EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

On the design of an energy-harvesting noise-sensing WSN mote

Wilson M Tan[1,2*] and Stephen A Jarvis[1]

Abstract

Wireless sensor networks (WSNs) could potentially help in the measurement and monitoring of noise levels, an important step in mitigating and fighting noise pollution. Unfortunately, the high energy required by the noise measurement process and the reliance of sensor motes on batteries make the management of noise-sensing WSNs cumbersome. Giving motes energy harvesting (EH) capabilities could alleviate such a problem, and several EH-WSNs have already been demonstrated. Nevertheless, the high-frequency nature of the data required to measure noise places significant additional challenges to the design of EH-WSNs. In this paper, we present a design and prototype for a mote extension which enables the mote to detect noise levels while being powered by energy harvesting. The noise level detection carried out by the system relies primarily on the concept of peak detection. Results of performance testing are presented. Aside from the hardware design and prototype, we also discuss methods of assigning charge times for application scenarios where there are multiple pulse loads. We also propose a new opportunistic method for charge time determination. Experiments demonstrate that the new method could improve analytically-derived duty cycles by at least 350%.

Keywords: Wireless sensor networks; Energy harvesting; Acoustics; Noise

Introduction

Noise pollution is becoming an increasing concern in many urban regions all over the world (for example, [1]). An important step in fighting and mitigating noise pollution is its classification and quantification. Efforts being made towards this goal include cellphone applications that measure noise and noise-related legislations (such as the European Union's Environmental Noise Directive [2] and the New York City Noise Code [3]).

Wireless sensor networks (WSNs) could potentially help with these efforts, as they could enable the simultaneous and continuous gathering of data over wide geographic regions. Several WSNs for noise pollution monitoring have been demonstrated [4-6]. WSN nodes ('motes') however, are usually powered by batteries which have to be frequently replaced. The length of time between battery replacements depends on the energy demand of the application: for those nodes that have energy-intensive sensors, or do lots of routing, this could be days, while for nodes that have simple sensors with very small duty cycles, this could be many months. The task of replacing these batteries on a regular basis makes the maintenance of such networks difficult.

As an alternative to batteries, WSNs could also be powered through energy harvesting (EH). Several EH-WSNs have been demonstrated [7-9]; nevertheless, despite these past successes, creating an EH-WSN for noise pollution monitoring is not straightforward. The difficulty lies in the nature of the data being gathered. While a temperature-sensing WSN could probably take a reading every minute and not lose accuracy, to measure noise, sound samples have to be taken. The human ear can hear frequencies of up to 20,000 Hz - to be able to digitally reconstruct a human-audible signal without missing any frequency component, samples should then be taken at the rate of the Nyquist frequency, or around 40,000 Hz. Sampling at such a high frequency is highly energy consuming, and the processing of the data gathered is challenging to implement on resource-constrained motes.

In this paper, we present a mote extension design which would enable a mote to detect noise while being powered by energy harvesting, specifically, solar energy harvesting.

*Correspondence: wilson.tan@warwick.ac.uk
[1]Department of Computer Science, University of Warwick, Coventry, UK
[2]Department of Computer Science, University of the Philippines, Quezon City, Philippines

While the system does *not* conform with the standards of measure utilized by any of the existing noise codes, such a network could still be immensely useful in many urban settings - in contrast to cellphone readings taken by individuals, the network could provide data and information that is uniform (since the the measuring devices would be of the same type) and properly contextualized (since each mote's location and orientation would be known). Such data could help in building noise maps, a goal shared by many 'smart city' initiatives all over the world (for example, [10]). In terms of administration, our design makes for a relatively low-cost and low-maintenance system, something that may not be currently achievable with WSNs that are designed to comply with existing noise codes.

A noise-sensing mote would have to regularly sample the microphone at high frequency, an operation that consumes a significant amount of current for an extended period of time. Using the radio for transmitting or receiving messages, or even just idle listening, also consumes a significant amount of energy. Such operations are called *pulse loads*, as they draw pulses of high current from the energy storage device. Since both operations have to be performed in a noise-sensing mote, it can be considered to be running a *multiple-pulse load application*.

Energy-harvesting WSN motes usually have a *boost capacitor* between itself and the energy storage device. This is to mitigate for the limits to the level of current that could be drawn from the energy storage device, imposed by the device's internal impedance [11]. Since the capacitor has a finite capacity, it has to undergo repeated cycles of charging (from the battery) and discharging (to the load). This means that while a high level of current draw is now possible, such a draw could still not be sustained for an indefinite period of time - after the boost capacitor charge is depleted, it has to be allowed to recharge or recover (in comparison, without a boost capacitor, the current draw may not be possible at all). When running pulse load applications, the charge time has to be carefully determined to ensure that the boost capacitor is sufficiently charged before drawing any current from it. Utilizing the right charge time is of paramount importance: set it too long and duty cycle (hence, performance) suffers, set it to a value too low and the system may not function at all. Fortunately, an analytical method is available for deriving the charge time of single-pulse applications. Such a method however, has not been derived before for multiple-pulse load applications.

This paper presents two primary contributions: firstly, a mote extension design for noise sensing in urban environments, and secondly, a discussion of charge time determination methods for applications with multiple pulse loads. Aside from deriving analytical-computational methods for charge time determination, we also propose our own method, called the *Opportunistic method*.

The remainder of the paper is organized as follows. The next section discusses the 'traditional' sound level measurement process as carried out by WSN motes. This is followed by a discussion of our design, the rationale behind it, and the derivation of the parameter values that were utilized in it. This is followed by a discussion of power management algorithms. We then discuss the implementation of the design, how it was tested and evaluated, and the results of the tests. In the second part of the paper, we discuss the methods available for assigning charge times to applications with multiple pulse loads, including our own proposed method. We then proceed to expound and describe the experiments carried out to test the effectiveness of the methods and the results of the said experiments. An overview of related and future work is then provided, after which we summarize and conclude the paper.

Sound level measurement in WSNs

Sound is a mechanical wave which uses air as a medium. As the wave travels, it induces pressure fluctuations which are then detected by the human ear, or in mechanical/electronic devices, a transducer. Greater energy in the wave translates to bigger pressure fluctuations, which humans then experience as the *loudness* of a sound. The human ear is an extremely sensitive device: the difference between the smallest and biggest pressures that the human ear can sense vary by 12-13 orders of magnitude. Representing such a huge range can be cumbersome in the linear scale, so sound pressure level (or loudness) is usually defined in a logarithmic scale, with the unit of *decibels* [12,13].

$$L_p(t) = 10 \times \log_{10}\left(\frac{p^2_{\mathrm{RMS}}(t)}{p^2_{\mathrm{ref}}}\right) (\mathrm{dB}) \tag{1}$$

In Equation 1, $L_p(t)$ is the instantaneous *sound pressure level* (SPL) of a sound, while $p_{\mathrm{RMS}}(t)$ is the root mean square (RMS) of the pressure. p_{RMS} is the standard reference pressure set at 20 μPa. 20 μPa is conventionally set as the minimum pressure detectable by the human ear.

$p_{\mathrm{RMS}}(t)$ is not truly instantaneous since the RMS has to be computed over a period of time. While the length of time over which the RMS must be computed is not standard, it must at least be equal to or longer than the period of the lowest frequency being measured. In a discrete-time system, assuming that $p(i)$ is the pressure measured at the sampling instance and N is the number of samples over which we are making the RMS computation, we have

$$p_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} p^2(i)} \tag{2}$$

While the length of time for the RMS computation could be adjusted, what is usually used in measuring noise

levels over extended periods of time is L_{eqT}, or the *time-averaged sound level*. L_{eqT} is taken from the average of successive p_{RMS} values. Again, assuming a discrete-time system and N as the number of p_{RMS} values taken over the time for which L_{eqT} is being computed, we have Equation 3 [12]:

$$L_{\mathrm{eqT}} = 10 \times \log_{10}\left(\frac{1}{N}\sum_{i=0}^{N-1}\frac{p_{\mathrm{RMS}}^2(i)}{p_{\mathrm{ref}}^2}\right)(\mathrm{dB}) \tag{3}$$

For the microcontroller of the mote to be able to calculate Equation 2, p should be in a form that is digital in nature: the pressure waveform therefore has to go through transformations. Firstly, there is the microphone/transducer, which senses the pressure fluctuations in the air using a thin membrane. The physical fluctuation of the membrane induces voltage fluctuations. The voltage fluctuation induced by the pressure fluctuation is defined by the *microphone sensitivity*, denoted by S in Equation 4 [14]:

$$S = 20 \times \log_{10}\left(\frac{E \times p_0}{E_{\mathrm{ref}} \times p}\right)(\mathrm{dB}) \tag{4}$$

E_{ref} is conventionally set to 1 V while p_0 is set to 1 Pa. E is the resulting voltage swing when pressure changes by p. With S defined (through a datasheet, for example), E could be easily derived from Equation 4.

The output of the microphone is a continuous voltage waveform. In most noise-measuring systems, the microphone output is also *A-weighted*, meaning its frequency components are attenuated or amplified to match how the human ear perceives different frequencies (since the human ear does not perceive frequencies equally [15]). The voltage waveform is also usually preamplified using an operational amplifier (op-amp)-based active amplifier before being processed by the microcontroller's analog-to-digital converter (ADC). The ADC samples the voltage waveform in discrete time steps and discretizes the voltage level into binary integers. The output of the ADC is a stream of binary integers, which are then stored in the microcontroller's memory. Two parameters define the operation of the ADC: the sampling rate and the word width. A higher sampling rate translates to a more faithful representation of the original signal. The word width is the number of bits available for representing the sampled value. For instance, if the word width is only two bits, the ADC would only be able to differentiate between four levels (since $2^2 = 4$). If the continuous voltage values vary between 0 and 1 V, values between 0 and 0.25 V would be encoded as 00 and values higher than 0.25 V but no higher than 0.5 V would be encoded as 01, etc.

The output of the ADC could already be processed by the microcontroller's CPU and used as input to Equation 2. While Equation 2 requires the sound pressure level, the ADC output would suffice as an input. Barring the precision loss introduced by the ADC, the relationship between the integer value and the original pressure level is linear: the output could be scaled later. Alternatively, the programmer could also choose to convert the ADC output into its *Pa* equivalent before having the CPU do the computation. The downsides of this approach are the extra computational steps and the need for floating point numbers, which are not supported by all microcontroller platforms.

Most noise codes rely on L_{eqT} values. It must be noted however, that L_{eqT} is not a perfect measure of noise, especially as experienced by people in an urban area. Since L_{eqT} takes the time average of noise measurement values, it is possible for intense but short noisy periods to be hidden by the measure [12]. Another important reason for not ignoring these short but noisy periods is the increasing amount of evidence which suggests that impulsive noise could be more damaging than noise that is more widespread temporally [16]. As an example, take an area that is generally very quiet but regularly suffers from aircraft noise. The L_{eqT} measurement in this case would be very low and would not reflect the disturbance caused by the aircraft passing overhead. In such a situation, the *peak* noise level detected within a time period might actually be a more informative and useful metric.

Design

Design rationale

While most of the previous work that involved sound sensing has done so by recording the sound waveform [5,17,18], we have opted for an approach that is centered on a peak detector, for the reasons cited above. This means that our system does not produce SPL values, but the peak noise level recorded within a period of time. Such information is insufficient for the requirements of most existing noise codes, but as discussed, such information, combined with an energy-harvesting/battery-free operation feature, would suit noise sensing in many urban settings.

Our decision to adapt a peak detector-based design is also motivated by the limited capabilities of the TelosB [19] - in a previous work [20], we demonstrated how at 10,000 Hz sampling rate, the limited memory space of the MSP430 [21,22] microcontroller only enables continuous sampling of up to 0.4 s. In comparison, for many systems that monitor L_{eqT}, p_{RMS} is usually computed over 1-s intervals [16]. It must be noted that L_{eqT} could still be computed despite this limitation because the 1-s sampling interval is not standard, and the ADC could be simply be activated again for another set of readings. There would potentially be a gap between the sets of readings (which would be a function of the microcontroller processing

speed), but even that could be minimized by parallelizing processing and sampling using direct memory access (DMA). Even with DMA however, two problems still remain. Firstly, the need to compute the RMS of the samples every 0.4 s would consume a lot of energy over time. Secondly, and more significantly, the number of such readings that could be accumulated would be severely constrained by the amount of space - it must be remembered that space is one of the factors that contributed to the 0.4-s limit in the first place. This could be alleviated by shorter sampling windows (i.e., shorter sampling sequences) or by frequently sending data to the sink. The former, however, would negatively affect the accuracy of the readings, while the latter would be costly in terms of energy.

In comparison, a peak detector-based system does not record the waveform at all and only generates a digital reading at the end of a sampling period. A significantly larger number of readings could then be stored, resulting in greater flexibility in how and when such readings are processed. For example, assuming that time bounds requirements are met, readings could be accumulated so that several are sent to the base station in a single packet. The timing of sending could also be dynamically set to coincide with periods of high energy generation, helping to keep the system in energy-neutral operation. Alternatively, the packet sent could be externally triggered, with the data pulled out of the mote by a mobile sink or data mule. Such options are simply very limited or not available in systems that are severely constrained in the amount of data that they could store.

It must be noted that a peak detector-based design does not necessarily consume less power than a design which carries out wave sampling. The system unfortunately still has to duty cycle, meaning it cannot be in the active state all of the time. Ways of alleviating this limitation will be discussed later. While the ADC is no longer continuously active and sampling the microphone at a high frequency, a new sub-circuit is added to the system, one which continuously runs during the active part of the operation cycle. Our measurements show that the power consumption of the sub-circuit is at par with (or just slightly smaller than that of) an ADC at a high-frequency sampling operation. Nevertheless, the flexibilities afforded by a peak detector-based system gives significantly more options for power management algorithms which could lead to energy savings in the long run.

Basic design

For our work, we used the ultra-low-power wireless sensor module TelosB [19]. TelosB was designed at UC Berkeley with three goals in mind: minimal power consumption, ease of use, and increased software and hardware robustness. It uses the 16-bit Texas Instruments MSP430 microcontroller [21,22] and the 2.4 GHz IEEE 802.15.4 compliant RF transceiver Chipcon CC2420 radio [23]. The TelosB motes that we used for this work were manufactured by Advanticsys [24] and marketed as CM5000 motes. Our TelosB motes run TinyOS, or Tiny Microthreading Operating System [25,26].

For the energy-harvesting component of our setup, we utilized the CBC-EVAL-09 [27]. The CBC-EVAL-09 is an evaluation kit manufactured by Cymbet Corporation (Elk River, MN, USA). It features several energy-harvesting transducers, along with with the EnerChip EP CBC915 Energy Processor [28] and the EnerChip CBC51100 100 uAh solid state battery module (with two EnerChip CC CBC3105 [29] solid state batteries connected in parallel).

The EnerChip EP CBC915 Energy Processor [28] serves as an interface between the transducers and the energy storage device. It employs advanced maximum power point tracking tracking algorithms, constantly matching the output impedance of the energy-harvesting transducers, thus ensuring high-efficiency energy conversion. The EnerChip EP CBC915 Energy Processor also facilitates communication with the microcontroller, providing information such as state-of-charge estimates and a calibration function.

For the sensor design, we utilized an ADMP401 MEMS microphone [30]. Compared to conventional electret microphones, MEMS microphones offer the advantage of minimal size, lower power usage, and a better signal-to-noise ratio (SNR) [31]. The microphone output is preamplified by an op-amp-based inverting amplifier with a gain of 100. The preamplified output is then fed into a peak detector circuit with a load-isolated storage capacitor. This peak detector design is more suited for long hold times than the simpler single op-amp-based alternative. The details of the peak detector design are discussed in detail in [32-34]. The output of the peak detector is connected to ADC channel 0 of the TelosB, where it is sampled at the end of a sensing period. All three operational amplifiers (op-amps) utilized in the design are OPA344s [35].

To facilitate duty cycling, the supply lines of the microphone, the preamplifier, and the peak detector are gated by the high-side switch ADP194 [36]. The ADP194 is digitally controlled by the TelosB using a digital I/O pin. During the sensing period, the *Enable* pin of the ADP194 is set to *high* by the TelosB, enabling the current to flow to the microphone, the preamplifier, and the peak detector.

The design also features a MAX323 [37] digital switch, which provides a digitally switched line between the storage capacitor and a discharge resistor (whose other end is connected to the ground). The switch is added since the sudden surge of power to the peak detector

causes it to have an overshot output in the beginning of a sampling period. To return the output to its normal level, a 'corrective discharge' is carried out by connecting the capacitor to the discharge resistor for a few milliseconds. Like the ADP194, the MAX323 is digitally controlled by the TelosB using a digital I/O pin.

The schematic of the expansion board (which contains the microphone, preamplifier, and peak detector) is shown in Figure 1.

Design parameters - sensing side

The first design parameter that needed to be derived was the value of C_{Ch}, or the storage capacitor. The storage capacitor stores the voltage that corresponds to the maximum loudness of sound that has been observed within an active period. It is connected directly to the ADC port of the microcontroller, which samples the voltage level of C_{Ch} at the end of an active period. The C_{Ch} is discharged after sampling so that it could start at a very low level in the next active period. It must be noted that the *storage capacitor* is different from the *boost capacitor*, the value of which would also be derived in a later subsection. The size of the storage capacitor depends on the charging speed required of the system, which in turn depends on the desired sound frequency range covered by the detector. The maximum rate at which the the voltage across C_{Ch} could change is either

$$\frac{dV_{\text{Ch}}}{dt} = \text{SR}_1 \tag{5}$$

or

$$\frac{dV_{\text{Ch}}}{dt} = \frac{\text{IO}_{\text{max}}}{C_{\text{Ch}}}, \tag{6}$$

whichever is smaller [32]. The values of SR_1 and IO_{max}, or the slew rate and short-circuit current of *OP-AMP 2* in Figure 1, are specified by the datasheet [35] as $\frac{0.8\text{V}}{\mu\text{s}}$ and 15 mA, respectively. The C_{Ch} is then dependent on the desired rate of voltage change. To derive this, we assume a maximum sound frequency of 10,000 Hz and require the system to be able to swing from the quiescent level to the upper rail (a 1.5 V swing) within a single cycle. A maximum frequency of 10,000 Hz was chosen for the initial design because most of the frequencies to which the human ear is sensitive can be found below 10,000 Hz [15]. Referring to Figure 2, assuming that $V_{\text{Swing}} \approx V_{\text{Amplitude}}$ and $t_1 \approx t_2$, we can simplify the computation of $\frac{dV_{\text{Ch}}}{dt}$ as

$$\frac{dV_{\text{Ch}}}{dt} = \frac{V_{\text{Swing}}}{2 \times t_1} \tag{7}$$

Remembering that t_1 is $\frac{1}{\frac{10000}{4}}$, Equation 7 gives $\frac{30{,}000\text{V}}{\text{s}}$, which is smaller than SR_1. This means that the speed is supportable by the operational amplifier subject to the correct C_{Ch} value. Substituting this value into Equation 6 and solving for C_{Ch} gives us 0.5 μF.

Ideally, the output of the peak detector should remain stable until a higher peak than the previous is seen in the input. In reality, the peak detector output decays over time due to leakage current (an effect called *voltage droop* [32]). The rate of discharge of C_{Ch} is

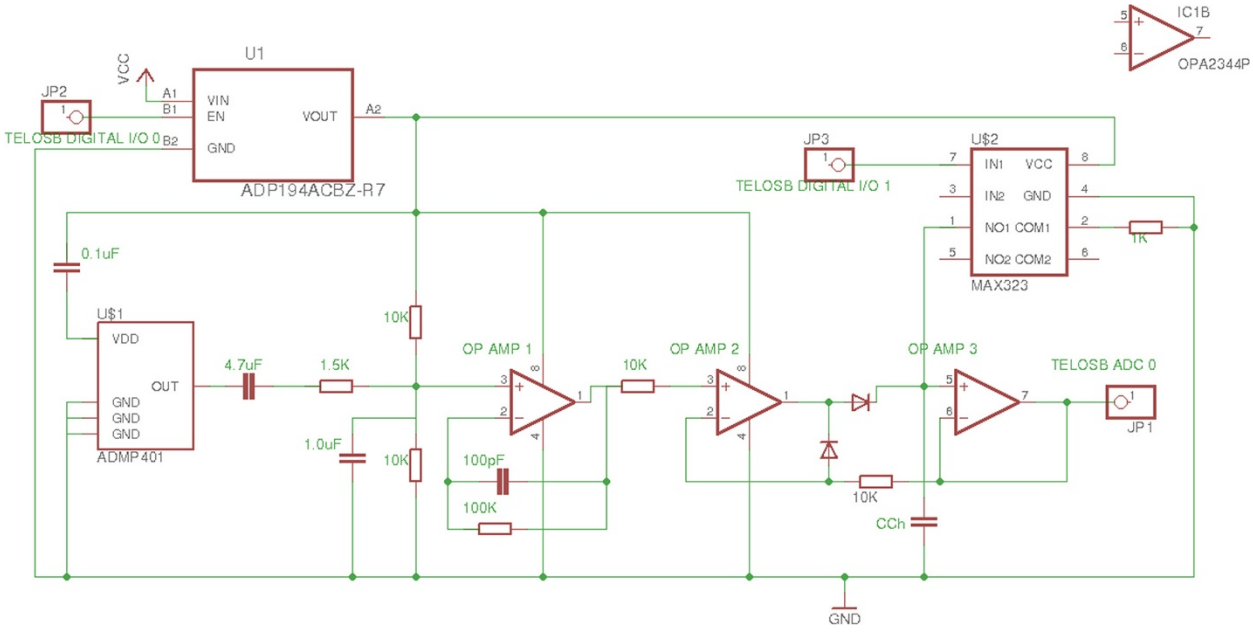

Figure 1 Design schematic.

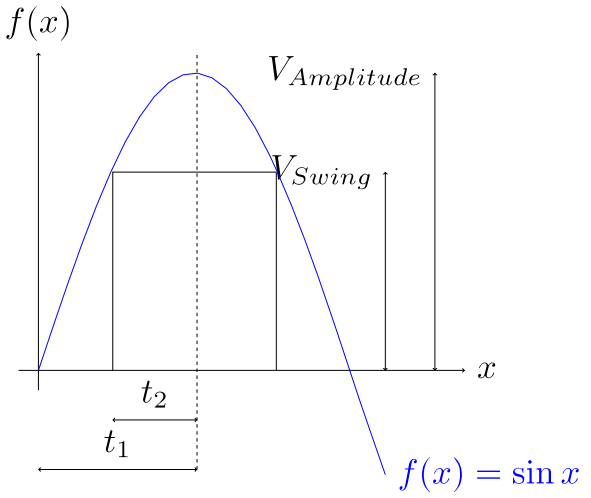

Figure 2 Diagram demonstrating simplifying assumptions adopted for parameter computation of C_{Ch} size.

$$\text{Voltage droop} = \frac{I_{\text{lk}}}{C_{\text{Ch}}}, \tag{8}$$

where I_{lk} is the leakage current. There are five sources of leakage current, namely: capacitor leakage, printed circuit board leakage, op-amp leakage, diode leakage, and reset switch leakage.

With the exception of op-amp leakage and diode leakage, all components of I_{lk} are difficult to quantify. Diode leakage is negligible since an ultra-low-leakage diode was utilized for the design. That leaves the op-amp leakage or the input bias current of the op-amp, which the datasheet gives as 10 pA maximum. Thus, $I_{\text{lk}} = 10$ pA. Using Equation 7 and the value derived for C_{Ch}, this gives us a voltage droop of $\frac{0.00002\text{V}}{\text{s}}$.

The voltage droop simply gives the rate at which the output of the peak detector decays over time. The actual decrease in the output depends on how much time has elapsed since the peak value was stored by the circuit. Since we do not know the exact time at which the peak detector stored a new peak value, this introduces uncertainty in the output. This output is further affected by the length of the active period (which is dictated by the duty cycle, to be discussed later). Since the microcontroller does not sample the peak detector output until the end of an active period, the longer the active period, the more opportunity there is for the inaccuracy of the peak detector output value to increase - this is especially true for peak values that are recorded very early on in the active period.

Design parameters - power supply side

The amount of current that could be drawn from an energy storage device is limited by the device's internal impedance. To compensate for this, a capacitor is usually inserted between the energy storage device and the load. This capacitor would be called the *boost capacitor*, to differentiate it from the *storage capacitor* whose size was derived in a previous subsection. For the sake of brevity however, all references to 'capacitor' should be taken to mean 'boost capacitor'. It must be noted that since the electrical charge has to be transferred from the energy storage device to the capacitor, it is still impossible to supply a high amount of current to the load for indefinite periods of time.

For the same level of current draw, a longer draw period would necessitate using a larger capacitor to store a greater amount of electrical charge from the primary energy storage device. However, a larger capacitor also takes longer to charge - therefore, the intervals *between* current draws, which we also call *charge time*, would also increase.

The relationship between the capacitor size, the level of current draw (in milliamperes), the length of the current draw, and the interval between draws could be derived analytically and computed: for instance, the energy storage system that we utilize for our work, the EnerChip CC CBC3105 [29], specifies through an application note [11] a formula for determining the capacitor size needed for supporting a specified level of current draw for a specified length of time. We state this in Equation 9. The equation for R_{Load}, which is a variable in Equation 9, is defined in Equation 10. Equation 9 and Equation 10's variables are defined in Table 1. For variables whose values remain constant across different computations, their values are specified in Table 2.

$$C = \frac{t_{\text{Draw}}}{R_{\text{Load}} \times ln\left(\frac{V_{\text{Max}}}{V_{\text{Min}}}\right)} \tag{9}$$

$$R_{\text{Load}} = \frac{V_{\text{out (average)}}}{I_{\text{pulse}}} \tag{10}$$

The formula for the charge time of a given specific capacitor is also given by [11], and is stated here in Equation 11. Equation 11's variables are also defined in Table 1.

$$t_{\text{Charge}} = -R_{\text{Bat}} \times C \times ln\left(\frac{V_{\text{Max}} - V_{\text{Chg}}}{V_{\text{Min}} - V_{\text{Chg}}}\right) \tag{11}$$

Solving Equation 9 and Equation 11, for example, a system utilizing a 5,000 μF capacitor should be able to support a current draw of 20 mA for 210 ms with a charge time of 45 s.

Defining duty cycle as the proportion of time the system is awake or active in a cycle, we now have Equation 12

$$\text{dutycycle} = \frac{t_{\text{Draw}}}{t_{\text{Draw}} + t_{\text{Charge}}} \tag{12}$$

Table 1 Equation 9 - Equation 12 variables

Variable	Description
C	Capacitance of external capacitor in parallel with the battery
t_{Draw}	Length of time that the current would be drawn
R_{Load}	Load resistance
V_{Max}	Final capacitor voltage that must be attained before next current draw
V_{Min}	Initial voltage when charging begins
t_{Charge}	Capacitor charge time
R_{Bat}	Battery resistance
V_{Chg}	Applied charging voltage on the capacitor
$V_{\text{out (average)}}$	Average voltage across the load during the current draw
I_{pulse}	Level of current draw(in *Amperes*)

Substituting Equation 9 and Equation 11:

$$\text{dutycycle} = \frac{R_{\text{Load}} \times ln\frac{V_{\text{Max}}}{V_{\text{Min}}}}{R_{\text{Load}} \times ln\frac{V_{\text{Max}}}{V_{\text{Min}}} - R_{\text{Bat}} \times ln\frac{V_{\text{Max}}-V_{\text{Chg}}}{V_{\text{Min}}-V_{\text{Chg}}}} \quad (13)$$

It must be noted that the only variable defined by the load which appears in Equation 13 is R_{Load}. Referring to Equation 10, only I_{pulse} determines the duty cycle of the system. The capacitor, in particular, does not figure in Equation 12. A larger capacitor would enable a longer draw time, but its ratio to the sum of the charge time and the draw time (the *total* period) would always be the same.

In some applications, the length of the current draw may be user-definable. An example of such a load would be the microphone. To have a longer draw time, we may opt for a larger capacitor, but the interval between draws would proportionately increase as well. In other applications, the length of the current draw is already defined. An example of such a load is a radio which has to send a message to a receiver which is duty cycling using low-power listening (*LPL*) [38]. If the wake-up interval of the receiver has already been decided or set, the designer has no choice on the size of the capacitor that must be installed on the sender.

Traditionally, duty cycling was seen as a necessity imposed by the limited amount of harvested energy. As can be seen in Equation 13 however, the hardware imposes its own cap on the duty cycle feasible. It is interesting to note that neither the storage size nor the solar panel output figures in Equation 13. Therefore, using larger batteries or larger solar panels will not help in solving the limit imposed by Equation 13.

Table 2 Equation 9 to Equation 12 variables and values

Variable	Value
V_{Max}	3.8 V
V_{Min}	1.8 V
R_{Bat}	7000 ohms
V_{Chg}	4.1 V
$V_{\text{out (average)}}$	3.4 V

If higher duty cycles than that imposed by Equation 13 are desired or needed, a possible solution would be to co-locate several sensor nodes. Enough nodes should be co-located so that the temporal coverage required is satisfied. If, for example, Equation 13 imposes a limit of 25% and 100% duty cycle is required, four nodes should be co-located, with the nodes taking turns in being active. The primary challenge in this solution would be the tight synchronization that would be required of the co-located nodes, and its main disadvantage would be the cost. A schematic for node co-location is shown in Figure 3.

Another possible solution would be to use one node with several energy-harvesting systems, each with its own solar panel and energy storage device. Upon depletion, the node would simply switch energy-harvesting systems, allowing the previously used system to recharge. As with the previous example, if there is a 25% limit, but a 100% duty cycle is required, four energy-harvesting systems would now be required. This would be less costly than the previous solution, and no synchronization would be needed, but there is the challenge of ensuring that the multiplexed power supply would be able to provide a supply level that is stable enough for continuous system operation. To the best of our knowledge, switched or multiplexed power supplies have never been used before in the context of overcoming duty cycle limitations in WSN nodes. The closest previous work is [39], where different subsystems in an embedded system were allowed to be powered simultaneously by different energy sources in an attempt to maximize energy harvesting efficiency. The schematic for node co-location is shown in Figure 4.

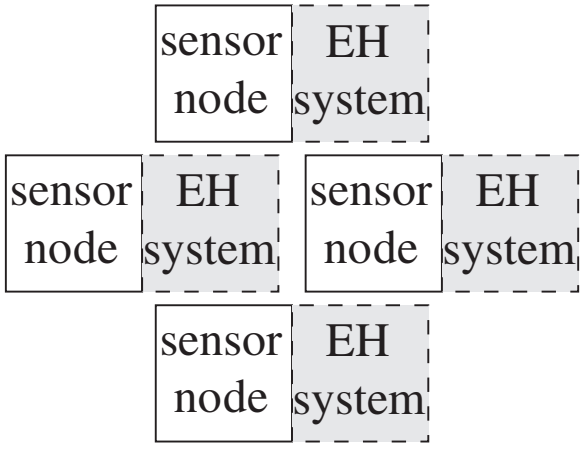

Figure 3 Node co-location schematic.

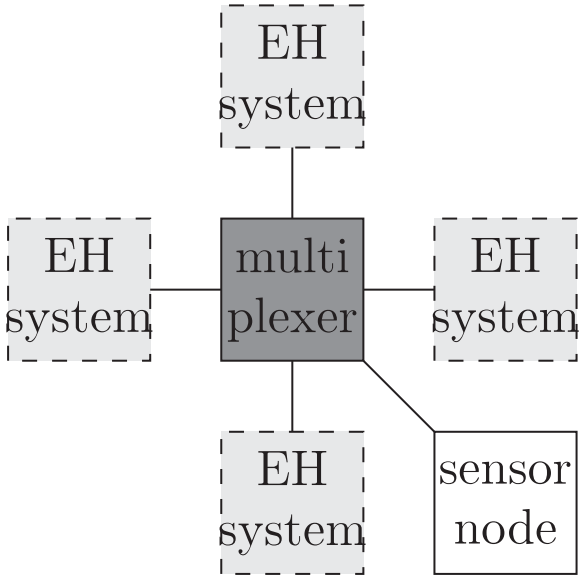

Figure 4 Power supply multiplexing schematic.

It must be noted that node co-location and power supply multiplexing only change the upper limit imposed by the physical system with regard to the maximum duty cycle achievable. The *actual* achieved duty cycle is still a function of the harvested energy and the energy usage pattern of the application. The amount of harvested (and stored) energy is affected by the node's location, harvesting efficiency, solar panel size, and storage capacity. The energy usage pattern of an application is usually tailored to the energy state of the system and the environment by power management algorithms. Power management algorithms are discussed next.

The microphone, preamplifier, and peak detector portion of our circuit was measured to have a collective current draw of 0.8 mA, translating to a duty cycle of 9.9%. We chose the boost capacitor size of 11,000 μF for a draw time of 10.05 s and charge time of 91.0 s. We believe that such values give an acceptable trade-off between the length of the draw time and the interval between the draws.

Power management

For an actual deployment to function, the motes need a power management algorithm. The power management algorithm manages the power consumption of the system, ensuring that system operation stays stable and functional despite variations in harvested energy. An obvious task for the power management algorithm would be adapting the system operation to the diurnal cycle, making sure that enough energy is harvested during the day so that the system remains functional even at night. The choice of power management algorithm would depend on several factors: the application type, the network topology, and the hardware available. We do not include a specific algorithm since the mote could be used for different types of applications and different network topologies. We do, however, discuss the factors that guide the choice and design of an appropriate power management algorithm.

Application type

Strictly speaking, a power management algorithm is any algorithm which changes an aspect of the system to meet energy constraints. What aspect is actually changed could vary from one algorithm to the next. For systems that regularly sample a single sensor, what is usually varied is the duty cycle. In [40], for example, the algorithm takes into account the predicted energy that would be harvested and assigns duty cycles to time periods called *frames*. The duty cycle assignment ensures that the node only consumes what it harvests, resulting in an infinite lifetime, or a mode of operation called *energy-neutral operation*. It also makes provision for when the actual harvested energy deviates from that which was predicted, increasing the duty cycles in subsequent frames when more energy is harvested than predicted and reducing the duty cycles when less energy is harvested. A power management algorithm such as that featured in [40] would be a possible good fit for systems that utilize our mote, although modifications are necessary. For example, [40] assumes that there is a utility level associated with the duty cycle and that there is a maximum duty cycle beyond which the utility would no longer increase. Even without an explicit consideration of utility however, in our system, the hardware already explicitly imposes a limit to the duty cycle. Such a limit would have to be taken into account when using the algorithm in [40].

Other power management algorithms (such as [41]) assume that the processor or microcontroller is capable of dynamic voltage and frequency scaling (DVFS) and thus changes the speed of the processor or microcontroller in accordance with the energy state of the system. This is not applicable to our system since the microcontroller of the TelosB, like other low-power microcontrollers, does not have the DVFS feature.

A mote could also possibly change the sensor or set of sensors being used in response to the energy state [42]. This is more applicable to motes that use multiple sensors, which is currently not the case in our prototype (although it could be easily extended with other sensors; for instance, one could simply activate one or more of the onboard sensors of the TelosB).

Application or task versioning [43] is also a possible power management technique. In application versioning, the application running on the mote changes depending on the energy state. This, however, may not be applicable to storage space-constrained systems like those used in WSNs.

Other algorithms facilitate power management by controlling the sequence or start time of tasks [44]. Algorithms that belong to this family are more suited for embedded systems that react to real-time events, rather than those designed for regular data gathering.

Network topology and configuration

The network topology and configuration also have an effect on the power management algorithm as it determines the additional tasks that a node has to do, in addition to sensing and sending its own data. The tasks change the characteristics of the application (for example, its periodicity) and the power management techniques that are applicable.

Topology-wise, networks could be divided into two: single-hop networks and multi-hop networks.

Nodes in multi-hop networks (except leaf nodes) have the additional task of acting as a relay for other nodes - therefore, they do not just send messages, but receive messages as well. How much energy is spent on this task depends largely on the number of other nodes the node is serving as a relay for, and the media access control (MAC) protocol utilized. The nodes could communicate either synchronously or asynchronously. In synchronous communication, message transmission consumes relatively little energy, but there is the challenge of regular clock synchronization (which consumes energy itself). Such an approach is utilized in [45]. Asynchronous communication like LPL does not need synchronization but could potentially consume more energy than synchronous communication (in LPL, the energy is usually spent by the sender, mainly in sending the prolonged preamble) and could possibly suffer from congestion.

Nodes in single-hop networks do not have to relay messages for other nodes, but depending on the terrain and deployment environment, such a topology may not always be feasible. Single-hop networks could be push-based, where nodes regularly send their data to a sink, or pull-based, where the nodes only send data after receiving a request to do so. The latter is usually used in systems where the base station is mobile and travels around the deployment area collecting accumulated data from the nodes.

Hardware

The hardware available for energy measurement/estimation affects the choice of power management algorithm. All power management algorithms assume the availability of some sort of information about the energy state of the system or the environment, and the availability and accuracy of that information depends on the hardware.

Kansal et al. [40] for instance, assume that the amount of energy being harvested is always known, a reasonable requirement given that the Heliomote [46], for which the algorithm in [40] was designed, is equipped with dedicated circuitry that measures the voltage and current (hence, power and energy) coming from the solar panels. Without dedicated hardware, motes could measure the energy state of the system by measuring the voltage level of the energy buffer (usually a rechargeable battery or capacitor). It must be noted however that the battery voltage indicates the combined effect of the energy harvesting and the energy consumption. Unless all loads are turned off, the actual amount of energy being harvested cannot be known. Even if the loads could be turned off, the buffer voltage level usually rises very slowly in response to changes in energy level and could therefore only offer energy data of poor resolution.

Physical setup and testing

An expansion board for the TelosB was created incorporating the sensor and other circuit extensions (see Figure 5).

Basic system operation, as seen through the output of the peak detector (sensor) and the microphone, is plotted in Figure 6. In the graph, the sampling period started at 57.8 s and ended at 69 s. The overshoot in the peak detector output at the beginning of the sampling period can be clearly seen, as well as its correction, brought about by the corrective discharge. Sound, generated through human speech, is generated at 60.0 s. The sound lasted for almost a second, and the peak detector output can be seen rising to the peak level of the sound. The retainment of the peak detector of the sound's peak level can be seen, even after the cessation of sound generation. It must also be noted that the voltage droop exhibited by the system is larger than that computed in a previous section, suggesting that the leakage current, or I_{lk}, was underestimated in the computations.

The power consumption of the system (and the peak detector block alone), as it goes through the different

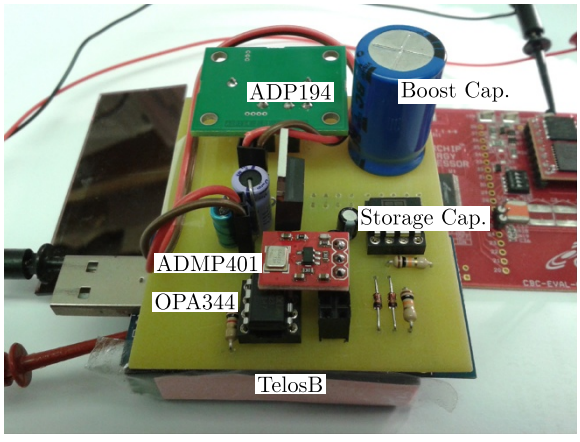

Figure 5 Expansion board, with the Enerchip and the solar panel in the background.

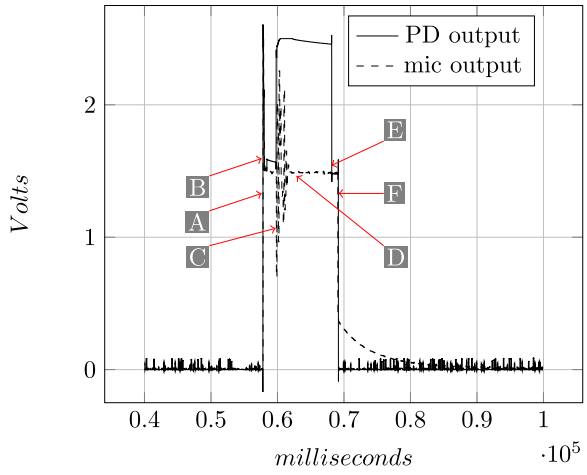

Figure 6 Sample operation. (A) beginning of active period. (B) Corrective discharge. (C) Sound is introduced. (D) Cessation of sound. (E) Sampling and discharging of storage capacitor. (F) End of active period.

phases of operation, is plotted in Figure 7. It can be seen that even when the peak detector block (which includes the MAX323 and the microphone) is not active, the system has a quiescent power consumption of 0.36 mW. When the peak detector block is active, power consumption rises, with a peak of 2.8 mW.

To test the system, the board was exposed to three different kinds of sound, of differing loudness levels. The three different kinds of sound utilized were 10,000 Hz tone, white noise, and pink noise. White noise is a random signal with flat or constant power spectral density, and pink noise, on the other hand, has a power density which is inversely proportional to the frequency. The sound was produced in 1-s pulses fired in succession with inter-pulse gaps of random lengths. The possible inter-pulse gap lengths however, were limited in such a way that at least a single pulse would be fired within any active period of the system. This is so that 'soundless' active periods would not skew the average of the readings. Random gap lengths were used so that we could test the effects of differing sound introduction points in an active period. The sound level was verified by a sound level meter (SLM) located very near to the mote's microphone. In between each active period, a packet is sent to the base station: the packet contains the reading made in the preceding active period. The mote was powered via energy harvesting in all of the experiments, with the solar panel illuminated by a lamp shining with a brightness of at least 1,300 lx. Before any of the experiments, the solar panel and the EH system were exposed to the light for 50+ min without any load to ensure that the batteries are fully charged once the experiment begins. Each experiment lasted for a duration which enabled the system to take 20 readings.

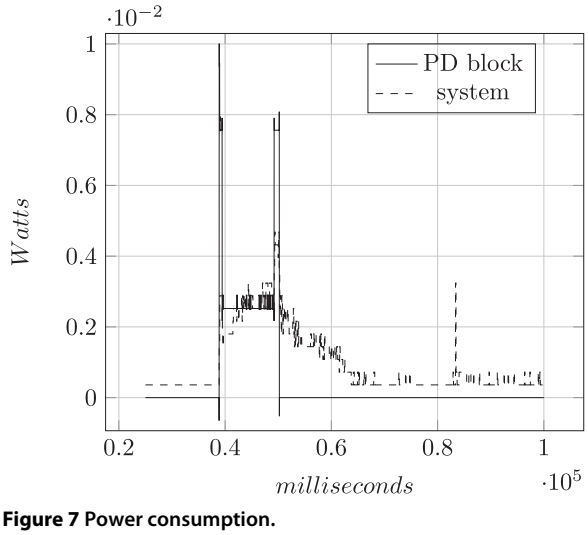

Figure 7 Power consumption.

The values produced by the system in the experiments are plotted in Figures 8, 9 and 10. It must be noted that Figure 8 has no data for 110 dB because of the limits imposed by our audio equipment. Figures 8, 9 and 10 show that the system could differentiate between different sound levels. The graphs are sigmoid in shape, with the tapering in higher decibel values indicating the microphone's limitation in differentiating very loud sounds (those greater than 100 dB). Likewise, the baseline value and the first bars on the left-hand side show that the microphone could not detect sound level intensities lower than 50-60 dB.

Using the data from the white noise experiment and exponential regression, we were able to find a function relating the sensor output to dB level. We removed the last data point (110 dB) since the resulting function no

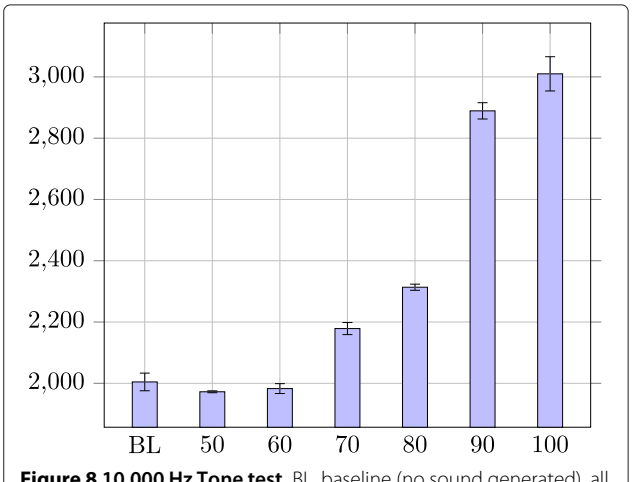

Figure 8 10,000 Hz Tone test. BL, baseline (no sound generated), all other numbers in the X-axis have dB as units.

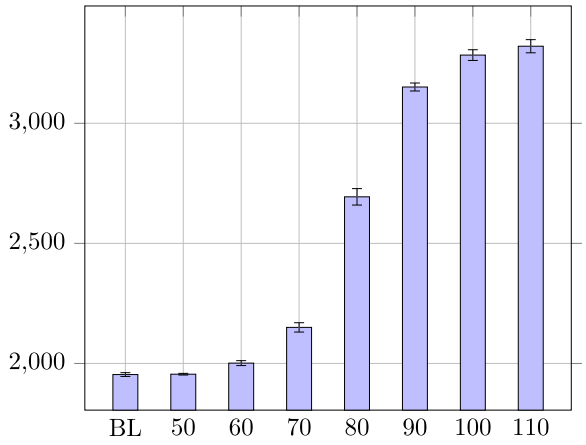

Figure 9 White noise test. BL, baseline (no sound generated), all other numbers in the X-axis have dB as units.

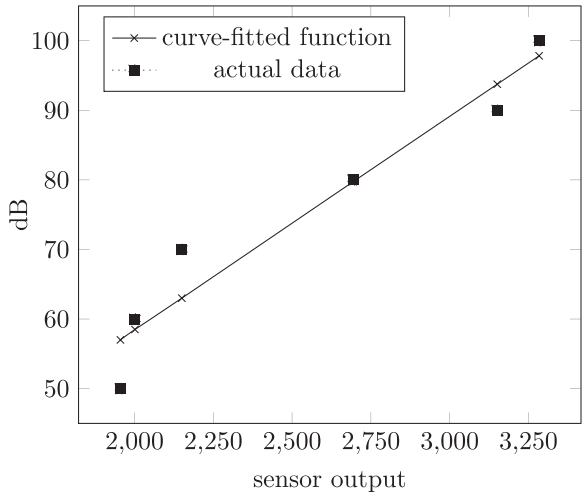

Figure 11 Function derived from curve-fitting the data points generated through experimentation. $f(x) = 0.518203434173811 \times 1.00708470990609^{x}$.

longer tracked the data points well. The resulting function is shown in Equation 14:

$$f(x) = 0.518203434173811 \times 1.00708470990609^{x} \quad (14)$$

The function, along with the experimentally-derived data points, is plotted in Figure 11. Below 70 dB, the function's maximum deviation from the actual value goes up to 7 dB. Beyond 70 dB, the maximum deviation goes down to 4 dB.

Theoretically, a table of finer resolution could be produced, which would enable a more accurate discernment of sound levels.

Methods for computing charge time in multiple-pulse load applications

For systems with only a single pulse component or only one component/function which consumes a large amount of current for an extended period of time, Equation 9 and Equation 10 would suffice in deriving the size of the capacitor needed. Equation 13 would then give the duty cycle of the system. For applications that have two or more components that have significant current draws (or what we call *multiple-pulse load applications*), however, Equation 9 and Equation 10 would not suffice.

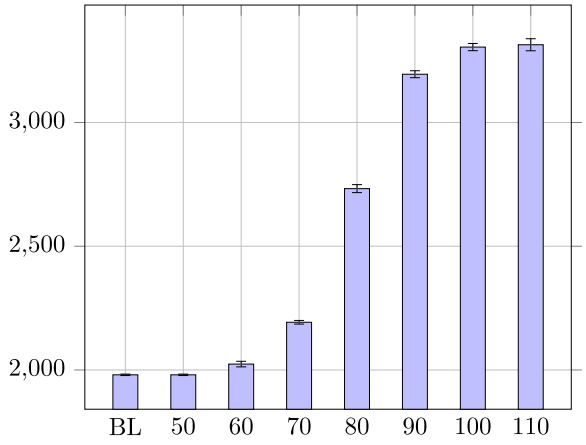

Figure 10 Pink noise test. BL, baseline (no sound generated), all other numbers in the X-axis have dB as units.

The system tested in the previous subsection acted like a single-pulse load application since it sent its packets to a base station that does not duty cycle. Thus, even when it is duty cycling its radio using LPL, since the destination does not duty cycle, the packet can be sent almost immediately at little cost to the sender. In situations however wherein the destination is also another mote powered by energy harvesting (duty cycling using LPL) the packet send operation would consume significantly more energy since the sender has to wait for the receiver to wake up, all the time keeping the radio on and drawing a high amount of current. In such a situation, the mote effectively becomes a dual-pulse load application, the two pulse loads being the sensor board and the radio. In the future, multiple pulse loads will become more and more common in wireless sensor network nodes as the sensors that are attached to motes grow, not just in number but also in sophistication (for example, [47]). These multiple-pulse load applications pose no problem for battery-powered systems since batteries could supply high levels of current for long periods of time. For energy harvesting-powered systems however, such multiple-pulse load applications would require a revision of Equation 9 and Equation 13 or the creation of new techniques that would ensure continuous system operation. In summary, the problem is this: *given multiple pulse loads, executed in succession, with possibly differing levels of current draws and draw length durations, what*

should be the size of the boost capacitor, and what should the intervals (charge times) be between the pulse loads?

In this section, we consider the case wherein there are two pulse loads in the system. Nevertheless, all of the methods could be easily and readily extended to accommodate three or more pulse loads.

References would be made to *smaller* and *larger* pulse loads. By this we mean with respect to the size of the boost capacitors computed for each pulse load using the method described earlier in the paper. *Pulse loads* will also be referred to simply as *loads*. Once again, since all capacitors involved in this subsection are boost capacitors, all references to 'capacitors' should be taken to mean 'boost capacitors'.

For all the methods described, the capacitor of the larger load is utilized (which easily answers the first part of the problem posed above). The larger capacitor is utilized because it will accommodate both current draws. The capacitor of the smaller draw could possibly prove insufficient for the larger draw, completely depleting it or dropping its voltage below the minimum level required for system operation.

Lazy method

In the simplest of the four methods, the computationally-derived charge time for the larger of the two loads (the one needing a larger capacitor) is simply used for both loads. The method is guaranteed to work: since the charge time is enough to recharge the capacitor after a large current draw, it would definitely suffice for the smaller current draw. The main advantage of the method is its simplicity, while its main disadvantage is the performance loss that comes with it. The charge time of the larger draw is usually larger than that of the smaller draw's original charge time - thus, the overall duty cycle of the system would be lower than what it would be if the two draws were simply concatenated. Since the smaller draw does not deplete the capacitor as much as the larger draw, it follows that the smaller draw could actually make do with a shorter charge time - something which would be taken advantage of by another method.

Concatenation method

In this method, the two or more loads are simply *concatenated*, or executed in sequence, each retaining its original, analytically-derived charge time. The reason this method can work is primarily because of the inherent conservativeness of the analytical method, and indeed, preliminary experiments have shown that this method can work for certain combinations of loads. The main advantage of this method is its simplicity: the computations necessary are exactly the same as that of the single-load system, only performed several more times. The disadvantage of this method is that it does not work for all load combinations: when the loads are vastly different in magnitude, the capacitor may not have enough time to be charged at a sufficiently high enough voltage level before the next load manifests itself.

Aggressive analytical method

In this method, the analytical method is optimized to reflect the fact that while there are now two different loads in the system, there is only one capacitor. The value of the heavy load charge time would remain the same - although it would now become the charge time *after* the execution of the heavy load and *before* the execution of the light load. This is because the capacitor's recovery time after a heavy load should remain the same as before - it is the heavy load's optimal capacitor size that is being used in the system after all. What will change and be newly computed by this method is the charge time *after* the light load and *before* the heavy load. To compute this:

1. Denote the t_{Draw} and R_{Load} of the *light* load as $t_{\text{DrawLight}}$ and $R_{\text{LoadLight}}$, respectively, and denote the optimal capacitor size for the heavy load as C, compute V_{MinLight} with

$$V_{\text{MinLight}} = \frac{V_{\text{Max}}}{e^{\frac{t_{\text{DrawLight}}}{R_{\text{LoadLight}} \times C}}} \tag{15}$$

   This is, in effect, the level at which the voltage across the capacitor will drop to when the light load makes its current draw.
2. The t_{Charge} for the light load (denoted as $t_{\text{ChargeLight}}$) could then be computed from Equation 2, using V_{MinLight} instead of V_{Min}.

Opportunistic method

All methods discussed so far focused on finding charge times that would enable multiple load applications to function in an energy harvesting setting. They have the advantage of ensuring predictability and uniformity in operation, as it is known exactly when loads would be executed, and at what intervals. There may be situations however, when predictability and regularity are not strictly required - one can, for instance, be more concerned about getting as many sensor readings as possible than getting such readings at perfectly defined intervals. In such situations, all of the previous methods are disadvantaged in their being conservative. Analytical methods are inherently conservative since they are designed to ensure system operation across many conditions. For instance, the rate at which power is supplied by the energy-harvesting system actually varies from one time to another: it is a function of the current light intensity applied to the solar panels as well as the storage system's state of charge. Nevertheless, when carrying out computations for the

analytical method, the value that must be assumed is the minimum supply rate, or at most, the average. Thus, there are moments wherein the system could actually support better performance (i.e., higher duty cycles). When predictability and regularity are not strictly required, such extra performance not being utilized could be considered as *wasted* performance.

To avoid such waste whenever possible, we propose a new method for charge time determination. Instead of utilizing predetermined charge times for a load, an *opportunistic* charge time is instead used: the system will execute a load whenever possible, subject to the charge level of the capacitor. For this, we rely on the MSP430's supply voltage supervisor (SVS), which also has counterparts in other microcontrollers. The SVS, when activated, continuously monitors the level of the supply voltage of the microcontroller, and sets the *SVSOP* bit of the 8-bit *SVSCTL* register to 1 whenever it goes below a user-predefined value. The enabling and disabling of the SVS, as well as the setting of the voltage threshold, are likewise done by setting certain bits in the *SVSCTL* register. To implement the method, the execution of the load must be gated or wrapped by another code which activates the SVS and checks the *SVSCTL* register. The pseudocode for such wrapping or gating, called *Gated_Exec*, is outlined in Algorithm 1.

Algorithm 1 Wrapping/gating code for Opportunistic Algorithm

```
1:  procedure GATED_EXEC (volt_limit, thresh_count)
2:      threshold_exceeded = False
3:      threshold_exceeded_counter = 0
4:      low_power_flag = 0
5:      while threshold_exceeded_counter < thresh_count do
6:          Delay (1 s)
7:          Enable SVS Monitoring, define low voltage as
            volt_limit
8:          Delay (10 ms)          ▷ Allow SVS monitor to settle
9:          From SVS register, set low_power_flag value
10:         Disable SVS Monitoring
11:         if low_power_flag ≠ 1 then
12:             threshold_exceeded = False
13:         else
14:             threshold_exceeded = True
15:         end if
16:         if threshold_exceeded is True then
17:             threshold_exceeded_counter =
                threshold_exceeded_counter + 1
18:             threshold_exceeded = False
19:         else
20:             threshold_exceeded_counter = 0
21:         end if
22:     end while
23:     Execute Load
24: end procedure
```

The code is parameterized by two values: *volt_limit* and *thresh_count*. Lines 5-22 comprise the main *while* loop of the code which prevents the execution of the load while conditions are not yet met, primary of which is the *threshold_exceeded_counter* reaching *thresh_count*. Line 6 imposes a 1-s delay between loop iterations. Line 7 enables the SVS and sets the voltage threshold (defined by the user through *volt_limit*). Line 8 gives time for the SVS to settle. In Line 9, the *SVSOP* bit of the *SVSCTL* register is checked, and its value copied to *low_power_flag*. Line 10 disables the SVS between loop iterations, to save power. Lines 11-15 check the value of the *low_power_flag*, effectively converting it to a boolean variable *thresh_exceeded*. A value of *True* for *thresh_exceeded* indicates that the supply voltage has exceeded that of the threshold. Lines 16-21 deal with determining the new value of *threshold_exceeded_counter*: it is incremented by 1 if the threshold is exceeded (*thresh_exceeded* is *True*) and reset to 0 otherwise. The purpose of the parameter *thresh_count* (to which *thresh_exceeded_counter* is compared) is to allow users to define a minimum amount of time by which the threshold voltage level must be exceeded *continuously* before considering the capacitor as sufficiently charged for load execution. This is important since the relationship between capacitor voltage and its state of charge is not linear - the rise of capacitor voltage slows down during the latter stages of the charging process. Thus, even if the threshold is set to a high value, it would not be prudent to immediately consider the capacitor as sufficiently charged just because its voltage was observed to surpass the said threshold - this is especially true for capacitors of significant size. Another use for this parameter is for rate control in routing protocols such as directed diffusion [48]. In directed diffusion, a node on the path to the sink may request upstream nodes to slow the generation of data or packets because it could no longer handle the packet volume. The execution of the actual load happens in Line 23, which will only be reached upon escaping the loop embodied in Lines 5-22.

Testing the methods for computing charge time in multiple-pulse load applications

To test the methods, we created a dual-pulse load application using LPL. The mote alternated between two kinds of sends, the first (Load 1) sends to a receiver with a wakeup interval of 0.26 s, and the second (Load 2) sends to a receiver whose wake-up interval was either 0.63, 1.02, or 1.42 s, making for a total of three setups. The sends were configured as broadcasts, so the node upon sending would attempt the send for as long as the amount of time specified by the receiver wake-up interval - thus, the length of the receiver wake-up interval is equivalent to the sender's current draw length for that load. In contrast, when doing acknowledgement-enabled unicasts, the

sender would stop the send upon receiving an acknowledgement from the receiver, to save energy. The capacitor sizes and charge times for the four loads (Load 1, Load 2A, Load 2B, and Load 2C) when each is considered in isolation are tabulated in Table 3.

To test that the setups are working, all the packets are received by the base station, which indicates packet reception through a terminal printout. The system was powered by energy harvesting during the experiments, with the same conditions as that in the previous experiment (same light intensity, pre-experiment exposure). In each setup, the capacitor utilized was that of the larger load: for example, in the Load 1-Load 2A setup, the capacitor used was 5,000 μF. Each experiment ran for a period sufficient enough so that 25 packets from each type of send was sent by the mote.

All methods were tested on all three setups and found to be functional. For the Opportunistic method, *thresh_count* was set to 1, and *volt_limit* was set to 3.35 V, for both loads. The application running on the mote was slightly modified for the experiment used to test the Opportunistic method - to enable us to keep track of the charge times, the length of the charge time preceding a send was sent within the packet as payload. The length of the charge time can be determined by the number of iterations of the loop embodied in Lines 5-22 of Algorithm 1. To ensure that the system is in continuous operation (not restarting), each packet also contained a sequence number that was regularly incremented after a send operation.

For the Load 1-Load 2B setup, Figure 12 plots the opportunistically determined charge times for both loads, along with the charge times determined by the Concatenation method.

It can be seen in Figure 12 that the opportunistically-derived charge times vary across time, indicative of the constantly changing charge rate of the energy-harvesting system. It must be noted that in the opportunistically-derived pair, the charge time for the heavy load is always smaller than that of the light load, in contrast with the analytically-derived pair. To understand this, one must remember our definition of charge time: it is the period of time before a current draw (or active period). A light load is preceded by the heavy load, which draws a significant amount of current from the capacitor. Thus, charging the capacitor to an acceptable voltage level would

Table 3 Load table

	Draw length (*s*)	Capacitor size (*uF*)	Charge time (*s*)
Load 1	0.26	2,000	29.18
Load 2A	0.63	5,000	70.7
Load 2B	1.02	8,000	114.5
Load 2C	1.42	11,000	157

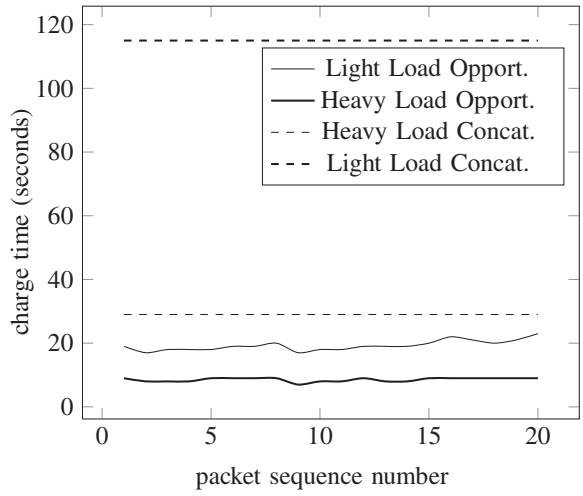

Figure 12 Charge times: Opportunistic method vs Concatenation method.

take time. By the same token, the heavy load is preceded by the light load, which relatively does not draw much current. Therefore, it may be more prudent to compare the opportunistically-derived heavy load charge time with the analytically-derived light load charge time, and vice versa. Even when such comparisons are made however, the opportunistically-derived values are still significantly less than that of the analytically-derived values. For instance, the average opportunistically-derived charge time for the light load is 19.25 s, and the analytically-derived charge time for the heavy load is 114.5 s.

The duty cycles associated with each method for the three setups are normalized against that of the Concatenation method and plotted in Figure 13 and Figure 14. Figure 13 plots the duty cycles associated with the three

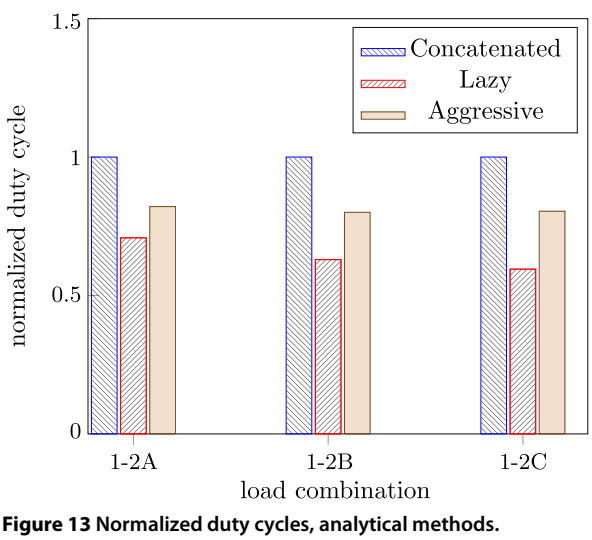

Figure 13 Normalized duty cycles, analytical methods.

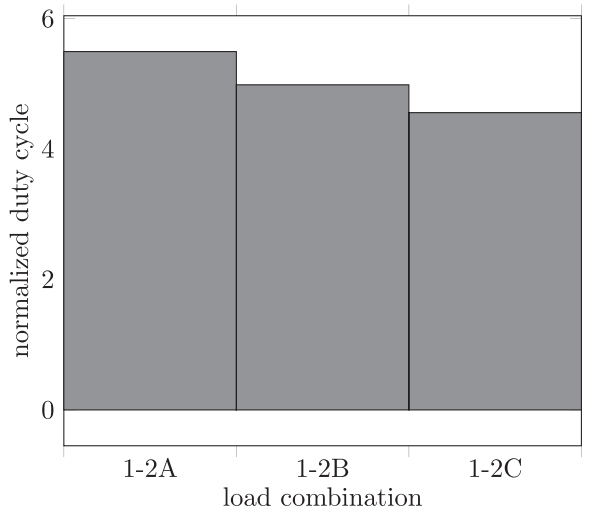

Figure 14 Normalized duty cycles, Opportunistic method.

analytical methods, while Figure 14 plots the average duty cycle measured when the Opportunistic method is used. The data for the Opportunistic method was plotted separately since its numbers are significantly larger than that of the other three.

Figure 13 shows the Lazy method and the Aggressive Analytical method being inferior to the Concatenation method. This is to be expected. As already mentioned, the Lazy method is guaranteed to work but has the drawback of performance loss, which grows proportionately with the disparity between the sizes of the light load and the heavy load. This too, is apparent when the results are compared across setups. The Aggressive Analytical method performed better than the Lazy method, but still worse than the Concatenation method. The Aggressive Analytical method is also guaranteed to work, but even with the other charge time tailor-fitted for the voltage drop caused by the light load, the value is still larger than that of its counterpart in the Concatenation method. The voltage level from which the charge period has to recover is higher (because the capacitor is larger), but the capacitor is now also more difficult to charge.

Figure 14 shows normalized data for the Opportunistic method. The Opportunistic method performed significantly better than all three other methods, with the lowest in the three setups showing a 350% improvement over the Concatenation method. These results signify that if the requirements of perfect data regularity and predictability could be relaxed, significantly more performance could be had from the system.

Related work

Noise measurement using WSNs has been attempted four times before: firstly, in the work presented in [17], [49], and [4]; secondly, in the work presented in [5] and [50]; thirdly, the work presented in [18] and [6], and most recently, our own work in [20].

Compared to the research presented in this paper, the first and second sets of studies are more complete end-to-end solutions for noise monitoring: they employed A-weighting and have customized networking protocols (presented in [49,50], respectively). The third study is notable for its use of fuzzy logic in inferring subjective noise annoyance from the sensor readings. Tan and Jarvis, 2013 [20], on the other hand, featured the first noise-sensing WSN motes to be powered by energy harvesting.

Table 4 summarizes the main differences between these previous studies.

This work is, in a sense, a direct follow-up to [20]. Tan and Jarvis, 2013 [20] discussed the difficulties of noise sensing using low-power WSN motes, difficulties that are compounded further by having the motes powered by energy harvesting. These difficulties include the limited storage space available for the microphone voltage readings, and the limited word width of the system. Both place limits on the length of time over which L_{eqT} could be taken. This work, in comparison, does away with L_{eqT} and uses the peak noise level detected within a time period instead. This design decision, brought about in part through discussions with domain experts at New York's Center of Urban Science and Progress (CUSP), leads to the minimization of the need for high-frequency ADC sampling, making limited storage space and word widths much less of an issue. Therefore, the continuous and uninterrupted period of time over which the mote could make its observation is much longer than in [20]. While no noise codes currently employ peak noise levels as their metric, it is still very deployable in many situations and would complement other noise-sensing systems that employ L_{eqT} as a measure. Moreover, our design being powered by energy harvesting makes it a good fit for low-maintenance urban sensing systems.

Just recently, Libelium released a Smart Cities sensor board [47] for the Waspmote [53]. The Smart Cities sensor board includes a linear displacement sensor for detecting cracks in building and infrastructures, a particle sensor for measuring air-suspended pollutants and a microphone for A-level weighted sound loudness. The Waspmote by default is battery-powered, although it does have provisions for being powered by energy harvesting through solar panels. To the best of our knowledge, there are no known reports or studies detailing the performance of the Waspmote and the Smart Cities sensor board under energy harvesting conditions. A detailed comparison of the energy-harvesting Waspmote with the Smart Cities sensor board and an energy-harvesting TelosB with a custom sensor board is part of our future work.

Table 4 Comparison of previous studies

	Reported error range/accuracy	Notes	Platform	Year
[17], [49], and [4]	± 2 dB	First work on noise sensing; utilized dedicated hardware for noise sensing **and** metric computation	Tmote Sky/ TelosB [19]	2008
[5] and [50]	± 2 dB		CiNet [51]	2010
[18] and [6]	± 3 dB	Used fuzzy logic-based method for inferring subjective noise annoyance	Sun SPOT [52]	2012
[20]	Unreported	Energy harvesting	TelosB [19]	2013
This work	± 7 dB	Energy harvesting, based on peak detection	TelosB [19]	2014

Even without the detailed technical study however, one aspect of the competing systems could already be compared: cost. The Waspmote is a ready-for-deployment system. The cost of the Waspmote and Smart Cities sensor board combination (*sans* energy-harvesting component) is 270 Euros. In comparison, the TelosB is a prototype/research platform costing 77 Euros. The energy harvesting system that we have utilized is part of an evaluation board package, but a cursory check of individual component costs indicate that the total cost of components for the energy-harvesting system does not exceed 30 Euros. The cost of the customized sensor board comes in at under 20 Euros. All in all, the cost of a fully customized TelosB-based system (in a single board) is less than half the cost of an equivalent Waspmote-based system, even if only a modest number of units are produced.

Future work

Planned future work for our mote extension design includes trying out new microphones to extend the range of sound intensities which the system could discern. In due course, we also intend to incorporate A-weighting to our design, either via a software implementation or a separate circuit.

Of primary concern in future design iterations is the voltage droop, which computations underestimated for the current design. Two strategies could be adopted: compensation and reduction.

We can opt to compensate for the voltage droop by adjusting the digital sampling strategy. In this work, a digital sample is taken at the end of the peak detector's active period. To compensate for the voltage droop, we could take several digital readings within an active period and just keep the highest among them. The main disadvantage that comes with this approach is the increased storage space utilization and operational complexity.

We can also aim for the outright reduction of voltage droop. This could be done by a better circuit board layout, or by using better op-amps. The storage capacitor size could also be increased (Equation 8), but in doing so, the range of frequencies detectable by the system would then decrease (Equation 6).

In this work, we also optimized the charge time of a cycle by monitoring the voltage of the boost capacitor (which is indicative of the capacitor's charge level) instead of relying on a precomputed value. A cycle has two parts: the active part (the current draw time) and the inactive part (the charge time), and our work optimized the latter. In the same way that the analytical method is very conservative in its charge time computation, and it is also very conservative in its capacitor size computation. Our experiments showed that the capacitor computed for a certain length and level of current draw could actually support a much longer current draw than what it was originally intended for. We presented methods for deriving the actual maximum current draw length that a capacitor could support in two previous studies [54,55]. While the value could be derived manually via experimentation, what is unique about the techniques presented in [54,55] is that they allow the nodes to derive the values themselves, without human intervention, even while in deployment. This allows nodes to tailor and optimize their operations in response to conditions that affect the capacitor, such as capacitor ageing. While the techniques presented in [54,55] were done in the context of a very specific type of pulse load (radio communication using LPL), generalizing the techniques is certainly possible. Part of our future work then is combining the work done in this paper with generalized versions of the techniques presented in [54,55]. The former would optimize the inactive part (charge time) of a cycle for a given charging condition, while the latter would optimize the active part (draw time) for a given hardware configuration (i.e., capacitor size). It will be interesting to see how much performance gain such a combined technique would yield, compared to the analytical method.

Conclusion

WSNs could potentially help in mitigating and preventing noise pollution which is increasingly becoming a concern

in many cities all over the globe. Adding EH capability to WSNs could provide a low-maintenance, low-cost system to city administrators. Unfortunately, the metrics on which many currently existing noise codes are based are a poor fit to current EH-WSNs because of their limited capabilities. In this paper, we presented a system which bases its output on peak noise levels, instead of L_{eqT}. While 'non-compliant' with existing noise codes, our design could potentially perform better than compliant EH-WSNs, at least when it comes to the length of continuous observation time possible. It also provides a wider range of options which could be taken advantage of by power management algorithms, leading to better energy utilization. The significant potential of such a design (and the inspiration for it) came to us partly through discussions with city agencies in New York and London.

We also presented and discussed several methods for computing charge times for energy-harvesting nodes that run multiple-pulse load applications. We proposed and tested an opportunistic method for use in cases wherein the requirement of predictability and regularity in data generation could be relaxed. Our method has been shown to exhibit 350%-400% improvement over analytical methods, when it comes to the generated duty cycle. While the Opportunistic method was first presented in the context of multiple-pulse load applications, it could readily be applied to single-pulse load applications.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

WMT designed and implemented the mote extension board, designed and implemented the algorithms for charge time determination, and wrote the manuscript; SAJ provided input at each stage of the project (including the analysis of the data), and edited the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work is funded in part by the *UK Technology Strategy Board (TSB) Emerging Technologies Programme*, Project 131187/26835-183208, *OPV-based Energy Harvesting for Urban Noise Pollution*. Author W. M. Tan is supported by the Republic of the Philippines' *Engineering Research and Development for Technology (ERDT) Program*. We are grateful to colleagues at New York's *Center for Urban Science and Progress (CUSP)* for their input into this research. We are also grateful to *AVNET Abacus* for the gift of the Cymbet EVAL-09 evaluation boards.

Received: 17 April 2014 Accepted: 30 September 2014
Published: 14 October 2014

References

1. BBC, Copenhagen: Noise pollution ruling in Denmark (2014). http://www.bbc.co.uk/news/world-europe-26404666. Accessed 7 July 2014
2. European Commission, in *Official Journal of the European Communities,*. Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the Assessment and Management of Environmental Noise (Publications Office of the European Union, Luxembourg, 2002)
3. New York City Government, New York City Noise Code (Local Law 113 of 2005), (2013). http://www.nyc.gov/html/dep/html/noise/index.shtml. Accessed 7 July 2014
4. S Santini, A Vitaletti, in *Proceedings of the 6th GI/ITG KuVS Fachgespraech Drahtlose Sensornetze (Expert Talk on Wireless Sensor Networks). FGSN '07*. Wireless Sensor Networks for Environmental Noise Monitoring (GI/ITG, Aachen, Germany, 2007), pp. 98–101
5. I Hakala, I Kivela, J Ihalainen, J Luomala, C Gao, in *Proceedings of the First International Conference on Sensor Device Technologies and Applications. SENSORDEVICES '10*. Design of low-cost noise measurement sensor network: sensor function design (IARIA, Wilmington, Delaware, USA, 2010), pp. 172–179
6. JA Mariscal-Ramirez, JA Fernandez-Prieto, MA Gadeo-Martos, J Canada-Bago, in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications. ISDA '11*. Knowledge-based wireless sensors using sound pressure level for noise pollution monitoring (IEEE Press, Piscataway, NJ, USA, 2011), pp. 1032–1037
7. D Musiani, K Lin, TS Rosing, in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks. IPSN '07*. Active sensing platform for wireless structural health monitoring (ACM, New York, NY, USA, 2007), pp. 390–399
8. L Selavo, A Wood, Q Cao, T Sookoor, H Liu, A Srinivasan, Y Wu, W Kang, J Stankovic, D Young, J Porter, in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems. SenSys '07*. Luster: wireless sensor network for environmental research (ACM, New York, NY, USA, 2007), pp. 103–116
9. I Stark, in *Proceedings of the 2nd Conference on Wireless Health. WH '11*. Converting body heat into reliable energy for powering physiological wireless sensors (ACM, New York, NY, USA, 2011), pp. 31–1312
10. Centre for Advanced Spatial Analysis (CASA), University College London, CityDashBoard: London (2014). http://citydashboard.org/london/. Accessed 7 July 2014
11. Cymbet Corporation, AN-1025: Using the EnerChip in Pulse Current Applications (2013). http://www.cymbet.com/. Accessed 7 July 2014
12. M Moser, *Engineering Acoustics*. (Springer, London, 2009)
13. D Bies, C Hansen, *Engineering Acoustics*. (Spon Press, London, 2003)
14. Analog Devices, AN-1112: Microphone Specifications Explained (2013). http://www.analog.com/. Accessed 7 July 2014
15. H Fletcher, WA Munson, Loudness, its definition, measurement and calculation. J. Acoust. Soc. Am. **5**(2), 82–108 (1933)
16. Casella, An Introduction to Industrial Noise Measurements and Hearing Issues in the Workplace, (2013). http://www.analog.com/. Accessed 7 July 2014
17. S Santini, B Ostermaier, A Vitaletti, in *Proceedings of the Workshop on Real-world Wireless Sensor Networks. REALWSN '08*. First experiences using wireless sensor networks for noise pollution monitoring (ACM, New York, NY, USA, 2008), pp. 61–65
18. JC Cuevas-Martinez, J Canada-Bago, JA Fernandez-Prieto, MA Gadeo-Martos, Knowledge-based duty cycle estimation in wireless sensor networks Application for sound pressure monitoring. Appl. Soft Comput. **13**(2), 967–980 (2013)
19. J Polastre, R Szewczyk, D Culler, in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks. IPSN '05,*. Telos: enabling ultra-low power wireless research (IEEE Press, Piscataway, NJ, USA, 2005)
20. WM Tan, SA Jarvis, in *Proceedings of the 2013 IEEE Conference On Wireless Sensor (ICWISE). ICWISE '13,*. Energy harvesting noise pollution sensing wsn mote: Survey of capabilities and limitations (IEEE Press, Piscataway, NJ, USA, 2013), pp. 53–56
21. Texas Instruments, MSP430x1xx Family User's Guide (2013). http://www.ti.com/. Accessed 7 July 2014
22. Texas Instruments, MSP430x12x Mixed Signal Microcontroller Datasheet (2013). http://www.ti.com/. Accessed 7 July 2014
23. Texas Instruments, CC2420: 2.4 GHz IEEE 802.15.4 /ZigBee-ready RF Transceiver Datasheet (2013). http://www.ti.com/. Accessed 7 July 2014
24. Advanticsys, Advanticsys Homepage (2013). http://www.advanticsys.com/. Accessed 7 July 2014
25. J Hill, R Szewczyk, A Woo, S Hollar, D Culler, K Pister, System architecture directions for networked sensors. SIGPLAN Not. **35**(11), 93–104 (2000)
26. TinyOS, TinyOS Homepage (2013). http://www.tinyos.net/. Accessed 7 July 2014
27. Cymbet Corporation, CBC-EVAL-09: EnerChip EP Universal Energy Harvester Eval Kit (2013). http://www.cymbet.com/. Accessed 7 July 2014
28. CBC915: EnerChip EP Energy Processor, (2013). http://www.cymbet.com/. Accessed 7 July 2014

29. Cymbet Corporation, EnerChip CC CBC3105 (2013). http://www.cymbet.com/. Accessed 7 July 2014
30. Analog Devices, ADMP401: Omnidirectional Microphone with Bottom Port and Analog Output (2013). http://www.analog.com/. Accessed 7 July 2014
31. Analog Devices, MS-2348: Low Self Noise: The First Step to High Performance MEMS Microphone Applications (2013). http://www.analog.com/. Accessed 7 July 2014
32. S Franco, *Design with Operational Amplifiers and Analog Integrated Circuits*. (McGraw-Hill, Singapore, 1988)
33. R Coughlin, F Driscoll, *Operational Amplifiers and Linear Integrated Circuits*. (Prentice Hall, London, 1987)
34. W Stanley, *Operational Amplifiers with Linear Integrated Circuits*. (Merrill, New York, 1994)
35. Texas Instruments, Low Power, Single-Supply, Rail-to-rail Operational Amplifiers: MicroAmplifier Series (2013). http://www.ti.com/. Accessed 7 July 2014
36. Analog Devices, ADP194: Logic Controlled, High-Side Power Switch (2013). http://www.analog.com/. Accessed 7 July 2014
37. Maxim, MAX323/MAX324/MAX325: Precision, Single-Supply, SPST Analog Switches (2013). http://www.maxim-ic.com/. Accessed 7 July 2014
38. TinyOS Core Working Group, TinyOS TEP 105: Low Power Listening (2007). http://www.tinyos.net/tinyos-2.x/doc/html/tep105.html. Accessed 7 July 2014
39. C Park, PH Chou, in *Proceedings of the 2004 International Symposium on Low Power Electronics and Design. ISLPED '04*. Power utility maximization for multiple-supply systems by a load-matching switch (IEEE Press, Piscataway, NJ, USA, 2004), pp. 168–173
40. A Kansal, J Hsu, S Zahedi, MB Srivastava, Power management in energy harvesting sensor networks. ACM Trans. Embed. Comput. Syst. **6**(4), Article 32 (2007)
41. A Sinha, A Chandrakasan, Dynamic power management in wireless sensor networks. Des. Test Comput. IEEE. **18**(2), 62–74 (2001)
42. C Alippi, G Anastasi, M Di Francesco, M Roveri, Energy management in wireless sensor networks with energy-hungry sensors. Instrum. Meas. Mag. IEEE. **12**(2), 16–23 (2009)
43. C Rusu, R Melhem, D Mosse, in *Proceedings of the 15th Euromicro Conference on Real-Time Systems*. Multiversion scheduling in rechargeable energy-aware real-time systems (IEEE Press, Piscataway, NJ, USA, 2003), pp. 95–104
44. C Moser, D Brunelli, L Thiele, L Benini, in *Proceedings of the 18th Euromicro Conference on Real-Time Systems*. Real-time scheduling with regenerative energy (IEEE Press, Piscataway, NJ, USA, 2006), pp. 10–27
45. F Ingelrest, G Barrenetxea, G Schaefer, M Vetterli, O Couach, M Parlange, Sensorscope: Application-specific sensor network for environmental monitoring. ACM Trans. Sen. Netw. **6**(2), 17–11732 (2010)
46. A Kansal, D Potter, MB Srivastava, Performance aware tasking for environmentally powered sensor networks. SIGMETRICS Perform. Eval. Rev. **32**(1), 223–234 (2004)
47. Libelium, Smart Cities Technical Guide (2013). http://www.libelium.com/. Accessed 7 July 2014
48. C Intanagonwiwat, R Govindan, D Estrin, J Heidemann, F Silva, Directed diffusion for wireless sensor networking. Networking, IEEE/ACM Transactions on. **11**(1), 2–16 (2003)
49. L Filipponi, S Santini, A Vitaletti, in *Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems. DCOSS '08*. Data collection in wireless sensor networks for noise pollution monitoring (Springer, Berlin, Heidelberg, 2008), pp. 492–497
50. I Hakala, I Kivela, J Ihalainen, J Luomala, C Gao, in *Proceedings of the The Fifth International Conference on Sensor Technologies and Applications. SENSORCOMM '11*. Design of noise measurement sensor network: Networking and communication part (IARIA, Wilmington, Delaware, USA, 2011), pp. 280–287
51. I Hakala, M Tikkakoski, I Kivela, in *Proceedings of the The Second International Conference on Sensor Technologies and Applications. SENSORCOMM '08*. Wireless sensor network in environmental monitoring - case foxhouse (IARIA, Wilmington, Delaware, USA, 2008), pp. 202–208
52. Oracle Labs, Sun SPOT World (2013). http://www.sunspotworld.com/. Accessed 7 July 2014
53. Libelium, Waspmote Datasheet (2013). http://www.libelium.com/. Accessed 7 July 2014
54. WM Tan, SA Jarvis, in *Wireless Days (WD) 2013 IFIP*. Self-determination of maximum supportable receiver wakeup intervals in energy harvesting wsn nodes (IEEE Press, Piscataway, NJ, USA, 2013), pp. 1–7
55. WM Tan, SA Jarvis, in *Proceedings of the 2013 IEEE Conference On Wireless Sensor (ICWISE). ICWISE '13*. Determination of maximum supportable receiver wakeup intervals in energy harvesting wsn nodes using a client-server setup (IEEE Press, Piscataway, NJ, USA, 2013), pp. 61–67

doi:10.1186/1687-1499-2014-167
Cite this article as: Tan and Jarvis: **On the design of an energy-harvesting noise-sensing WSN mote.** *EURASIP Journal on Wireless Communications and Networking* 2014 **2014**:167.