

# A Two-Prong Approach to Energy-Efficient WSNs: Wake-Up Receivers plus Dedicated, Model-Based Sensing

☆

Usman Raza<sup>a,\*</sup>, Alessandro Bogliolo<sup>b</sup>, Valerio Freschi<sup>b</sup>, Emanuele  
Lattanzi<sup>b</sup>, Amy L. Murphy<sup>c</sup>

<sup>a</sup>*Toshiba Research Europe Limited, Bristol, BS1 4ND, United Kingdom*

<sup>b</sup>*University of Urbino, Piazza della Repubblica 13, Urbino, Italy*

<sup>c</sup>*Bruno Kessler Foundation, Via Sommarive 18, Trento, Italy*

---

## Abstract

Energy neutral operation of WSNs can be achieved by exploiting the idleness of the workload to bring the average power consumption of each node below the harvesting power available. This paper proposes a combination of state-of-the-art low-power design techniques to minimize the local and global impact of the two main activities of each node: sampling and communication. Dynamic power management is adopted to exploit low-power modes during idle periods, while asynchronous wake-up and prediction-based data collection are used to opportunistically activate hardware components and network nodes only when they are strictly required. Furthermore, the concept of “model-based sensing” is introduced to push prediction-based data collection techniques as close as possible to the sensing elements. The results achieved on representative real-world WSN case studies show that the combined benefits of the design techniques adopted is more than linear, providing an overall power reduction of more than 3 orders of magnitude.

**Keywords:** Wireless Sensor Network, Wake-up receiver, Energy harvesting, Model-based sensing, Energetic sustainability

---

---

☆ A preliminary version of this paper appeared in the proceedings of 9<sup>th</sup> IEEE International Symposium on Industrial Embedded Systems (SIES 2014) [1]

\*Corresponding author

*Email addresses:* [usman.raza@toshiba-trel.com](mailto:usman.raza@toshiba-trel.com) (Usman Raza), [alessandro.bogliolo@uniurb.it](mailto:alessandro.bogliolo@uniurb.it) (Alessandro Bogliolo), [valerio.freschi@uniurb.it](mailto:valerio.freschi@uniurb.it) (Valerio Freschi), [emanuele.lattanzi@uniurb.it](mailto:emanuele.lattanzi@uniurb.it) (Emanuele Lattanzi), [murphy@fbk.eu](mailto:murphy@fbk.eu) (Amy L. Murphy)

## 1. Introduction

Advances in energy harvesting (EH) have begun to shift the decade-old research goal of the WSN community from energy efficiency to energy autonomy. Nevertheless, building a self-sustaining EH-WSN remains a challenging task, especially in indoor environments where the harvested energy can be several orders of magnitude less than the consumed energy. For example, a typical, optimized application can spend milliwatts in sensing and communication, while the typical harvesting range for an indoor harvester remains on the order of microwatts [2, 3]. Therefore, work to reduce power consumption is still required to achieve energy autonomy.

Dynamic power management (DPM) represents the cornerstone to any power reduction approach. By targeting workload idleness and putting a node's micro-controller (MCU) into a low-power inactive mode, DPM saves energy. In such a state, consumption drops from hundreds of milliwatts, seen for high consuming activities such as radio transmission, to a fraction of a microwatt. Clearly as nodes spend more time in this low-power state, system lifetime improves. Nevertheless, the activities of a typical node, including sampling, channel listening and receiving, and transmitting, require the MCU to switch to high power modes, with each transition additionally incurring a cost. Our work proposes a unique combination of hardware and software techniques to reduce the frequency and duration of various high-consumption tasks, not only increasing the time a node can spend in the low-power state, but decreasing the number of transitions out of this state. This results in huge energy savings without compromising the application requirements.

We propose a novel technique for WSN applications with periodic data collection, such as those where sensor data forms the input of a control loop, thus requiring timely communication of the sensed values at a central controller. Our proposed architecture employs (1) a wake-up receiver alongside (2) a dedicated sensing peripheral that incorporates (3) a data reduction software module. The first and third techniques aim to decrease the cost of data collection by focusing on the use of the radio, as it represents one of the most power hungry components on the node, while the second technique achieves additional savings by reducing the cost of frequent sensor sampling. The combination of these techniques results in very long idle periods with rare occurrences of power hungry tasks such as radio transmissions and receptions. Low-power modes, provided by multiple underlying hardware components, are exploited by the dynamic power manager during the idle periods to conserve energy.

To illustrate the potential of these techniques, we briefly address each, beginning with the communication. Traditional system architectures employ sophisticated duty cycling medium access control (MAC) techniques to achieve significant reductions in consumption by putting the radio to sleep for extended periods. Nevertheless, nodes still waste energy in two primary ways. First, they must periodically wake up and listen to the channel in case a node is attempting to transmit to it. If there is nothing to receive, this energy spent listening is wasted, leading to idle listening. On the other side, a sender with data to communicate must transmit until the receiver wakes up, often leading to long transmission times among unsynchronized nodes. Wake-up receivers are a novel hardware approach to eliminate these two main sources of overhead. Specifically, they provide an ultra low power receiver that is always on and listening to the channel, either the same channel used for communication or a dedicated, out-of-band channel. When a packet is to be transmitted, a preamble is generated by the transmitter to trigger the wake-up of the data radio on the receiving node. This eliminates idle listening by turning on the main radio module only when there is a packet to be received. Further, it reduces the transmission time by ensuring that the receiver is ready to receive immediately after transmission of the preamble, thus avoiding the repeated transmissions typical of duty cycling protocols.

While exploiting a wake-up receiver leads to significant lifetime gains in many scenarios, applications that collect data at high frequency still incur significant costs to transmit the raw data. To ameliorate these costs, several techniques have been proposed to reduce the amount of data sent without compromising the application requirements. In the technique we consider, each node calculates a model for its data and communicates this model to the sink. The sink then uses these models to predict the data samples at each node. As long as the real samples closely match the model, no data is communicated, however as soon as the real data deviates significantly from the data estimated by the model, a node generates a new model and transmits it to the sink. Such approaches have the potential to eliminate 90 to 99% of the transmissions [4], depending on the type of data being sampled and the sophistication of the model.

Interestingly, the combination of a data reduction technique and the wake-up receiver reduces the radio cost to a point where periodic sensor sampling, normally considered a low-consumption task, actually accounts for a significant portion of the energy consumption. Therefore, to make the sampling cheaper, we introduce a novel hardware-software based technique called Model-based Sensing (MBS). MBS delegates the tasks of periodi-

cally sampling the sensors and the running data reduction algorithm to a dedicated ultra-low power hardware peripheral. This peripheral operates without involving the power hungry primary MCU as long as the data reduction technique is successful in suppressing the sensed data. Only rarely, when there is a need to transmit data, is the main MCU turned on to transmit the data.

By combining these techniques, we increase the length of the idle periods of the MCU and the radio transceiver, which each offer multiple low-power modes to save energy. Additionally, ancillary components such as flash memories and real-time clocks, can be opportunistically turned on and off to save power. In general, the low-power modes and component states must be carefully managed to meet the workload and minimize consumption. This paper shows the potential savings of this combination of techniques in concrete, real-world case studies representative of many periodic data collection applications. The resulting system consumes only a few microwatts, resulting in a system lifetime improvement of *three orders of magnitude*, and reaching the point where indoor energy harvesters can sustain node operation.

In summary, the contributions of this paper include:

- a system architecture that uniquely combines three software and hardware techniques to drastically reduce node workload and offload the frequent and expensive activities from the main MCU. The MBS module is presented for the first time in this paper. Our combination results in ultra-lightweight applications with long idle periods between consecutive activities.
- a case-study based evaluation of the savings achievable when the low-power modes of the radio, MCU and various peripherals exploit the increased idleness achieved by our architecture. To fully examine the contributions of each technique, we evaluate them both individually as well as in various combinations.

The rest of the paper is organized as follows: Section 2 provides an overview of related work on data reduction techniques and ultra low-power WSNs; Section 3 describes the overall system architecture; Section 4 introduces the design principles of model-based sensing; Section 5 and Section 6 describe the experimental setup and present the results respectively; Section 7 concludes the work.

## 2. Background and Related Work

This section offers background on the techniques used throughout this paper to effectively exploit DPM: prediction-based data collection, wake-up receivers and ultra low-power sensor nodes.

### 2.1. *Prediction-based data collection*

As a first step toward reducing consumption, we turn to techniques that reduce the data that must be transmitted. Many such techniques exist such as data compression, in-network data processing and data aggregation [5], but we focus on prediction-based data collection due to its simplicity and demonstrated effectiveness on scenarios with periodic data collection, e.g., where a WSN serves as a component in a control loop.

In prediction-based data collection, the original application-required sampling period is maintained, but the total amount of data transmitted is reduced [5] by generating a model for the sensed data. This model is used at the sink to approximate the sampled data points. With each new sample, the node verifies that it falls within allowable error tolerances. If so, no action is taken, but if not, a new model is generated and transmitted to the sink. If the model closely approximates the data trend, the network communication is significantly reduced, up to 99% in some cases [6, 7, 8]. Various types of models have been studied. Probabilistic models [9, 10] approximate data with a user-specified confidence, but special data characteristics must be encoded by domain experts. Alternate techniques employ linear regression [11], autoregressive models [12] and Kalman filters [13], but with sizeable memory and computational requirements, making them difficult to implement on resource-limited nodes. A simpler, linear approach [6], detailed in Section 3.1, was recently proposed by some of the authors of this paper, and is adopted for the case study here.

### 2.2. *Wake-up Receiver*

Wake-up receivers are a viable solution to enable low-power, asynchronous communication, essentially by triggering the activation of the primary radio [14]. Several wake-up solutions have recently been developed [15, 16, 17, 18, 19, 20], each optimizing different parameters such as working power, sensitivity, distance range, latency, and operating frequency. These solutions use the radio channel to convey triggering signals, but alternative proposals use out-of-band signals such as ultrasound. Notably, this paper evaluates two wake-up modules for the VirtualSense platform i) an ultrasonic wake-up receiver recently developed by some of the authors of this paper [21] and ii) a

radio wake-up receiver presented in [19]. Both outperforms state-of-the-art radio wake-up receivers with a sub- $\mu\text{A}$  quiescent current consumption.

Recent researches have shown the potential for asynchronous wake-up schemes in some application scenarios [22, 1]. Specifically, studies [23, 24] have shown that, in comparison to duty cycling protocols, wake-up receivers offer longer system lifetimes, lower latency, and better reliability in multiple application domains. These studies note that wake-up receivers are particularly effective for applications with ultra-low traffic, as they avoid the idle listening incurred at routing nodes to periodically check for incoming packets. This paper builds on this observation, specifically showing how traditional applications with moderate data rates can be converted into ultra-low traffic applications with prediction-based data collection, thus enhancing the effectiveness of wake-up receivers and DPM.

### *2.3. Ultra-low Power Platforms*

Careful design of embedded wireless hardware platforms is key to energy efficient WSNs. Over the last 15 years, many efforts have been devoted to devise novel ultra-low power MCUs and radio transceivers. First generation WSN platforms included Rene [25], Mica [26] and Telos [27]. Afterwards, the design of embedded architectures for sensor nodes has mainly relied on 16-bit MCUs and IEEE 802.15.4 compliant transceivers. Recent trends show a shift towards 32-bit platforms that meet strict WSN energy requirements [28]. All of these platforms come with the ability to opportunistically turn off hardware components in periods of idleness. As the runtime power consumption depends on various dynamic power management features available within these nodes, this paper focuses on one single node named VirtualSense, a platform designed by some of the authors of this paper. The next section focuses on the various elements VirtualSense offers for dynamic power management.

## **3. System Architecture**

The primary contributions of this paper are first, a novel combination of technologies from hardware to software to achieve fully energy autonomous systems and second, the concrete, case-study based evaluation of multiple configurations of these technologies. Figure 1 offers a very high level overview of the configurable components we consider, dividing them between software and hardware, then further dividing the hardware among those belonging to the VirtualSense [29] platform (the MCU and the data transceiver), the wake-up receiver, the model based sensing (MBS) element

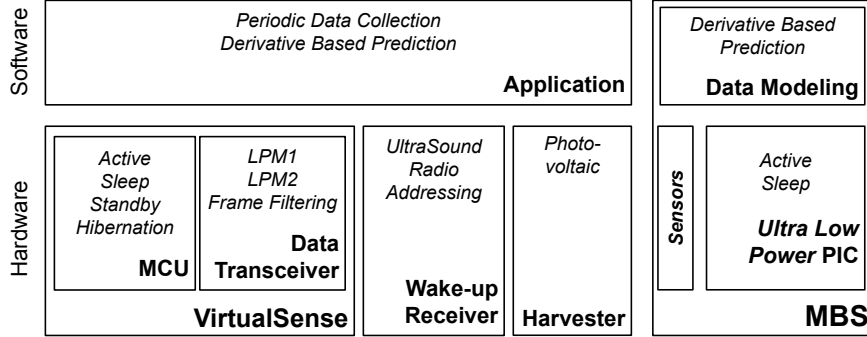


Figure 1: High level system architecture showing the primary configurable components and the configuration options available for each of them.

(MBS), and the energy harvester. This section offers a high level description of most of these components, emphasizing their configuration options. As MBS itself is a novel contribution of this paper, we discuss it separately in Section 4.

### 3.1. Derivative-Based Prediction (DBP)

To reduce the amount of data transmitted by each node, we developed an easily implementable data prediction technique that captures the data trends. With DBP, first described in [6], we adopted a linear model computed from  $m$  data samples, the first and last  $l$  points we refer to as edge points. The linear model is calculated as the slope of the line connecting the average of the first  $l$  edge points with the average of the last  $l$  edge points. This computation resembles the calculation of the derivative, hence the name *Derivative-Based Prediction*.

On initialization,  $m$  points are collected, then the first model is generated and sent to the sink. Subsequently, each sensor sample is checked against the value the model predicts. If the reading is within a given tolerance, no action is taken as the sink will also use the model to approximate the sensor sample. However, if the application tolerances are exceeded, a new model is generated and transmitted to the sink.

To offer a brief example, consider an outdoor light sensor. At sunrise the linear, DBP model will be an upward sloping line. At some point, however, the light levels will cease to increase and the upward sloping model will

be replaced with a flatter model, corresponding to the daytime light levels. While this explanation is over-simplified, it offers the intuition of DBP.

It is worth mentioning that we have applied DBP to several real data sets ranging from soil temperature to indoor temperature and light values [4]. In all cases, a reasonably tuned DBP produces data reduction rates above 89% and in most cases above 98%. These savings are sufficient to exploit the combination of technologies explored in this paper.

DBP, or any data reduction approach, can either be run on the main MCU or on pushed closer to the hardware and run as part of MBS sensing peripheral. We discuss the latter case in Section 4 and highlight its benefits in Section 6.

### 3.2. *VirtualSense*

VirtualSense [29] is an open-hardware ultra low-power sensor node featuring a Java-compatible virtual runtime environment. The software stack of VirtualSense is based on the Contiki operating system (OS) [30] and the Darjeeling Virtual Machine (VM) [31], suitably modified to make it possible for a Java programmer to fully exploit the low-power states of the underlying MCU, a MSP430F5418a as well as the data transceiver, a CC2520.

#### 3.2.1. *Microcontroller Unit*

VirtualSense features four categories of power states: *active*, *standby*, *sleep*, and *hibernation*. In *standby* the CPU is not powered, but the clock system is running and the unit is able to wake itself by means of timer interrupts. In *sleep* both the CPU and the clock system are turned off and the unit is woken up only by an external interrupt. In *hibernation* even the memory system is switched off and there is no data retention requiring a complete reboot of the OS at wake-up, together with a restore of the VM heap.

Power consumption varies significantly across different states of VirtualSense. In the *active* mode, the average power consumption is approximately 13mW when processing and 66mW for transmitting, while the consumption reduces to 14.67 $\mu$ W in *standby*, 1.32 $\mu$ W in *sleep*, and 0.36 $\mu$ W in *hibernation*. We also note that the time to transition from one state to another is non-negligible. Specifically, the transition to *active* is 25ms from *standby* and *sleep*, and 500ms from *hibernation*. WSN applications that do not need to maintain execution state can also use a memory-less hibernation mode (*ML Hibernation*), which does not preserve the VM state, reducing the wake-up time to 27ms. In summary, with effective DPM, power savings of several orders of magnitude can be achieved.



Table 1: Main parameters of ultrasonic and radio wake-up modules.

<b>WuR type</b>	<b>US</b>	<b>Radio</b>
<b>Radiation pattern</b>	55° at -6dB	Omnidirectional
<b>Frequency</b>	40KHz	868MHz
<b>Range</b>	15m	20m
<b>Tx power Standby</b>	40nW	690nW
<b>Tx power Active</b>	37mW	78mW
<b>Rx power Listening</b>	1640nW	462nW
<b>Rx power Decoding</b>	14 $\mu$ W	49 $\mu$ W
<b>Throughput</b>	20bps	10Kbps

### 3.2.2. Data Transceiver

VirtualSense provides several low-power communication features by exploiting the inactive modes (LPM1 and LPM2) of the CC2520 radio transceiver, as well as the hardware frame filtering (FF) capabilities that prevent the reception of non-intended packets.

The CC2520 is controlled by the main MCU through the serial peripheral interface bus (SPI) and six general purpose I/O pins (GPIOs). LPM2 is the lowest power consumption mode in which the digital voltage regulator is turned off, no clocks are running, no data is retained, and all analog modules are in the power down state. In this state, power consumption is around 13.5 $\mu$ W, but the embedded controller needs to be rebooted at wake-up. In LPM1, the digital voltage regulator is on, but no clocks are running. In this state, power consumption is around 3mW, while all data/configurations are retained and the analog modules can be controlled by the main MCU. During transmission, the power consumption of the entire transceiver, due to the RF module plus the embedded microcontroller, ranges from 48.6mW (at -18dBm output power) to 100.8mW (at +5dBm output power), while in standard receive mode the power consumption is 69.9mW.

The frame filtering function (FF) rejects non-intended frames not matching the local address. With FF enabled, the transceiver can be switched immediately to the LPM2 in case of a non-intended frame without a need to process the rest of the frame and wake up the MCU [32].

### 3.3. Wake-up Receivers

Table 1 reports the primary features of the two wake-up receivers (WuRs) used in this study, characterized by their use of ultrasound or radio waves.

Table 2: Amount of harvestable energy available in indoor environments. [33, 34]

Harvester	Power density
Photovoltaic	less than 10 $\mu\text{watt}/\text{cm}^2$
Electromagnetic	1-4 $\mu\text{watt}/\text{cm}^3$
Vibration (electrostatic)	3.8 $\mu\text{watt}/\text{cm}^2$
Radio Frequency	0.1 $\mu\text{watt}/\text{cm}^2$
Acoustic noise	0.003-0.096 $\mu\text{watt}/\text{cm}^3$

### 3.3.1. Ultrasonic WuR

The ultrasonic WuR considered in this work consists of a transmitter and of a receiver based on piezoelectric transducers working at 40KHz with a 2KHz bandwidth [21]. Nodes equipped with this WuR are triggered upon detection of an ultrasonic carrier signal. Optional selective triggering can be obtained by encoding 8 bit addresses through *On-Off-Key* (OOK) modulation of the carrier.

The receiver consumes 1640nW during listening periods and 14 $\mu$ W when decoding the preamble or the address. Power consumption by the transmitter is 40nW in standby and 37mW in active mode used to encode and transmit the wake-up signal. These parameters are compatible with a 15m operating range. The overall throughput of the system amounts to 20bps. Wake-up triggering times range from 50ms in non-addressing mode to 450ms for 8-bit addressing mode. Ultrasound transducers are notably directional in that the main radiation lobe is 55° at -6dB.

### 3.3.2. Radio WuR

The radio wake-up solution we adopted is based on a recently developed ultra low power radio WuR [19]. This module works at 868MHz within a 20m range; it is equipped with a high sensitivity receiver (-42dBm) with a power consumption of only 462nW (while listening) and of 49 $\mu$ W (during decoding). The power attributable to the transmitter is only 690nW in standby and 78mW during transmission. Similar to the ultrasonic WuR, this radio WuR features the capability to selectively address nodes, thanks to a mechanism based on OOK modulation. Wake-up times range from 130 $\mu$ s without addressing to 0.8ms for selective triggering with a 1 byte address (more than two orders of magnitude lower than that of the ultrasonic WuR) and the overall throughput is 10Kbps. The radiation pattern of the associated antenna is omnidirectional.

### 3.4. Energy Harvester

In this paper, our objective is to reduce node consumption to a rate that can be supported in a typical indoor environment with state of the art energy harvesting techniques. Table 2 outlines five such technologies, indicating the amount of harvestable energy each provides. These values are orders of magnitude lower than those in typical outdoor environments.

The WSNs deployed in the both case studies considered in this paper are exposed to low-intensity artificial light from lamps. This light can be readily converted by a photovoltaic cell into electrical energy. Therefore, photovoltaic cells optimized for low illuminance are an ideal choice. In this paper, we consider the Panasonic AM-1816 [35], a palm-sized cell designed to support small electronics indoors, even under low-intensity fluorescent lights.

## 4. Model-based Sensing

Traditionally, WSN nodes are designed to periodically *sense* the environment and then immediately *send* this data to a sink node. However, several applications interrupt this procedure, suppressing the transmission of many sensed data, either through prediction-based data collection techniques as briefly discussed in Section 2.1 or transmitting only significant events such as a detected forest fire, volcanic eruption, or landslide. The commonality between prediction-based data collection and event-driven applications is that both sense the data periodically, then process it to determine whether or not to send any data. When effective, these techniques avoid transmission in a large majority of the cases, nevertheless, the mote’s power hungry MCU must still be woken up to sense and process the sample. Turning on the MCU frequently incurs a significant cost for both switching to and staying in a high power mode. Our observations show that this sensing cost accounts for the most significant energy consumption once the communication costs are drastically reduced with a data reduction technique and a wake-up radio. To reduce this sensing cost, we introduce a novel *hardware-software* technique called *model-based sensing (MBS)*, which moves the sensing and processing tasks from relatively power hungry MCU to a dedicated hardware peripheral.

Figure 1 shows our approach with the MBS module containing sensors and an ultra-low power microcontroller on the hardware side and a data processing logic on the software side. This software can either implement the functionality of an event-driven application or a data reduction technique such as prediction-based data collection. In this paper we focus on the

latter, noting that MBS proposes *moving* the data modeling software from the primary MCU (labeled “application” in the figure) into the MBS module, allowing sampling and processing without requiring the expensive, primary MCU. Only when samples fall outside the model must the MCU be turned on, together with the primary radio transceiver, for transmission of a new model to the sink. We select the microcontroller for MBS module such that it consumes orders of magnitude less energy than main MCU, enabling significant energy savings.

**Implementation Details.** In this paper, we use Derivative-based Prediction [6] as the basis for MBS. The model is piece-wise linear and is computed based on  $m$  samples. Hence, each sample needs to be digitized, stored in memory, and processed in order to decide whether a new model must be transmitted or not. Processing entails integer sums, divisions, and comparisons that can be performed by an off-the-shelf ultra low power peripheral interface controller (PIC). The reference architecture proposed in this paper is based on Microchip PIC16F1825, which draws only 36nW in sleep mode with data retention and  $10.8\mu\text{W}$  while running at 32KHz. It accommodates 12 16-bit ADC channels and 1024 bytes of RAM, which can hold up to 512 ADC samples. We chose this PIC because its power consumption in processing and sleep modes is three orders of magnitude less than that of the MCU used in our VirtualSense mote.

Although, in principle, pure hardware implementations could be envisioned to minimize the number of components to be embedded into the sensing layer, the cost and the power consumption of state-of-the-art PICs do not motivate the development of application-specific integrated circuits.

## 5. Experimental Setup

Our unique combination of the technologies presented in Section 3 and Section 4 exploits the lowest consumption modes of the VirtualSense platform for as much time as possible, moving to a low-power MCU state between activities such as transmissions and receptions. Using DBP decreases the costs in multiple ways. Consider that in any system, a node transmits its own data, forwards data from other nodes, and unnecessarily overhears packets destined to other nodes. Each of these events requires the node and the primary data radio to be switched to a high power consumption mode. DBP reduces the total traffic in the network, thus reducing the frequency of all these events, and consequently increasing the time the node can remain in the lowest power mode. However, with DBP running as the main application on the mote, the main MCU still needs to process each

periodic sample, resulting in relatively high sampling costs. We have solved this problem by delegating the tasks of sampling and running DBP to a specialized ultra-low power sensing peripheral called MBS, obviating the need for expensive and frequent wake-ups of main MCU. Moreover, overheads related to idle listening and message overhearing are further reduced by using a wake-up receiver and the hardware frame filtering capabilities of the radio transceiver. Our multifaceted hardware software approach intends to radically reduce the power consumption of the WSNs.

The results presented in the next section come from a series of simulations with DBP, VirtualSense, wake-up receivers and MBS, performed with actual data collected from two real-world WSNs deployed in a road tunnel and an office building. The estimated power consumption is based on real, empirical measurements of power consumed by VirtualSense and the wake-up receivers in different power modes. This section offers details on both case studies, the power measurements and concludes with an estimation of energy harvested from photovoltaic cells, as our final goal is an energetically sustainable system.

### 5.1. Case Studies

The two case studies described in this section correspond to a real-world deployment and a testbed, representative of sparse and dense WSNs respectively. Both applications require periodic data samples to be shared with a central control system.

#### 5.1.1. TUNNEL: Adaptive Lighting in Road Tunnels

Our first case study is based on a pilot deployment in a real road tunnel in Trento, Italy [36]. In this 260 m tunnel, a WSN of 40 nodes is deployed to periodically measure the light levels. The nodes close to the entrance are exposed to sunlight, while nodes deep in the tunnel only detect the artificial light from the lighting system. In all cases, the light levels detected by the sensors every 30 s are transmitted over a 15-hop network, shown in Figure 2, to reach a gateway at the entrance of the tunnel. The values are then used by a control system to gradually adjust the intensity of the lamps throughout the tunnel to meet the legislated light levels. The control system was designed in collaboration with lighting engineers to tolerate a limited amount of data loss and to accommodate some degree of error in the quality of the sensed values.

In this paper, we evaluate the system’s power consumption with multiple hardware and software configurations. While the next section addresses the hardware configurations, here we consider the application layer, as it is

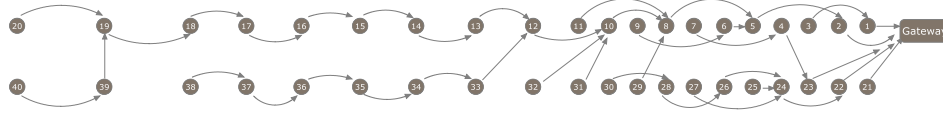


Figure 2: Layout of nodes in the tunnel and a sample data collection tree with a depth of 15 hops.

affected by the case study itself. Specifically, we must consider the amount of data reported by each node with and without the DBP data prediction algorithm. For this, we used actual data traces collected from the tunnel over a 47-day winter period, a total of 5,414,400 samples. Without DBP, we assume each sample is transmitted by each node immediately after being sampled. With DBP, instead, each node transmits only when the model changes. When running DBP only on the MBS module, we assume that the main MCU and data transceiver would not be turned on unless there is a model change. For the purposes of this study, we configured DBP to allow the predicted data values to deviate from the actual light values by at most 5% or 15 lux. Further, at most two consecutive samples can fall outside this error bound before a new model generation is triggered. These settings allow DBP to reduce the total sent traffic by 99.74% w.r.t. periodic reporting [4].

The data collection topology, shown in Figure 2, is generated using an implementation of unit disk graph model in Cooja simulator [37]. The communication range is set to 15m, the minimum range of the two WuRs in Table 1, while the interference range is set to double of the communication range. This topology along with the total network traffic is fed to the power consumption model described in Section 5.2 to estimate the power consumption.

#### 5.1.2. INTEL: *Indoor Environmental Monitoring*

Our second study is based on arguably one of the first publicly available datasets collected from a WSN, specifically an indoor, 54 node Mica2Dot deployment inside the Intel Berkeley Research Lab [38, 39]. The laboratory is approximately  $40 \times 31 \text{ m}^2$ . The dataset consists of 36 days worth of environmental data sampled every 31 s including light, temperature and humidity, resulting in 2.3 million values for each sensor type.

As with the tunnel, this data displays daily cycles, with light showing the most irregularities as the humans in the environment directly affected it. We configured DBP such that the predicted data values can deviate from the actual values by at most 5% or 15 lux for light, 1% for humidity

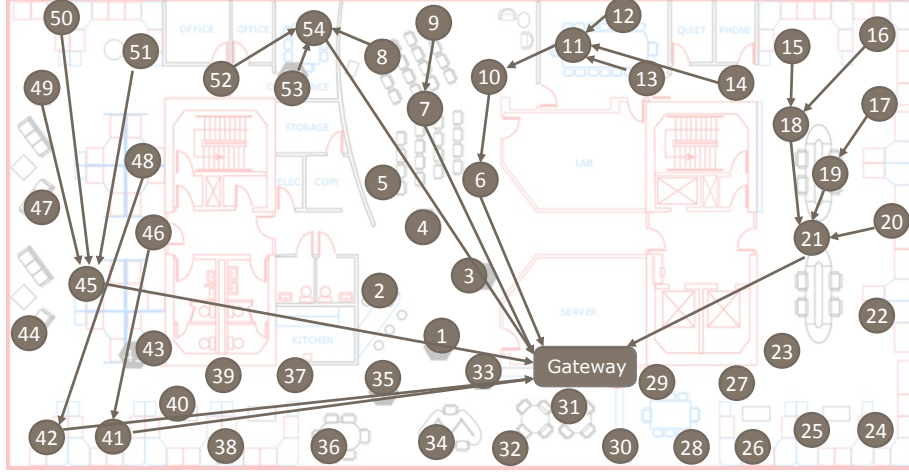


Figure 3: Layout of nodes in the Intel-lab testbed together with the data collection tree. For the sake of readability, direct connections between nodes not involved on multi-hop paths are not represented.

and 0.5 degree Celsius for temperature. This configuration reduced the periodic transmissions by 97.58% for light, 99.50% for humidity and 99.60% for temperature.

To model the real-world channel and interference conditions experienced by the main data transceiver, we use publicly available aggregate connectivity data [39]. Based on the provided link qualities, we construct a 4-hop collection tree, as shown in Figure 3. Given these link qualities and the total amount of traffic sent and forwarded over the collection tree, number of link level transmissions has been estimated and fed to our power consumption model described next.

## 5.2. Power Models and Simulations

Power simulations were obtained starting from the functional state diagram reported in Figure 4, which describes the behavior of a WSN node capable of exploiting idle periods during its workload in order to save power.

The state diagram is represented in terms of states and transitions. In particular, the *Wait* state represents a family of inactive modes fully accessible and exploitable by the dynamic power manager. In *Wait* state, the node reacts to three types of events: *i*) reception of an intended packet; *ii*) overhearing of an unintended packet; *iii*) wake up from timed interrupt for periodic tasks (i.e., sampling). The *Process* state is representative of activities such as sampling a given physical quantity and evaluating the need to

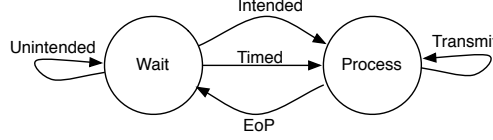


Figure 4: Reference functional state diagram.

transmit a sample according to the prediction strategy, or relaying an incoming packet toward destination (i.e the sink). When the processing ends (EoP), the node transitions back to the *Idle* state.

The functional state diagram allows us to derive the following five operating conditions accounting for node power consumption:

1. waiting;
2. hearing an unintended packet;
3. receiving and routing a packet;
4. waking up autonomously to sample then transmit;
5. waking up autonomously to sample but not transmit thanks to DBP.

The power consumption model was used to accurately estimate the consumption based on the runtime behavior of VirtualSense nodes [40]. In particular, we computed the average power consumption as a weighted average, where weights are represented by the actual occurrence rates of each operating condition as derived from real-world traffic data (for each of the two case studies). Indeed, the datasets obtained from the case studies were fed to the simulator, which captures the behavior and power consumption of each node, taking into account all the configuration options of the hardware and software components involved, as shown in Figure 1. In details, the simulator allows us to:

- select the low-power mode of the MCU;
- select the low-power modes of the data radio transceiver (with and without hardware frame filtering);
- choose between the ultrasonic or radio wake-up modules;
- activate either software DBP, or MBS.

When the wake-up receiver is not in use, we assume the primary radio runs the standard ContikiMAC protocol with a 100ms wake-up interval.



The power consumption values of the MCU [40], the radio transceiver [40], the ultrasonic [21] and the radio wake-up [19] modules, and the new MBS module were accurately characterized by means of real-world measurements.

It is worth noticing that the simulator takes traffic statistics from real INTEL testbed or from TUNNEL simulation in Cooja [37]. Hence, the radio and acoustic channels are not modeled within the simulator. Nevertheless, the effects of the data radio channel are inherently represented by the input data according to the models or experimental conditions used to generate them, as detailed in Section 5.1.

As for the radio/acoustic wake-up communication, all the experiments were conducted within the nominal operating conditions of the devices (in terms of range and orientation), where the effects of non-idealities are negligible. Within these conditions, the triggering reliability is near-perfect. Pushing the wake-up technologies beyond their nominal conditions would require a thorough analysis of the effects of false positives (false-negative) wake-up events on power consumption (data packet loss). Such analysis goes beyond the scope of this work.

### *5.3. Estimation of Harvestable Energy*

To estimate the harvestable energy for our nodes, we use the actual light values collected in our case studies. These light values are already expressed in lux for INTEL. The raw sensor values from TUNNEL sensors is also converted to lux, thanks to an accurate calibration process described in [36]. For estimating the harvestable energy, we assume that the same sensed illuminance is produced from a fluorescent light source and is incident on a Panasonic AM-1816, a palm-sized photovoltaic cell, which then is attached to a harvester described in [41] with a charging efficiency of 79%. Our model for harvestable energy is then derived by a piecewise linear model using empirical measurements available for the photovoltaic cell in [35] and the harvester in [41].

## **6. System Evaluation**

Our goal is to evaluate system power consumption with multiple different configurations of the hardware and software components of our architecture, as outlined in Figure 1. This section reports the achievable savings in both case studies, first discussing those enabled by various combinations of these components, then highlighting those achieved by the two WuRs and the novel MBS module.

### 6.1. Energy Savings

For our case studies, Tables 3 and 4 show 23 different configurations, each of which corresponds to a specific hardware states. The first row shows a standard node configuration that uses the *standby* mode of MCU and the *LPM1* mode of the data transceiver with no WuR. We consider this hardware configuration, in combination with a software layer that does not use the DBP prediction scheme as the baseline. Power consumption values, shown in  $\mu\text{W}$ , are computed as averages over all the nodes of the WSN being studied.

When the data is sent periodically, without DBP or MBS, energy efficiency is achieved by using a suitable low power hardware configuration. We observed a lifetime improvement of 19x and 92x (ID 11) for TUNNEL and INTEL respectively. The larger improvement for INTEL compared to TUNNEL is due to the higher node density of the former, which causes excessive overhearing and unnecessary triggering of WuRs in most hardware configurations, resulting in higher baseline consumption. The FF capabilities and addressing mode of the WuRs (Radioa and USa) are, therefore, more effective in mitigating this effect for the more dense INTEL deployment, enabling larger savings.

In all configurations, adding DBP reduces the network traffic, reducing power consumption. Additionally, as expected, increasing the use of low-power modes also reduces consumption. Without the WuRs (IDs 1-5), power consumption reductions are modest, approximately eight times for both case-studies, even when considering a configuration that exploits the *sleep* mode of the MCU and avoids unnecessary overhearing with the data radio (FF).

Intuitively, the addition of the WuR should have a significant impact by reducing idle listening in the TUNNEL and INTEL applications where data is generated roughly only twice per minute. However, the observed lifetime improvement varies significantly across the wake-up technologies. For example, in TUNNEL, the improvement is 19x for Radio (ID 11) compared to only 3.3x for US (ID 7). A better performance of Radio WuR is due to its three orders of magnitude better throughput and therefore a lower wake-up triggering cost w.r.t. the US WuR.

Instead, combining the radio WuR with the extremely low data rate of DBP (ID 11, DBP) results in a significant 463 times improvement, a result that is much larger than the improvements attainable by each technique in isolation. This remarkable result is a concrete demonstration of the benefits of eliminating idle listening with the very low transmission rates achieved

with DBP. Additionally, as the node no longer needs to forward frequent data packets on behalf of other nodes, it can spend more time between samples in the low power mode of the MCU. Without MBS, the sampling interval (30 s for TUNNEL and 31 s for INTEL) defines the maximum time a MCU can stay in a low power mode. MBS, instead, gets rid of this limitation by offloading the sampling task from main MCU, enabling even longer idle periods for the MCU. Maximally, with MBS, the MCU can stay in low power mode between two consecutive model updates. To offer an example, if a TUNNEL node does not observe significant variations in light throughout the night, a model will not change and the MCU can likely stay in a low power mode for this whole duration. These very long periods of idleness result in *three orders of magnitude* lifetime improvement. Specifically, the addition of MBS improves the lifetime from 463x to 2052x for TUNNEL (ID 11).

Missing entries in Tables 3 and 4 indicate infeasible hardware configuration. For example, the hibernation mode of the MCU has a shut-down and a wake-up time that cannot be accommodated with the sampling rates of the target applications when prediction-based data collection is not exploited to reduce the traffic. Similarly, memory-less hibernation modes are not compatible with the software implementation of DBP, which requires data retention. It is worth noting that all configurations become feasible with MBS, which achieves benefits by reducing the traffic, reducing the number of MCU state transitions, and providing an auxiliary memory in the sensing layer that enables ML hibernation.

In summary, these numerical results bolster our argument that the individual techniques of prediction-based data collection, DPM, MBS, and wake-up receivers, while individually capable of achieving improvements, are even more powerful when combined into a single system. The synergistic effect results in more than 3 orders of magnitude improvement in lifetime in both case studies presented here. The performance of the two WuRs and MBS under different hardware-software configurations is discussed in more detail in the following subsections.

## 6.2. Comparison of Wake-up Receivers

System performance is sensitive to the WuRs and the network traffic. As highlighted by Table 1, the radio WuR offers three orders of magnitude higher throughput than the US WuR and therefore can send and receive the triggering signal more quickly, saving energy. For this reason, the radio WuR *always* achieves higher energy efficiency than the US WuR.

Table 3: System-wide energy savings in the tunnel case study. The gray cells indicate the baseline for calculating the power consumption improvement ratio of all other configurations. (Hib.=Hibernation)

ID	Hardware Configuration			no-DBP		DBP		MBS	
	MCU	Data Transceiver	Wake-Up	[ $\mu$ W]	Ratio	[ $\mu$ W]	Ratio	[ $\mu$ W]	Ratio
1	Standby	LPM1	none	5989	1.0x	3460	1.7x	3453	1.7x
2	Standby	LPM2	none	3532	1.7x	759	7.9x	749	8.0x
3	Standby	LPM2+FF	none	2988	2.0x	759	7.9x	749	8.0x
4	Sleep	LPM2	none	3520	1.7x	746	8.0x	736	8.1x
5	Sleep	LPM2+FF	none	2976	2.0x	745	8.0x	736	8.1x
6	Sleep	LPM2	US	2346	2.6x	15.4	388x	5.4	1102x
7	Sleep	LPM2+FF	US	1795	3.3x	15.1	397x	5.1	1175x
8	Sleep	LPM2	USa	-	-	15.7	383x	5.7	1060x
9	Sleep	LPM2	Radio	1696	3.5x	13.8	435x	3.8	1591x
10	Sleep	LPM2+FF	Radio	1145	5.2x	13.4	446x	3.4	1748x
11	Sleep	LPM2	Radioa	318	<b>19x</b>	12.9	<b>463x</b>	2.9	<b>2052x</b>
12	Hib.	LPM2	US	-	-	1664	3.6x	57.3	104x
13	Hib.	LPM2+FF	US	-	-	1663	3.6x	57.0	105x
14	Hib.	LPM2	USa	-	-	1617	3.7x	10.3	582x
15	Hib.	LPM2	Radio	-	-	1663	3.6x	56.9	105x
16	Hib.	LPM2+FF	Radio	-	-	1663	3.6x	56.5	106x
17	Hib.	LPM2	Radioa	10042	0.6x	1617	3.7x	8.9	676x
18	ML Hib.	LPM2	US	2424	2.5x	-	-	4.5	1324x
19	ML Hib.	LPM2+FF	US	1873	3.2x	-	-	4.2	1431x
20	ML Hib.	LPM2	USa	-	-	-	-	4.7	1275x
21	ML Hib.	LPM2	Radio	1775	3.4x	-	-	4.1	1444x
22	ML Hib.	LPM2+FF	Radio	1224	4.9x	-	-	3.8	1572x
23	ML Hib.	LPM2	Radioa	328	18x	-	-	3.3	1837x

Table 5 reports the percentage reduction in power consumption because of using the most efficient configuration of the radio WuR over that of the US WuR. For the denser INTEL deployment, the addressing modes of the both WuRs (USa and Radioa) are more efficient than the broadcast mode (US and Radio), which unnecessarily wakes up many neighbors.

Compared to USa (ID 8, DBP), Radioa (ID 11, DBP) reduces the average node power consumption by 54.38% for light, 26.01% for humidity and 22.66% for temperature, as shown in Table 5. The larger improvements represent the datasets generating more DBP models. For example, INTEL light is the most difficult dataset to be predicted by DBP, generating 152 models per hour compared to only 32 and 26 packets per hour for humidity and temperature. Therefore, relatively larger number of DBP models benefit from the efficient radio wake-up triggering, resulting in more power savings compared to the humidity and the temperature datasets. A better energy efficiency of USa than Radioa is also clearly reflected on the lifetime improvement observed for the humidity and temperature datasets.

Table 4: System-wide energy savings in the Intel-lab case study (light). The gray cells indicate the baseline for calculating the power consumption improvement ratio of all other configurations. (Hib.=Hibernation)

ID	Hardware Configuration			no-DBP		DBP		MBS	
	MCU	Data Transceiver	Wake-Up	[ $\mu$ W]	Ratio	[ $\mu$ W]	Ratio	[ $\mu$ W]	Ratio
1	Standby	LPM1	none	6323	1.0x	3529	1.8x	3522	1.8x
2	Standby	LPM2	none	3900	1.6x	834	7.6x	825	7.7x
3	Standby	LPM2+FF	none	2996	2.1x	812	7.8x	803	7.9x
4	Sleep	LPM2	none	3888	1.6x	821	7.7x	812	7.8x
5	Sleep	LPM2+FF	none	2984	2.1x	799	7.9x	790	8.0x
6	Sleep	LPM2	US	2530	2.5x	75.1	84x	65.6	96x
7	Sleep	LPM2+FF	US	1615	3.9x	52.7	120x	43.3	146x
8	Sleep	LPM2	USa	-	-	30.3	208x	20.9	303x
9	Sleep	LPM2	Radio	2368	2.7x	69.9	91x	60.4	105x
10	Sleep	LPM2+FF	Radio	1453	4.3x	47.6	133x	38.1	166x
11	Sleep	LPM2	Radioa	68.7	<b>92x</b>	13.8	<b>457x</b>	4.4	<b>1442x</b>
12	Hib.	LPM2	US	-	-	4755	1.3x	3234	2.0x
13	Hib.	LPM2+FF	US	-	-	4733	1.3x	3212	2.0x
14	Hib.	LPM2	USa	-	-	1600	4.0x	79.1	80x
15	Hib.	LPM2	Radio	-	-	4752	1.3x	3230	2.0x
16	Hib.	LPM2+FF	Radio	-	-	4729	1.3x	3208	2.0x
17	Hib.	LPM2	Radioa	-	-	1585	4.0x	63.8	99x
18	ML Hib.	LPM2	US	2648	2.4x	-	-	67.5	94x
19	ML Hib.	LPM2+FF	US	1733	3.6x	-	-	45.4	140x
20	ML Hib.	LPM2	USa	-	-	-	-	20.0	316x
21	ML Hib.	LPM2	Radio	2487	2.5x	-	-	63.6	99x
22	ML Hib.	LPM2+FF	Radio	1572	4.0x	-	-	41.3	153x
23	ML Hib.	LPM2	Radioa	71.1	89x	-	-	4.7	1325x

For TUNNEL, the improvement achieved by the radio WuR (ID 11, DBP) over US WuR (ID 7, DBP) is significant (14.42%) but less than the one for the INTEL dataset. The efficiency advantage of the radio WuR over the US WuR is limited by two factors. First, DBP suppresses 99.74% periodic transmissions, generating *only* 13 packets per hour in the TUNNEL network on average. Second, most traffic comes from the nodes exposed to direct sunlight at the entrance of tunnel (e.g., 1-3 and 21-23). Because the gateway is deployed at the entrance, the direct links between these nodes and the sink can exploit the efficient WuRs over only a single hop in most cases and that also for ultra-low traffic. Both factors limit the advantage of the radio

Table 5: Benefits of using the the radio WuR over the US WuR.

Case study	Dataset	Avg. Network Data Rate	Power Savings
INTEL	Light	152 pkts/hour	54.38%
	Humidity	32 pkts/hour	26.01%
	Temperature	26 pkts/hour	22.66%
TUNNEL	Light	13 pkts/hour	14.42%

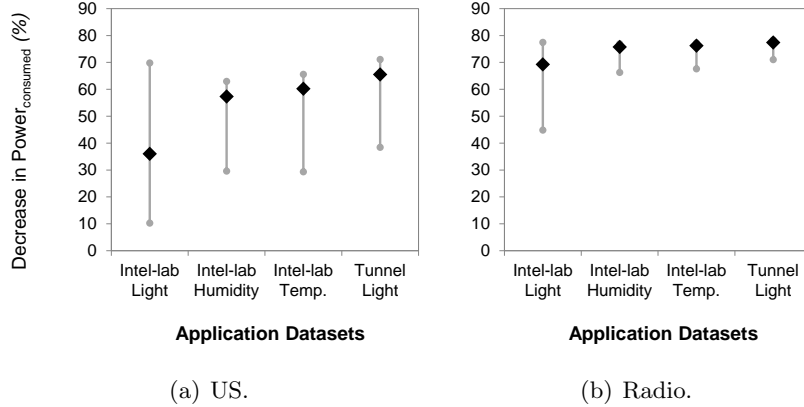


Figure 5: Average, minimum and maximum reduction in power consumption enabled by the separate sensing layer for the network nodes

WuR over the US WuR to 14.42% in this particular scenario. However, it is worth-mentioning that the WuRs achieve a remarkable two orders of magnitude reduction (397x in ID 7, 463x in ID 11) in power consumption compared to the base configuration by cutting the costs of idle listening and overhearing.

To sum up, we evaluated our architecture with two different WuRs, highlighting better energy efficiency of the radio WuR over the US WuR. In addition, an omni-directional nature of the radio WuR brings other advantages in that a data collection tree can be constructed on-demand and adapted to dynamic network conditions. By extending our evaluation of the US WuR [1] to the radio WuR, we show a considerable improvement in our system performance. Thanks to the modular nature of the VirtualSense mote, integrating new WuRs is straight-forward.

### 6.3. Performance of MBS

Thanks to the combination of DBP and WuRs, we greatly reduce the communication cost, leaving sampling cost to dominate the node consumption. Therefore, we now turn our attention to the ability of MBS in reducing the sampling cost. With MBS, a hardware peripheral can sense and process the samples without involving the power-hungry MCU of the mote as long as DBP needs not to update the model. This enables significant savings, as explained next.

Figure 5 shows the percentage, minimum and maximum power savings enabled by MBS for the network nodes. We notice that the average node power consumption improves remarkably from 35% to 76%, irrespective of the WuR technology. Moreover, the minimum improvement for a single

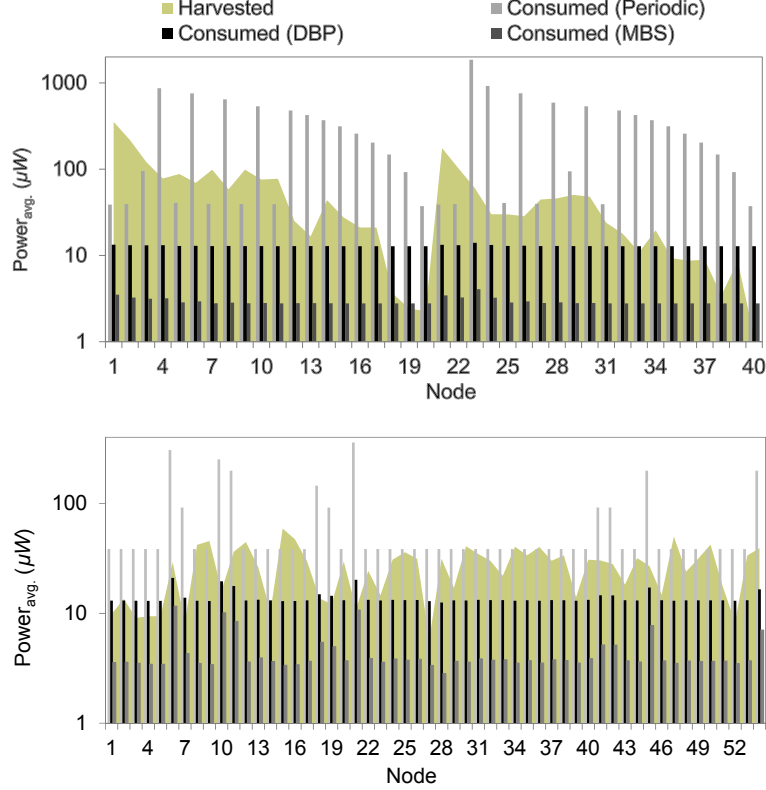


Figure 6: Comparison of energy consumed and harvested, plotted on a logarithmic scale for TUNNEL (up) and INTEL (down).

node is significant, above 45% when a radio WuR is used. The lower energy efficiency of US WuR and a higher DBP traffic rate for INTEL datasets shift the overall cost from sampling to communication, thus slightly dropping the percentage power savings enabled by our sampling peripheral. Overall, when MBS is combined with other techniques from our architecture, the average power consumption drops from milliwatts to a few microwatts. Specifically, Tables 3-4 highlight that the average power consumption is reduced merely to  $2.9\mu_W$  and  $4.4\mu_W$  (ID 11, MBS) for TUNNEL and INTEL respectively.

#### 6.4. Energetic Sustainability

Next we return to our initial motivation: namely the creation of an energy-neutral system with a reasonable, indoor energy harvester. Referring to Table 2, we note that the power density of an indoor harvester is typically

less than a few microwatts/cm<sup>2</sup>. Therefore, to keep the size of the harvester similar to the size of a node, the overall power consumption should be on the same order.

We first consider this specifically for the tunnel dataset. Figure 6 shows the power consumed in TUNNEL in three different software/hardware configurations against the energy harvested by a single photovoltaic cell (namely a Panasonic AM-1816 [35]). When nodes do not use DBP, all the sensed values are sent to the sink and most nodes consume more than they can harvest due to large forwarding overhead, making them not energetically sustainable. Using DBP without using MBS allows most nodes to consume less than they harvest. Nevertheless, several nodes deep in the tunnel harvest so little energy that they would need a large number of solar panels to be energetically sustainable. Overall, we would require 72 photovoltaic cells to sustain 40 nodes. Instead, if we consider the hardware sensing layer, MBS, almost all nodes can sustain their operation with only a single photovoltaic cell, reducing the requirement from 72 to only 44 cells, a significant savings.

We now turn our attention to INTEL network. Compared to TUNNEL, it is characterized by a shallow data collection topology with a maximum diameter of only four hops. As most nodes are directly connected to the gateway (see Figure 3) and only a few nodes relay any traffic, their overall power consumption is far less than the TUNNEL nodes. Furthermore, an analysis of light traces collected from Intel Research Lab. suggests that these nodes are exposed to a higher illuminance. It means that smaller photovoltaic cells can be used to achieve energy neutrality, save costs, and miniaturize the sensor node design. Figure 6 shows the energy harvested by the same but ten times smaller photovoltaic harvester than the one used for TUNNEL. When the nodes report periodic data, all forwarding nodes as well as many non-forwarding nodes consume more power than they can harvest. By decreasing the overall network traffic substantially, DBP reduces the consumed power below the harvested power for most but not all the nodes. MBS, instead, makes all the nodes energy neutral, enabling cost-efficient perpetual data collection.

## 7. Conclusion

Obtaining energy neutral operation for WSNs is a challenging task, especially in indoor settings, due to the sampling and communication tasks that require nodes to exit their extreme low power states. This paper introduced the concept of model-based sensing and demonstrated its effectiveness to reduce power consumption when combined with several other state-of-the-art



techniques. Specifically, advanced dynamic power management is adopted at each node, prediction-based data collection is used to reduce data traffic by avoiding the transmissions of data fitting the current model, wake-up receivers are used to avoid idle listening in asynchronous communication, and data storage and processing capabilities are granted to the hardware sampling layer to allow it to discard useless samples without waking up the main MCU.

Experimental results using data conducted in two real-world WSN case studies show that the synergistic application of the different techniques offers power saving of up to three orders of magnitude, representing a significant improvement with respect to the state of the art. Most important, the proposed approach brings the average power consumption of the WSNs used in our case studies within the power budget of indoor photovoltaic harvesters, thus achieving energy neutral operation. This study concretely demonstrates that combining hardware and software techniques is key to energy neutrality for energy harvesting WSNs deployed in indoor conditions.

## Acknowledgements

This work was partially funded by the European Institute of Innovation & Technology (EIT ICT Labs—Activity 15171.).

## References

- [1] A. Bogliolo, V. Freschi, E. Lattanzi, A. Murphy, U. Raza, Towards a true energetically sustainable wsn: A case study with prediction-based data collection and a wake-up receiver, in: *Industrial Embedded Systems (SIES)*, 2014 9th IEEE International Symposium on, 2014, pp. 21–28. doi:10.1109/SIES.2014.6871181.
- [2] B. Zhang, R. Simon, H. Aydin, Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling, *IEEE Transactions on Industrial Informatics* 9 (1) (2013) 514–526. doi:10.1109/TII.2011.2174247.
- [3] T. Xiang, Z. Chi, F. Li, J. Luo, L. Tang, L. Zhao, Y. Yang, Powering indoor sensing with airflows: A trinity of energy harvesting, synchronous duty-cycling, and sensing, in: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, SenSys '13*, ACM, New York, NY, USA, 2013, pp. 16:1–16:14. doi:10.1145/2517351.2517365.

- [4] U. Raza, A. Camerra, A. Murphy, T. Palpanas, G. Picco, Practical data prediction for real-world wireless sensor networks, *Knowledge and Data Engineering, IEEE Transactions on* 27 (8) (2015) 2231–2244. doi:10.1109/TKDE.2015.2411594.
- [5] G. Anastasi, M. Conti, M. D. Francesco, A. Passarella, Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks* 7 (3) (2009) 537 – 568. doi:j.adhoc.2008.06.003.
- [6] U. Raza, A. Camerra, A. Murphy, T. Palpanas, G. Picco, What does model-driven data acquisition really achieve in wireless sensor networks?, in: *Pervasive Computing and Communications (PerCom)*, 2012 IEEE International Conference on, 2012, pp. 85–94. doi:10.1109/PerCom.2012.6199853.
- [7] E. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic, Edge mining the internet of things, *Sensors Journal, IEEE* 13 (10) (2013) 3816–3825. doi:10.1109/JSEN.2013.2266895.
- [8] F. Aderohunmu, G. Paci, D. Brunelli, J. Deng, L. Benini, M. Purvis, An application-specific forecasting algorithm for extending wsn lifetime, in: *Distributed Computing in Sensor Systems (DCOSS)*, 2013 IEEE International Conference on, 2013, pp. 374–381. doi:10.1109/DCOSS.2013.51.
- [9] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, W. Hong, Model-driven data acquisition in sensor networks, in: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30, VLDB '04, VLDB Endowment*, 2004, pp. 588–599.
- [10] D. Chu, A. Deshpande, J. Hellerstein, W. Hong, Approximate data collection in sensor networks using probabilistic models, in: *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, 2006, pp. 48–48. doi:10.1109/ICDE.2006.21.
- [11] D. Tulone, S. Madden, Paq: Time series forecasting for approximate query answering in sensor networks, in: *Proceedings of the Third European Conference on Wireless Sensor Networks, EWSN'06*, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 21–37. doi:10.1007/11669463-5.
- [12] D. Tulone, S. Madden, An energy-efficient querying framework in sensor networks for detecting node similarities, in: *Proceedings of the 9th ACM International Symposium on Modeling Analysis and Simulation*

of Wireless and Mobile Systems, MSWiM '06, ACM, New York, NY, USA, 2006, pp. 191–300. doi:10.1145/1164717.1164768.

- [13] A. Jain, E. Y. Chang, Y.-F. Wang, Adaptive stream resource management using kalman filters, in: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD '04, ACM, New York, NY, USA, 2004, pp. 11–22. doi:10.1145/1007568.1007573. URL <http://doi.acm.org/10.1145/1007568.1007573>
- [14] E.-Y. Lin, J. Rabaey, A. Wolisz, Power-efficient rendez-vous schemes for dense wireless sensor networks, in: Communications, 2004 IEEE International Conference on, Vol. 7, 2004, pp. 3769–3776 Vol.7. doi:10.1109/ICC.2004.1313259.
- [15] C. Hambeck, S. Mahlknecht, T. Herndl, A 2.4 uw wake-up receiver for wireless sensor nodes with -71dbm sensitivity, in: Circuits and Systems (ISCAS), 2011 IEEE International Symposium on, 2011, pp. 534–537. doi:10.1109/ISCAS.2011.5937620.
- [16] X. Huang, P. Harpe, G. Dolmans, H. de Groot, A 915mhz ultra-low power wake-up receiver with scalable performance and power consumption, in: ESSCIRC (ESSCIRC), 2011 Proceedings of the, 2011, pp. 543–546. doi:10.1109/ESSCIRC.2011.6044942.
- [17] S. Marinkovic, E. Popovici, Nano-power wireless wake-up receiver with serial peripheral interface, Selected Areas in Communications, IEEE Journal on 29 (8) (2011) 1641–1647. doi:10.1109/JSAC.2011.110913.
- [18] C. Petrioli, D. Spenza, P. Tommasino, A. Trifiletti, A novel wake-up receiver with addressing capability for wireless sensor nodes, in: Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on, 2014, pp. 18–25. doi:10.1109/DCOSS.2014.9.
- [19] M. Magno, L. Benini, An ultra low power high sensitivity wake-up radio receiver with addressing capability, in: Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on, 2014, pp. 92–99. doi:10.1109/WiMOB.2014.6962155.
- [20] M. Magno, D. Boyle, D. Brunelli, E. Popovici, L. Benini, Ensuring survivability of resource-intensive sensor networks through ultra-low power overlays, Industrial Informatics, IEEE Transactions on 10 (2) (2014) 946–956. doi:10.1109/TII.2013.2295198.

- [21] E. Lattanzi, M. Dromedari, V. Freschi, A. Bogliolo, A Sub-uA Ultrasonic Wake-Up Trigger with Addressing Capability for Wireless Sensor Nodes, *ISRN Sensor Networks 2013*. doi:10.1155/2013/720817.
- [22] V. Jelicic, M. Magno, D. Brunelli, V. Bilas, L. Benini, Analytic comparison of wake-up receivers for wsns and benefits over the wake-on radio scheme, in: *Proceedings of the 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, PM2HW2N '12*, ACM, New York, NY, USA, 2012, pp. 99–106. doi:10.1145/2387191.2387206.
- [23] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. Gamm, L. Reindl, Has time come to switch from duty-cycled mac protocols to wake-up radio for wireless sensor networks?, *Networking, IEEE/ACM Transactions on PP (99) (2015) 1–1*. doi:10.1109/TNET.2014.2387314.
- [24] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, C. Petrioli, Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems, in: *Computer Communications (INFOCOM), 2015 IEEE Conference on*, 2015, pp. 522–530. doi:10.1109/INFOCOM.2015.7218419.
- [25] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, in: *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS IX*, ACM, New York, NY, USA, 2000, pp. 93–104.
- [26] J. Hill, D. Culler, Mica: a wireless platform for deeply embedded networks, *Micro, IEEE 22 (6) (2002) 12–24*. doi:10.1109/MM.2002.1134340.
- [27] J. Polastre, R. Szewczyk, D. Culler, Telos: enabling ultra-low power wireless research, in: *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, 2005, pp. 364–369. doi:10.1109/IPSN.2005.1440950.
- [28] M. P. Andersen, D. E. Culler, System Design Trade-Offs in a Next-Generation Embedded Wireless Platform, *Tech. Rep. UCB/EECS-2014-162*, EECS Department, University of California, Berkeley (Aug 2014).

- [29] E. Lattanzi, A. Bogliolo, VirtualSense: A Java-Based Open Platform for Ultra-Low-Power Wireless Sensor Nodes, *International Journal of Distributed Sensor Networks* 2012. doi:10.1155/2012/154737.
- [30] A. Dunkels, B. Gronvall, T. Voigt, Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors, in: *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, LCN '04*, IEEE Computer Society, Washington, DC, USA, 2004, pp. 455–462. doi:10.1109/LCN.2004.38.
- [31] N. Brouwers, K. Langendoen, P. Corke, Darjeeling, a feature-rich VM for the resource poor, in: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys '09*, ACM, New York, NY, USA, 2009, pp. 169–182. doi:10.1145/1644038.1644056.
- [32] E. Lattanzi, A. Bogliolo, Hardware filtering of non-intended frames for energy optimisation in wireless sensor networks., *International Journal of Sensor Networks* 15 (2) (2014) 121–129. doi:10.1504/IJSNET.2014.060725.
- [33] S. Basagni, M. Y. Naderi, C. Petrioli, D. Spenza, *Wireless Sensor Networks with Energy Harvesting*, in: *Mobile Ad Hoc Networking: The Cutting Edge Directions*, IEEE Series on Digital and Mobile Communication, John Wiley and Sons, Inc., Hoboken, NJ, 2013, Ch. 20, pp. 701–736.
- [34] B. H. Calhoun, D. C. Daly, N. Verma, D. F. Finchelstein, D. D. Wentzloff, A. Wang, S. Cho, A. P. Chandrakasan, Design considerations for ultra-low energy wireless microsensor nodes, *Computers, IEEE Transactions on* 54 (6) (2005) 727–740. doi:10.1109/TC.2005.98.
- [35] L. Panasonic Eco Solutions Amorton Co., Am-1816, <http://panasonic.net/energy/amorton/en/products/spec/AM-1816.html>.
- [36] M. Ceriotti, M. Corrà, L. D’Orazio, R. Doriguzzi, D. Facchin, S. Guna, G. P. Jesi, R. L. Cigno, L. Mottola, A. L. Murphy, M. Pescalli, G. P. Picco, D. Pregolato, C. Torghele, Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels, in: *Proc. of the Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2011.

- [37] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with cooja, in: Local Computer Networks, Proceedings 2006 31st IEEE Conference on, 2006, pp. 641–648. doi:10.1109/LCN.2006.322172.
- [38] P. Buonadonna, D. Gay, J. Hellerstein, W. H. Hong, S. Madden, Task: sensor network in a box, in: Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on, 2005, pp. 133–144. doi:10.1109/EWSN.2005.1462005.
- [39] S. R. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, Tinydb: An acquisitional query processing system for sensor networks, ACM Trans. Database Syst. 30 (1) (2005) 122–173. doi:10.1145/1061318.1061322.
- [40] A. Bogliolo, E. Lattanzi, V. Freschi, Idleness As a Resource in Energy-neutral WSNs, in: Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems, ENSSys '13, 2013, pp. 12:1–12:6. doi:10.1145/2534208.2534214.
- [41] D. Porcarelli, D. Brunelli, M. Magno, L. Benini, A multi-harvester architecture with hybrid storage devices and smart capabilities for low power systems, in: Power electronics, electrical drives, automation and motion (SPEEDAM), 2012 international symposium on, IEEE, 2012, pp. 946–951. doi:10.1109/SPEEDAM.2012.6264533.