and thus more energy is available for increasing the QoS. The main limitations of this work are that only one-hop network is considered and the lack of collisions avoidance scheme. Also, the addressing capabilities of the WURx are not used.

5 METHODOLOGY

In this section, we present the energy model we are currently working on and two PMs exploiting this model.

5.1 Energetic Model

We suppose that the time is divided into equal length cycles of duration T. Energy conservation over any cycle k leads to the following equation:

$$P_H(k)T - P_L T - \frac{1}{\eta}E_C(k) = \Delta E_S(k) \qquad (1)$$

where, $P_H(k)$ is the average harvested power, P_L is the leakage power and is assumed to be constant. $E_C(k)$ is the energy consumed by the node, and $\Delta E_S(k)$ is the variation of stored energy. The factor $\eta \in [0,1]$ takes into account a reduced energy efficiency due to power conversion.

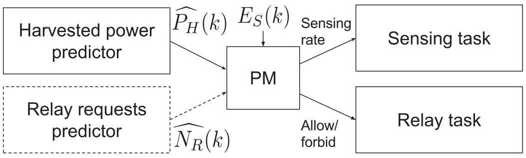

Figure 1: Global software architecture of our system. The relay requests predictor is only needed for the Predictive Allocator PM (see section 5.2.2), and is therefore dashed. The PM sets the sensing task execution rate and allow/forbid the execution of the relay task.

ENO-Max condition is achieved when consumed energy equals harvested energy, *i.e.* $\Delta E_S(k) = 0$. Therefore, ensuring ENO-Max operating state means maintaining the stored energy at a constant value denoted E_S^{ref}.

In practice, the PM is executed at the beginning of every cycle. We denote the residual energy $E_{res}(k) = E_S(k) - E_S^{ref}$, where $E_S(k)$ is the amount of stored energy. Moreover, we have to predict the production of the energy source over the cycle. We denote the *predicted* average harvested power over the cycle k $\widehat{P_H}(k)$. We can now rewrite (1) as follows:

$$\widehat{P_H}(k)T + E_{res}(k) - P_L T - \frac{1}{\eta}E_C(k) = 0 \qquad (2)$$

The PM has two levers for controlling the node power consumption: setting the sensing task rate and allowing/forbidding the execution of the relay task when a relay request arises. If the relay task is not allowed to be executed, then the relay request is simply ignored. Figure 1 shows the global software architecture of our system.

Let us now model the node activity. Two separate cases are defined: the packets are forwarded as soon as they are ready (immediate transmission), and the packet are aggregated and sent later.

5.1.1 Immediate Transmission

In the first case, we assume that the packets are forwarded as soon as they are ready. Then we have:

$$\begin{aligned} E_C(k) = N_S(k)(E_S + E_T) + N_R(k)(E_R + E_T) \\ + P_S T_{Sleep}(k) \end{aligned} \qquad (3)$$

where $N_S(k)$ and $N_R(k)$ are the number of sensing tasks and relay tasks respectively executed over the cycle. E_S, E_R and E_T refer to constant energy costs for performing a sensing task, a relay task and a packet transmission respectively. We thus can rewrite (2) as follows:

$$\widehat{P_H}(k)T + E_{res}(k) - P_L T - \frac{1}{\eta}\Big(N_S(k)(E_S + E_T) + N_R(k)(E_R + E_T) + P_S T_{Sleep}(k)\Big) = 0 \tag{4}$$

where P_S is the node power consumption when it is in sleep state and $T_{Sleep}(k)$ the time spent in sleep state. As we assume low duty cycles, we approximate $T_{Sleep}(k)$ by T for all k.

5.1.2 Aggregation

We assume in this section that we use packets aggregation. We propose to postpone the transmission of all the packets generated or received during the cycle $k-1$ to the next cycle k. To reduce collisions, we send them at a random time. We denote $E_{TA}(k)$ given in (5) the cost of sending all the packets which have been ready to be sent during the cycle $k-1$.

$$E_{TA}(k) = \Big(N_S(k-1) + N_R(k-1)\Big)E_T \tag{5}$$

We thus have:

$$E_C(k) = N_S(k)E_S + N_R(k)E_R + E_{TA}(k) + P_S T \tag{6}$$

and we can rewrite (2) as follows:

$$\widehat{P_H}(k)T + E_{res}(k) - P_L T - \frac{1}{\eta}\Big(N_S(k)E_S + N_R(k)E_R + E_{TA}(k) + P_S T\Big) = 0 \tag{7}$$

5.2 Possible Designs of Power Managers

We propose in this section two designs of PMs on which we are currently working on. These PM share a common global design and are divided into two components. The first component is called *Cyclic PM* (CPM) and is executed at the beginning of every cycle. The second component, called *WURx interrupt handler* (WIH), is executed after each WURx interrupt. The aim of the latter component is to decide whether a relay task should be performed or not in response to a relay request. This design is illustrated in Figure 2.

We assume that a QoS constraint is the maximum duration, denoted T_S^{max}, separating two sensing tasks. In the rest of this section, only the equations for the packets aggregation case are shown, but the equations describing the immediate transmission case can be obtained similarly.

5.2.1 Reactive Allocator PM

The idea of this PM is to always try to maximize the sensing rate. When a relay request arises, the WIH allows a relay task to be executed only if it estimates that it will not increase the sensing period over T_S^{max}.

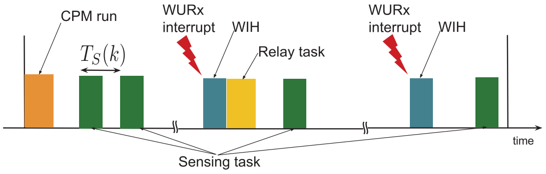

Figure 2: Illustration of the behaviour of the PMs. The CPM is executed at the beginning of the cycle. Then, after every WURx interrupt, the WIH is executed and decides whether the relay task is executed or not. On this illustration, the relay task is executed after the first WURx interrupt, but not after the second. Finally, the sensing tasks are executed periodically according to the rate set by the PM.

At the beginning of each cycle, the CPM computes the greatest number of sensing tasks achievable over the cycle by allocating all the available energy to this task, *i.e.* $N_R(k) = 0$. From (7) we have:

$$\widetilde{N_S^0}(k) = \frac{1}{E_S}\Big(\eta(\widehat{P_H}(k)T + E_{res}(k) - P_L T) - P_S T - E_{TA}(k)\Big) \tag{8}$$

and the initial sensing rate is thus $T_S^0(k) = \frac{T}{\widetilde{N_S^0}(k)}$.

After the i^{th} relay request which arise at the time t_i, the WIH estimates the greatest number of sensing tasks achievable until the end of the cycle *if a relay task is performed*. If we denote $N_S^i(k)$ the number of sensing tasks performed since the beginning of the cycle, then:

$$\widetilde{N_S^i}(k) = \frac{1}{E_S}\Big(\eta(\widehat{P_H}(k)T + E_{res}(k) - P_L T) - P_S T - E_{TA}(k) - iE_R - N_S^i(k)E_S\Big) \tag{9}$$

The WIH can then compute the expected sensing rate $\widetilde{T_S^i}(k) = \frac{T - t_i}{\widetilde{N_S^i}(k)}$. If $\widetilde{T_S^i}(k) \leq T_S^{max}$, then a relay task is performed and the new sensing rate is set to $\widetilde{T_S^i}(k)$. Otherwise, the relay request is ignored and the sensing rate stays unchanged.

This PM favors the sensing task over the relay task. It aims to always maximize the sensing rate by not pre-allocating energy for the relay task. If a relay request arise, then the node will perform a relay task only if that does not deteriorate the sensing rate below the fixed limit.

5.2.2 Predictive Allocator PM

This PM allocates energy budgets for each task (sensing and relay) by estimating the needs of the relay task based on past observations. Unlike the previous

PM, the CPM sets the sensing rate at a fixed value for the whole cycle. Moreover, the CPM also sets a maximum number of relay requests that can be satisfied over the cycle, denoted $N_R^{max}(k)$. Then, the job of the WIH is simply to allow execution of relay tasks while less than $N_R^{max}(k)$ of them were performed since the beginning of the cycle.

At the beginning of each cycle, the CPM estimates the number of relay requests that will arise based on past observations. This predicted value is denoted $\widehat{N_R}(k)$. Then, it computes the corresponding number of sensing tasks that can be performed:

$$\widetilde{N_S}(k) = \frac{1}{E_S}\Big(\eta(\widehat{P_H}(k)T + E_{res}(k) - P_L T) - P_S T - E_{TA}(k) - \widehat{N_R}(k)E_R\Big) \tag{10}$$

To ensure sensing QoS constraint, the PM chooses $N_S(k) = max\left\{\widetilde{N_S}(k), \frac{T}{T_S^{max}}\right\}$. Then, the sensing rate for the cycle k is set to $T_S(k) = \frac{T}{N_S(k)}$ and $N_R^{max}(k)$ is set to:

$$N_R^{max}(k) = \frac{1}{E_R}\Big(\eta(\widehat{P_H}(k)T + E_{res}(k) - P_L T) - P_S T - E_{TA}(k) - N_S(k)E_S\Big) \tag{11}$$

Unlike the previous PM, this one allocates energy for the relay task at the beginning of every cycle. The amount of energy allocated for the relay task is based on prediction of the future amount of relay requests that will arise. If the number of relay requests is overestimated, then too much energy will be allocate for relaying at the expense of the sensing task. Therefore, the sensing rate will be suboptimal. Conversely, if the number of relay requests is underestimated, then some of them will be ignored. In that case, the sensing rate is larger than what it would be if the number of relay requests was accurately predicted. But ignoring relay requests can decrease the overall network performance.

5.2.3 Network Scale Considerations

For each node u, we assume that the routing layer provides a set of admissible forwarders $F(u)$. Then, when a node wants to forward a packet, it chooses one of the forwarders and sends a wake-up beacon, *i.e.* a relay request. If the forwarder refuses the request, then it tries another one, until one of the forwarders accepts it. We think that this will favor routing path containing nodes with high energy harvesting rates.

Another important consideration is how to avoid nodes which are favored by the routing algorithm to be flooded by relay requests and thus to always run with low sensing rates. One possibility is to allow nodes to refuse every relay request after running a relay task for a period of time. The duration of this period will depend on the energetic status of the node. We believe that this scheme will favor the use of nodes with good energy harvesting rate because they will have shorter "deaf" periods.

Another possibility is to enable the nodes to exchange their energetic status with their one-hop neighbors. Then, when a node needs to forward a packet, it can choose randomly among the admissible forwarders using a statistical distribution that favor nodes with a good energetic status. If a node u is aware of some energetic status metric $S(v)$ (e.g. $N_R^{max}(k)$) for all $v \in F(u)$, then it can choose randomly among its admissible forwarders with the probability of choosing the node v being $\frac{S(v)}{\sum_{i \in F(u)} S(i)}$.

Moreover, we are currently working on the design of a MAC protocol to efficiently implement the relay request mechanism.

6 EXPECTED OUTCOME

In the current state of our work, we did not evaluate our proposal. We plan to first experiment using the OMNeT++ simulation framework (OMNeT++, 2014), and then on real hardware using the PowWow platform (pow, 2014) and the WURx from (Magno and Benini, 2014).

REFERENCES

(2014). PowWow: Power Optimized Hardware and Software FrameWork for Wireless Motes. http://powwow.gforge.inria.fr/. [Online; accessed December-2014].

Huang, P., Xiao, L., Soltani, S., Mutka, M., and Xi, N. (2013). The evolution of mac protocols in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 15(1):101–120.

Kansal, A., Hsu, J., Zahed, S., and Srivastava, M. B. (2007). Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*.

Le, T. N., Magno, M., Pegatoquet, A., Berder, O., Sentieys, O., and Popovici, E. (2013a). Ultra low power asynchronous mac protocol using wake-up radio for energy neutral wsn. In *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems*, ENSSys '13, pages 10:1–10:6, New York, NY, USA. ACM.

Le, T. N., Pegatoquet, A., Sentieys, O., Berder, O., and Belleudy, C. (2013b). Duty-cycle power manager for thermal-powered wireless sensor networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pages 1645–1649.

Magno, M. and Benini, L. (2014). An ultra low power high sensitivity wake-up radio receiver with addressing capability.

Magno, M., Jackson, N., Mathewson, A., Benini, L., and Popovici, E. (2013). Combination of hybrid energy harvesters with mems piezoelectric and nano-watt radio wake up to extend lifetime of system for wireless sensor nodes. In *Architecture of Computing Systems (ARCS), Proceedings of 2013 26th International Conference on*, pages 1–6.

Magno, M., Marinkovic, S., Brunelli, D., Popovici, E., O'Flynn, B., and Benini, L. (2012). Smart power unit with ultra low power radio trigger capabilities for wireless sensor networks. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '12, pages 75–80, San Jose, CA, USA. EDA Consortium.

Marinkovic, S. and Popovici, E. (2011). Nano-power wireless wake-up receiver with serial peripheral interface. *Selected Areas in Communications, IEEE Journal on*, 29(8):1641–1647.

OMNeT++ (2014). OMNeT++ Network Simulation Framework. http://www.omnetpp.org/. [Online; accessed November-2014].

Vigorito, C., Ganesan, D., and Barto, A. (2007). Adaptive control of duty cycling in energy-harvesting wireless sensor networks. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON*, pages 21–30.