Lucid Dreaming: Reliable Analog Event Detection for Energy-Constrained Applications

Sasha Jevtic[†]   Mathew Kotowsky[‡]   Robert P. Dick[†]   Peter A. Dinda[†]   Charles Dowding[⋆]

sjevtic@eecs.northwestern.edu, {kotowsky, dickrp, pdinda, c-dowding}@northwestern.edu

[†]EECS Dept.	[‡]Infrastructure Technology Inst.	[⋆]Civil & Environmental Engg.
Northwestern University	Northwestern University	Northwestern University

ABSTRACT

Existing sensor network architectures are based on the assumption that data will be polled. Therefore, they are not adequate for long-term battery-powered use in applications that must sense or react to events that occur at unpredictable times. In response, and motivated by a structural autonomous crack monitoring (ACM) application from civil engineering that requires bursts of high resolution sampling in response to aperiodic vibrations in buildings and bridges, we have designed, implemented, and evaluated *lucid dreaming*, a hardware–software technique to dramatically decrease sensor node power consumption in this and other event-driven sensing applications.

This work makes the following main contributions: (1) we have identified the key mismatches between existing, polling-based, sensor network architectures and event-driven applications; (2) we have proposed a hardware–software technique to permit the power-efficient use of sensor networks in event-driven applications; (3) we have analytically characterized the situations in which the proposed technique is appropriate; and (4) we have designed, implemented, and tested a hardware-software solution for standard Crossbow motes that embodies the proposed technique. In the building and bridge structural integrity monitoring application, the proposed technique achieves 1/245 the power consumption of existing sensor network architectures, thereby dramatically increasing battery lifespan or permitting operation based on energy scavenging. We believe that the proposed technique will yield similar benefits in a wide range of applications. Printed circuit board specification files permitting reproduction of the current implementation are available for free use in research and education.

Categories and Subject Descriptors: B.8 [Hardware]: Performance and Reliability; C.3 [Computer Systems Organization]: Special-Purpose and Application-Based Systems; J.2 [Computer Applications]: Physical Sciences and Engineering.

General Terms: Design, Experimentation, Management, Measurement, Performance, Reliability.

Keywords: Sensor networks, power consumption, event detection, sensing.

1. INTRODUCTION

Wireless sensor networks have the potential to serve as platforms for a wide range of environmental monitoring applications. Applications can be considered at many levels, from the individual sensors, to the individual node hardware and software, to the local wireless network formed by nodes, and finally to that network's interaction with the broader world. Our work focuses on interaction among sensors, microcontrollers, and software within individual wireless sensor network nodes.

In this context, two universal research problems come to the fore: the maintenance problem and the unpredictable event problem. How can we arrange for nodes to operate without frequent intervention? Low maintenance is necessary to allow large-scale deployments in remote environments. It is prevented by short battery life, hence we focus on increasing battery life. How can we arrange for nodes to react to environmental events that occur at unpredictable times? We cannot assume that interesting data will be presented on a silver platter whenever requested. Jointly addressing the maintenance and unpredictable event problems requires changes to the conventional sensor network node architecture, allowing response to events at any time while maintaining ultra-low power consumption. We claim that addressing the problem requires a combined hardware and software approach. As described in Sections 2 and 5, attempts to solve these problems with software, alone, have resulted in high power consumption or missed events.

This work is motivated by applications that have the following characteristics:

1. They are extremely power-sensitive. The nodes are powered by batteries that can be replaced only after months or years of operation.

2. Low-power sensors and computational elements can be used for detection of events but not necessarily for recording detailed measurements of them.

3. Events are rare and the computation and/or communication they trigger is short relative to the event interarrival time.

4. Event interarrival times are unpredictable.

5. It is preferable not to miss, or ignore, events.

Section 3 describes the specific motivating application we target. In that application, events are structural vibrations. They cause a sensor voltage to exceed a threshold, resulting in a burst of high-resolution data logging.

Communication is not a significant power sink for our exemplar application, or other related applications, because sensor data logs and events need not be aggregated in real-time. Thus, queuing collected data on the node and sending batch transmissions allows the radio to be powered down most of the time. Modern ad-hoc sensor network protocols [3, 4] can similarly keep radio transmitter and receiver off most of the time.

Surprisingly, given that such applications are legion, existing and proposed sensor network node hardware and software do not adequately support them. The power consumption of the microcontroller and primary sensor are considerable for the following reasons:

1. Event detection is done in software via a sleep-read-test-jump polling loop. Polling requires that the primary sensor, analog-to-digital converter (ADC), and microcontroller remain in active states resulting in high power consumption.

2. Event arrival times cannot be accurately predicted and events should not be lost. Therefore, the amount of time spent in the sleep state, whether deterministic or random, must be small.

We describe the design, implementation, and evaluation of *lucid dreaming*, a hardware/software technique permitting long battery lifespans in applications requiring the detection of unpredictable events. Specifically, lucid dreaming eliminates the need for the primary sensor, ADC, and microcontroller to remain continuously active. The key idea is that event detection can be done in *analog hardware* much more efficiently than as code running on the microprocessor. Our analog hardware, *Shake 'n Wake*, wakes up a standard Crossbow mote [23, 18, 9] by raising a hardware interrupt. The interrupt handler in turn causes high resolution sampling to occur.

In our exemplar application, event detection is straightforward: an event interrupt is generated when the sensor's voltage level exceeds a sensor and application-specific threshold. Of course, this is a quite broadly useful event generation function for many applications. However, as described in Section 6, we believe that lucid dreaming can also be generalized to more complex event generation functions.

2. RELATED WORK AND CONTRIBUTIONS

A number of researchers have considered designing hardware, communication or power control protocols [24, 30,

16], multi-channel paging [2], and power management algorithms [28] to increase battery lifespans in wireless sensor networks. Work on low-power communication is largely orthogonal to the idea described in this article, and can be used in combination with it.

The architectural visions of Hill et al. [14] as well as Polastre, Szewczyk, and Culler [22] have had great impact on research and design of sensor networks. As described by Raghunathan et al. in their excellent survey [25], energy consumption is a major concern in most sensor network research. However, most previous research on low-power sensing architectures focuses on *periodic* sensing applications in which sensor network nodes may safely enter low-power modes at times of their choosing with the knowledge that data of interest will be available whenever they choose to wake up. Although periodic sensing is appropriate for some applications, many applications require the ability to reliably sense and/or react to events that occur at *unpredictable* times, e.g., the structural integrity monitoring application described in Section 3. Previous research on such event-driven applications [17, 19, 29] has relied on existing sensor network architectures. However, this has proven to be a poor fit, leading to high power consumption that results in battery lifespans on the order of hours or days instead of months or years.

Researchers have attempted to use sophisticated event prediction algorithms to improve the power consumption of existing sensor network architectures when used in event-driven applications [28]. However, without perfect prediction accuracy, such techniques must necessarily miss critical events or waste battery energy. Furthermore, the predictability of events is largely domain-dependent and evaluating it is often a goal of the application research using the sensor network. For many applications, including the one described in Section 3, events are too unpredictable for such methods to be feasible.

Researchers have previously used low-power notification techniques to reduce the amount of time during which high-power hardware must remain active. For example, Agarwal, Schurgers, and Gupta propose the use of low-power Bluetooth radios to activate high-power 802.11b radios [2]. Most closely related to our work is that of Schott et al. [27] and Dutta et al. [12]. Schott et al. describe their modular heterogeneous distributed sensing architecture in which each module may modify its state, and therefore power consumption, in response to local events and mission [27]. The scope and heterogeneity of their architecture is impressive, encompassing low-power microcontroller based nodes, 32-bit embedded microprocessors, and field-programmable gate arrays. However, this work relies on a wake-up timer to control exiting the lowest-power state. Therefore, if ultra-low-power operation is required, the technique is best suited to periodic sampling or sensing of events that occur at predictable times. Our proposed technique might be used to complement and enhance their power control infrastructure.

Dutta et al. have carefully considered minimizing power consumption in event-driven applications, identified the difficulty of detecting rare, random, and ephemeral events using existing sensor network architectures, and proposed a new architecture that uses duty cycling and wakeup circuits to reduce power consumption [12]. Duty cycling sensors to reduce power consumption must necessarily increase the probability of missing random events. This problem is al-

leviated, to some degree, by allowing sensors to wake up other nearby sensors in response to events. Although this idea is applicable in dense sensor deployments for detecting vehicles and soldiers (the intended application of Dutta et al.), it cannot be used in cases where the events of interest are truly ephemeral, i.e., they last for only a moment and do not imply that other events will, with high probability, be observed in the neighborhood of the previous event, as is the case for our motivating structural integrity monitoring application. Dutta et al. also describe the properties of a number of wake-up circuits. Unfortunately, all the sensors and wake-up circuits described have disturbingly high power consumption, i.e., from 880 μW to 19,400 μW. We point out the difficulties Dutta et al. faced only to make clear the importance and difficulty of the low-power event-driven sensing problem.

Our work makes the following main contributions:

1. We identify the primary mismatches between existing sensor network architectures and event-driven applications;

2. We propose a hardware–software technique to permit the power-efficient use of sensor networks in event-driven applications;

3. We have analytically characterized the situations in which the proposed technique is appropriate; and

4. We have designed, implemented, and tested a hardware-software solution for standard Crossbow motes that embodies the proposed technique.

The average power consumption of our sensor and wakeup circuit is 15 μW, which is more than two orders of magnitude lower than the best previously reported. In the building and bridge structural integrity monitoring application, the proposed technique achieves 1/245 the power consumption required by existing sensor network architectures, thereby increasing battery lifespan to the shelf life of the batteries or permitting operation based on energy scavenging [20, 26]. We believe that the proposed technique will yield similar benefits in a wide range of applications. Printed circuit board specification files permitting reproduction of the current implementation for free use in research and education are available from the authors.

3. MOTIVATION

Shake 'n Wake was motivated by our discussions with a civil engineering group that is deploying sensor networks based on Crossbow mote technology. It was clear that existing sensor network architectures were inadequate for their fairly typical structural integrity monitoring application. We believed that a sensor network node architecture addressing their specific needs would be useful in a broad class of event-driven sensing applications.

The objective of the Autonomous Crack Monitoring (ACM) project [11, 10, 6] is Internet-enabled remote monitoring of cracks in, or deformations of, structures to provide timely information about the health of critical infrastructure components such as bridges and buildings. Time-series data collected from sensors can be analyzed to identify trends and automatically alert engineers and/or regulatory authorities of impending problems. The ACM group's original system [10] is being deployed to compare environmental (long-term) and blast-induced (dynamic) crack width changes in residential structures, and has lead to a new approach to monitoring and controlling construction vibrations. It is a wired system that requires constant power and significant maintenance.

The ACM group is working to replace the existing wired system with a wireless sensor network [15, 21, 11]. Their goal is to support a year of reliable, unattended operation powered only by the two AA batteries in each of the wireless nodes. The work on this application recently won third place honors in the 2005 Crossbow Smart Dust Challenge [15].

At its core, crack monitoring is a *trigger-log-push* application. High resolution data are needed when the crack is in motion. Crack motion events occur at unpredictable times. Hence, we want to *trigger* when crack motion begins, *log* at the limits of the sampling resolution available until motion subsides, and later *push* the log to an analysis center.

This kind of application fits poorly to existing sensor network node technology, such as the Crossbow motes the ACM group is using, and to future node technologies of which we are aware. In the ACM application, logging must be done at high resolution. This results in high power consumption. However, we are only concerned with the logs for a relatively short duration after an event, i.e., the onset of crack motion, occurs. Current node hardware provides a wakeup timer, but this does nothing to improve the situation because the delay until the next event is not predictable. This leaves the designer with two unsatisfactory choices: sample at a high rate all the time, resulting in inadequate battery lifetimes, or use the wakeup timer to implement some sampling schedule, which will result in undetected events. Neither choice is acceptable for large-scale critical infrastructure monitoring.

The ACM application uses a string potentiometer and a geophone [7, 8], which is illustrated in Figure 1. Geophones are un-powered devices that produce output voltages. When used to monitor a crack, motion induces voltage fluctuation. In the default ACM configuration, the string potentiometer is attached to an ADC input on the mote and the application detects the onset of crack motion by continually sampling the ADC and comparing the sampled value to a threshold. It is the effect of this polling loop that we have moved from software running on the ATMega128 microcontroller and ADC to the custom hardware of the Shake 'n Wake board.

4. TECHNICAL DESCRIPTION

Lucid dreaming is a hardware/software technique for reducing power consumption in sensor network nodes that react to events detected via, potentially straightforward, computations on values measured using sensors. The proposed technique has relatively few requirements, and is viable in a large number of applications. Moreover, the technique may be used with platforms in addition to the MICA2 and MICAz, although doing this would require a slightly different printed circuit board design.

Figure 2 provides a high-level overview of lucid dreaming as used in our motivating application. The technique has two main components:

- **Hardware:** Custom analog hardware observes the sensor, detects events based on these observations, and notifies the microcontroller when more sophisticated processing is required. In our example hardware, Shake 'n Wake, events are detected when the geophone out-

Figure 1: Geophone connected to Shake 'n Wake board mated to Crossbow mote.

put voltage exceeds a threshold. Other detection methods, e.g., low-power finite state machines, may be used in other applications. Although we use separate sensors for event detection and data logging, the primary sensor may also be used for event detection if its power consumption is sufficiently low. When an event occurs, the hardware raises an interrupt.

- **Software:** The sensor network node is placed in a low-power standby state whenever there is no sensing, data processing, or communication work to be done. The node can be activated either with a timer (for example, to drive communication), or when a sensor event occurs. In the low power state, the microcontroller is placed in power-down mode, from which it may only be awakened by a hardware interrupt or the watchdog timer. ADCs are powered down and communication interfaces are temporarily disabled. The microcontroller is halted until an external hardware interrupt occurs. In response to an event interrupt, the microcontroller resumes full-power normal operation, at which point it may activate its ADC and store a series of samples from the primary sensor.

We begin by describing the criteria under which the lucid dreaming technique can be applied. Next, we describe our hardware implementation. Finally, we describe the software side of our implementation.
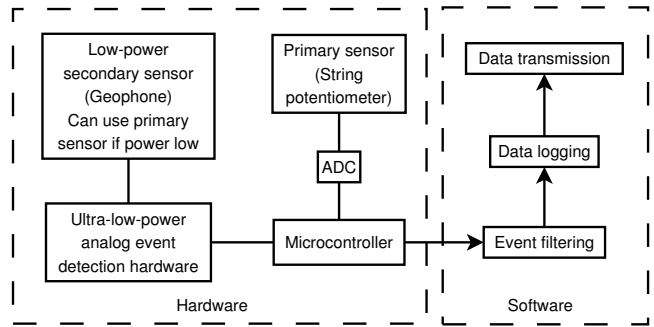

Figure 2: Lucid dreaming system overview.

4.1 Criteria for Viability

Lucid dreaming works exceptionally well for our motivating application. We also believe it will be applicable to a range of other event-driven sensor network applications of the kind we described in the introduction, resulting in power savings that depend on a number of application-specific parameters. However, several criteria must be met in order for the technique to be applicable. We now elaborate on these criteria.

- **Sensor and sensor support circuit power requirements must be modest.** Lucid dreaming requires that a sensor be continuously active which, in some cases, necessitates that the sensor be biased continuously. If support circuitry (such as a filter or amplifier) is required, it must also be continuously powered. The power consumption of our technique when no event is occurring is the sum of the power consumptions of the sleeping microcontroller, the wakeup circuitry, the sensor, and their associated electronics. Hence, as sensor power consumption increases, the benefit of the proposed technique decreases. Fortunately, many sensors have power consumptions that are much lower than that of the fully active sensor network node.

  The geophone used in the ACM application represents an ideal sensor for use with our technique as it is completely self-powered, and does not require amplification. Requirements for powered sensors or active support circuits reduce the energy savings realized by the technique.

  To maximize the power savings possible from the proposed technique, it may be necessary to add a secondary sensor that exhibits favorable power consumption and output characteristics solely for the purpose of event detection. For example, in the ACM application, the geophone is used to detect events. However, upon detecting an event, the system activates a second sensor with much higher power consumption to take a series of detailed measurements.

  It is the power consumption of the sensor used for event detection, not data logging, that is critical. The event detection sensor need not respond linearly, sample at high resolution, have full-scale output, or possess other ideal characteristics. Thus, a variety of unconventional sensors, or sensors operating in unconventional ways, may be used as event detection sensors, e.g.,

  – Solar cells, for light;

- Unbiased microphones, for audio;

- Piezoelectric elements, for vibration; and

- Peltier elements, for temperature differences.

- **Event arrival times should be difficult to predict exactly.** If it is known when the next event is likely or sure to occur, then lucid dreaming is no more effective than conventional timer-based periodic or predictive wake-up is.

- **Events should be infrequent and quickly processed.** As events become more frequent or more time-consuming to process, the mote spends an increasing proportion of its time active, decreasing the effectiveness of lucid dreaming. Many applications that record or react to infrequent phenomena in the environment, e.g., the ACM application, satisfy these criteria.

- **Communication should be infrequent and short.** The effectiveness of the technique depends on the communication behavior of the application. Sensor network nodes often participate in mesh network schemes that require them to wake up and communicate from time to time to perform data aggregation. If communication is frequent and intense, its energy costs may dominate the power savings provided by lucid dreaming. The proposed technique is applicable when moderate to small amounts of data are transferred in response to infrequent events.

- **Event detection should be simple enough to implement using low-power hardware.** Events are detected based on sensor observations. For some applications, detecting events of interest may be quite complex. A key idea in lucid dreaming is moving event detection from software into very low power analog hardware. Constraints on power consumption will generally limit the complexity of this hardware. Our hardware for the ACM application implements threshold detection. Hardware implementation of more complex functions, such as filtering or low-power finite state machines, is also possible, albeit with larger power requirements. Fortunately, lucid dreaming event detection hardware may safely generate some false positive event indications, which are subsequently eliminated without impacting correctness by the sensor network node microcontroller. Thus, even if it is impractical to implement perfectly-accurate event detection in low-power hardware, the proposed technique can still be used in conjunction with hardware that generates occasional false positives to reduce overall mote activation frequency and, therefore, average power consumption. Because the Shake 'n Wake hardware and an attached sleeping mote use significantly less power than an active mote, it is likely that reducing any substantial quantity of false positives through Shake 'n Wake hardware enhancements will be beneficial.

4.2 Hardware

The hardware component (Shake 'n Wake) is the heart of the lucid dreaming technique. It is a simple, ultra-low-power optimized threshold detection circuit designed for direct attachment to a Crossbow MICA2 or MICAz mote. The Shake
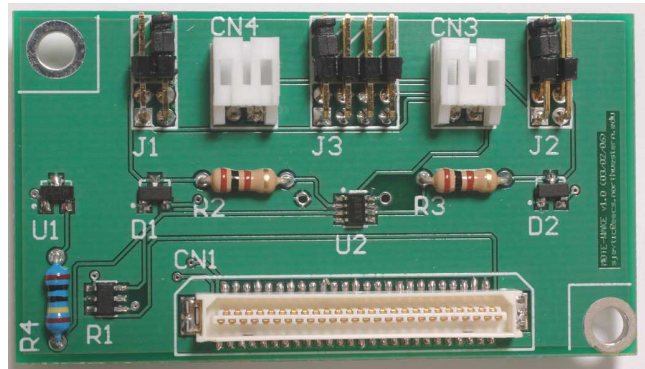

Figure 3: Shake 'n Wake printed circuit board.

'n Wake printed circuit board layout (Gerber files) and bill of materials are available for those wishing to build or have built their own Shake 'n Wake boards.

The Shake 'n Wake printed circuit board (Figure 3) measures $1.25\,\text{in} \times 2.25\,\text{in}$, and has mounting holes and a set of Hirose 51-pin mote expansion connectors that are compatible with MICAz and MICA2 motes. The connectors, which pass through all signals, allow Shake 'n Wake to be placed at an arbitrary location in a MICA2/MICAz hardware stack. The mounting holes, which are connected to GND and surrounded by generous keep-out regions, allow Shake 'n Wake to be physically secured to the hardware stack with ease, while simultaneously avoiding the risk of shorts or other damage. Shake 'n Wake is a two-layer board. The unused area on the top copper has been designated as a polygon fill connected to GND, while the unused area on the bottom copper is a polygon fill connected to VCC. This technique provides some of the benefits of VCC/GND planes, e.g., distributed decoupling capacitance and shielding, without the expense of a four-layer board, which would be required for full power planes. Shake 'n Wake is powered directly from the mote's VCC/GND, as made available on the 51-pin Hirose expansion connectors.

Figure 4 is the schematic diagram for Shake 'n Wake. Its printed circuit board implementation is illustrated in Figure 3. Sensors may be connected to CN1 and/or CN3; J1 and J2 are jumpers used to enable/disable the sensors on CN1 and CN3, respectively. Disabling an unused input, if any, is necessary both to save power and prevent spurious event detection. An input protection network consisting of diodes and resistors protects the hardware from large transients which may result from vigorous shaking of the geophone, electrostatic discharge, or other sources. D1 and D2 are high-performance Schottky clamping diodes; they combine high switching speed with exceptionally low forward voltage and series resistance. R2 and R3 are current limiting resistors that further reduce the system's exposure to damaging transients. Due to exceptionally high input impedance of the comparator, R2 and R3 cause virtually no drop in the magnitude of the incoming sensor signal.

Following the input protection network, the sensor signals are passed to the inverting inputs of the low-power dual comparators contained in U2. The comparators feature $4\,\text{mV}$ of hysteresis internally, providing both noise immunity and clean switching in the presence of a low slew rate, noisy input. The non-inverting inputs of the comparators are connected to a programmable voltage divider subsystem. The
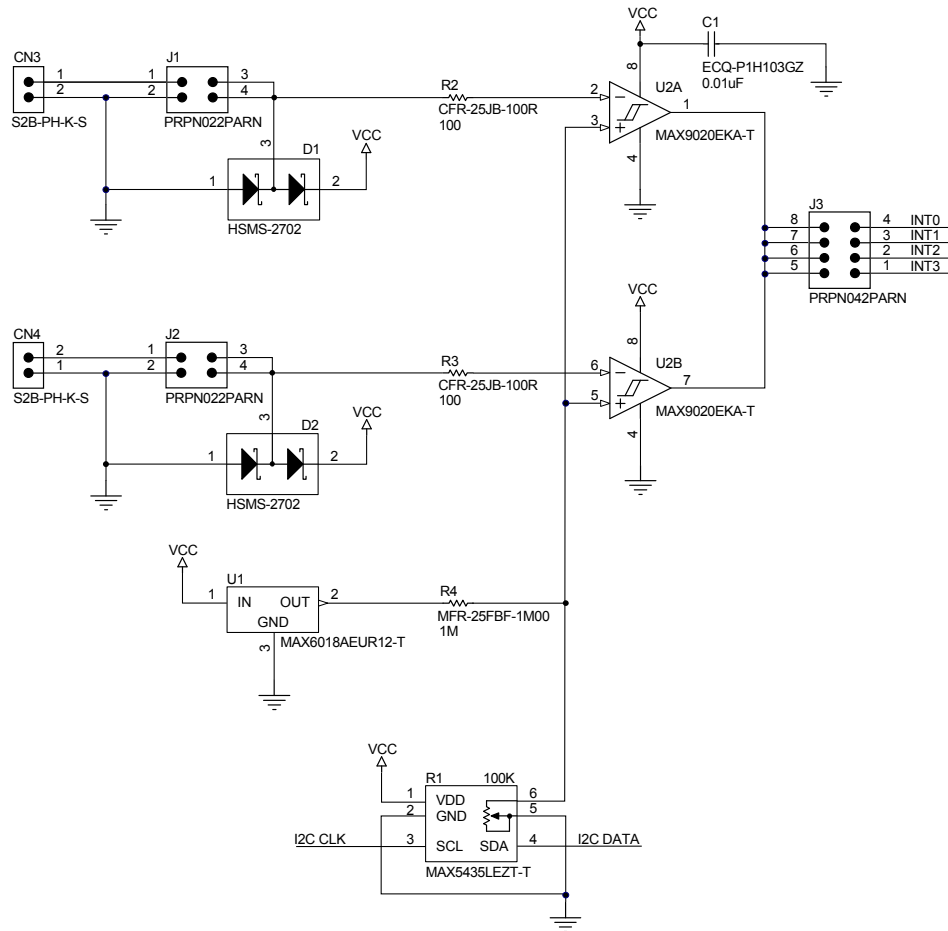
Figure 4: Shake 'n Wake schematic.

output of the comparators are open-drain, allowing them to be directly connected to the active low/level sensitive interrupt lines of the ATMega128L microcontroller in a wired-OR configuration merely by enabling the ATMega128L's internal pull-up resistors. This configuration conserves resources by avoiding the use of a second interrupt line or an OR gate. Thus, whenever the voltage of an enabled sensor input exceeds that of the non-inverting input voltage level, an ATMega128L interrupt line of the user's choice is taken low. The user may select from INT[0..3], as provided on the Hirose connector using J3; these correspond to ATMega128L interrupts INT[5..8], respectively.

The voltage divider subsystem consists of a low-power precision $1.263\,\mathrm{V}$ voltage reference, allowing the inverting input to both comparators to remain constant over the life of the mote batteries without the addition of a voltage regulator and providing immunity from power supply transients. The voltage reference output is connected to a fixed precision $1\,\mathrm{M}\Omega$ resistor in series with a $100\,\mathrm{K}\Omega$, 32-tap digital potentiometer with nonvolatile wiper memory. The digital potentiometer, connected to the mote's I²C bus provides programmatic selection of the voltage provided to the noninverting inputs of the comparators, thereby effectively enabling remote selection of the wakeup stimulus threshold. Although the I²C address of the digital potentiometer is fixed, it does not conflict with any addresses currently in

use in the node hardware we support. Furthermore, alternate addresses may be obtained with the substitution of otherwise identical variants of the digital potentiometer offered by the device's manufacturer. The fixed resistor serves two roles. First, it concentrates the range of possible output voltages of the voltage divider system around the voltage of interest. Second, it greatly increases the resistance of the voltage divider network, thereby avoiding overload on the voltage reference and reducing power consumption in the voltage divider itself.

The Shake 'n Wake hardware design is robust and versatile, but has limitations. First, the high impedance of its voltage divider network, while helping to save power, precludes the connection of mainstream multimeters to the non-inverting comparator inputs to observe the threshold voltage. Such devices do not offer sufficient input impedance to observe the voltage divider output without affecting it. Although this poses no problem during operation, it complicates debugging. Second, the Shake 'n Wake hardware lacks provisions for hot installation/removal due to the design of the Hirose 51-pin connectors used for compatibility with Crossbow MICA2 and MICAz motes. This connector has no mechanism to guarantee that supply rails make contact prior to I/O lines. Furthermore, there is no general mechanism to prevent corruption during an insertion/removal event on any of the interfaces that are made accessible through this connector.

4.3 Software

We program the node hardware in NesC [13] within the TinyOS [18] operating system. The software side of lucid dreaming consists of a small extension to the run-time and some library functions. Note that the technique can also be used within other operating environments such as MANTIS OS [1], or even without a third-party runtime environment. Our original Shake 'n Wake demonstration application was a simple super-loop written in C.

An interrupt service routine for wakeup is introduced. This ISR does not presently do anything. Its execution is simply a side-effect of the interrupt bringing the mote out of sleep. The intent is that after the ISR executes, the mote continues executing the code immediately after the point at which it entered sleep mode.

A library routine called the "sleep preparation routine" is provided. This small function enables the interrupt that activates the Shake 'n Wake board and writes to a sleep register to put the mote into a low-power sleep mode. A second library routine is provided to configure the digital potentiometer, allowing the program to change the threshold level at which an event is generated by Shake 'n Wake.

5. POWER CONSUMPTION AND PERFOR-MANCE MODELS AND MEASUREMENTS

We now present power and performance models for our implementation of lucid dreaming and discuss the results of bench tests with the Shake 'n Wake printed circuit board. The proposed models can be used by application developers to quickly determine the degree to which the proposed technique will improve power consumption. We show the behavior of the models for a range of parameter values corresponding to current hardware and applications. The symbols for our models can be found in Table 1.

5.1 Power Consumption and Battery Lifetime

The average power consumption, P_{AVG_SO}, of a system using software polling event detection can be approximated as follows:

$$
\begin{aligned}
P_{AVG\_SO} =& (F_{DC} \cdot D_{DC})(P_{AC} + P_{S1}) + \\
& (F_{MC} \cdot D_{MC})(P_{AC} + P_{RT}) + (1 - F_{DC} \cdot D_{DC} \\
& - F_{MC} \cdot D_{MC})(P_{AC} + P_{S1})
\end{aligned} \tag{1}
$$

The average power consumption of an equivalent system that detects events using lucid dreaming can be approximated as follows:

$$
\begin{aligned}
P_{AVG\_LD} =& (F_{DC} \cdot D_{DC})(P_{AC} + P_{S1}) + \\
& (F_{MC} \cdot D_{MC})(P_{AC} + P_{RT}) + \\
& (1 - F_{DC} \cdot D_{DC} - F_{MC} \cdot D_{MC})(P_{ZZ}) + \\
& P_{S2} + P_{MW}
\end{aligned} \tag{2}
$$

For the sake of simplicity, both models assume that data collection and communication are mutually exclusive events; this assumption is accurate for the types of applications where the lucid dreaming technique is most appropriate (e.g., applications with infrequent events and infrequent communication).

Depending on the sensor network architecture, changes in processor state or radio state may have significant energy costs, i.e., the power consumption of the processor or radio may increase before they become available for computation or communication. This effect can be modeled by increasing the average duration for event processing, D_{DC}, and/or average duration of communication events, D_{MC}, to include the state transition times.

The literature reports values for P_{RT}, P_{AC}, and P_{ZZ} [5]. P_{S1} and P_{MW} were determined empirically in our lab. P_{S2} is the result of our geophone being a self-powered sensor. F_{DC}, F_{MC}, D_{DC}, and D_{MC} are taken from our experience with the ACM application.

We now illustrate the impact of changing the parameters appearing in our models for a number of applications, sensors, and sensor network node architectures. As indicated in Section 2, some researchers have considered the use of reduced and/or predictive duty cycling in order to reduce power consumption. These approaches cannot be used in applications for which missing events is unacceptable and events have durations that are short compared to the proposed duty cycle period; note that the period must not be short because initializing a mote carries overhead. Even if missing some events is acceptable, in most applications it is not desirable.

Figure 5 displays the battery life of a sensor network node used in the ACM structural integrity monitoring application as a function of the average number of events per day and the tolerable probability of missing each event. We used a typical battery life of 2,600 mAH for each of the AA alkaline cells. This graph compares three approaches: (1) the proposed lucid dreaming approach, a similar approach using the lowest-power analog wake-up hardware for event-driven applications (2.64 mW) we were able to find in the literature [12], and a duty cycling approach. The lucid dreaming and 2.64 mW sensor approaches are guaranteed to detect all events. If events are not predictable, the probability, per event, that the duty cycling approach misses an event is directly related to the proportion of time the system is inactive. As demonstrated in the figure, lucid dreaming consistently outperforms the 2.64 mW sensor approach by well over an order of magnitude. It has lower power consumption than the duty cycling approach except when the number of events per day is extremely high, i.e., over 1,000, and the acceptable event miss probability is very high, i.e., over 0.9. For the ACM application, the expected number of events per day is 10. In this application, the use of lucid dreaming increases the battery life of the application from 10.91 days to 2,669 days, i.e., the battery life is bounded only by the shelf life of the AA batteries used to power the sensor nodes.

The current Crossbow port of TinyOS supports the use of low power states for the processor and radio between the individual samples in a series. During bench tests, this resulted in lower average power consumption during sampling than reported for a MICA2 with a continuously-active microcontroller. However, even if we assume that the power consumption, P_{AC}, is reduced to 1/10 the reported value, the Shake 'n Wake hardware still increases the battery life in the ACM application by 92.6×.

Next, we model schemes in which the arrival of events is predicted. In such schemes, the mote predicts the interval to the next event, and then puts itself to sleep for that interval. Any such predictor will produce both false negatives and false positives. A false negative is the failure to predict an event that does occur in the interval. A false positive is the

Table 1: Definitions of Symbols Used in Mathematical Equations

Variable	Description	Example value for ACM
P_{AVG_LD}	Average power consumption for lucid dreaming	1.3×10^{-4} W
P_{AVG_SO}	Average power consumption for polling solution	3.0×10^{-2} W
P_{AVG_PR}	Average power consumption for event prediction	No example value
P_{RT}	Power consumption of mote radio in transmitting state	3.0×10^{-2} W
P_{AC}	Power consumption of mote CPU in active state	2.4×10^{-2} W
P_{ZZ}	Power consumption of mote CPU in sleeping state	3.0×10^{-5} W
P_{S1}	Power consumption of primary sensor and data acquisition system	5.7×10^{-3} W
P_{S2}	Power consumption of secondary/wakeup sensor	0 W
P_{MW}	Power consumption of Shake 'n Wake hardware	1.6×10^{-5} W
F_{DC}	Average frequency of an event resulting in data collection	1.2×10^{-4} Hz
F_{MC}	Average frequency of a communication transmission	1.2×10^{-5} Hz
D_{DC}	Average duration of an event resulting in data collection	3.0 s
D_{MC}	Average duration of a communication transmission	104.0 s
F_{TP}	Average frequency of true positives	No example value
F_{FP}	Average frequency of false positives	No example value
Γ_{FN}	False negative probability (type I error)	No example value
Γ_{FP}	False positive probability (type II error)	No example value
Γ_{TP}	True positive probability $(1 - \Gamma_{FN})$	No example value
Γ_{TN}	True negative probability $(1 - \Gamma_{FP})$	No example value

prediction of an event that does not occur in the interval. False negatives decrease power consumption, because the mote is not awakened, and increase the miss rate, because the mote should be awakened. False positives increase power consumption, because the mote is awakened when it should not be, and do not affect the miss rate, because we assume the awakened mote can determine that the event has been falsely predicted.

The model used for evaluating the lucid dreaming technique in the presence of a wide range of parameters assumes Poisson arrival processes for actual events, true positives, and false positives. The mean frequencies of the latter are derived from the former. Let the mean frequency of true positives (correctly predicted events) be

$$F_{TP} = F_{DC} \cdot \Gamma_{TP} = F_{DC}(1 - \Gamma_{FN}) \qquad (3)$$

and the mean frequency of false positives be

$$F_{FP} = F_{DC} \cdot \Gamma_{FP} \qquad (4)$$

where the Γ_{FN} is the false negative probability and Γ_{FP} is the false positive probability. Our model for the average power consumption using event prediction is then a variant of that for lucid dreaming (Equation 2):

$$
\begin{aligned}
P_{AVG\_PR} =& (F_{DC}(\Gamma_{FP} + (1 - \Gamma_{FN}))D_{DC})(P_{AC} + P_{S1}) + \\
& (F_{MC} \cdot D_{MC})(P_{AC} + P_{RT}) + \\
& (1 - F_{DC}(\Gamma_{FP} + (1 - \Gamma_{FN}))D_{DC} \\
& - F_{MC} \cdot D_{MC})(P_{ZZ}) \qquad (5)
\end{aligned}
$$

Event prediction involves a tradeoff between power consumption and the probability of missing an event. Furthermore, this tradeoff depends on the nature of the predictor bias. For an unbiased predictor, the false positive and false negative rates will be identical ($\Gamma_{FP} = \Gamma_{FN}$). In this situation, the power consumption for event prediction will be virtually identical to that of lucid dreaming: Equation 5 converges to Equation 2. However, the probability of missing an event in the event prediction scheme will be Γ_{FN}, which may be large, while the miss probability in lucid dreaming will always be zero.

5.2 Experimental Measurements

We have conducted tests of the Shake 'n Wake printed circuit board. When used to wake the microcontroller in response to vibration, its power consumption is 16.5 µW. We have successfully used in-system programming of Shake 'n Wake's non-volatile Maxim MAX5435LEZT-T potentiometers to vary the event interrupt triggering threshold across a wide range of voltages. Measurements of the MICA2 in different power states [5], and the impact of the Shake 'n Wake board upon the amount of time spent in each power state, indicate that for the ACM structural integrity monitoring application, the combined long-term average power consumption of the MICA2 processor–radio board, the MDA300 data acquisition board, the Shake 'n Wake board, and the sensors will be reduced from 29.8 mW to 121.8 µW by using the Shake 'n Wake implementation of lucid dreaming, i.e., battery life will be increased from 10.91 days to seven years. In other words, battery life will be limited only by the shelf life of the batteries. Moreover, the use of energy scavenging begins to merit consideration.

6. CONCLUSIONS AND FUTURE WORK

There is a mismatch between existing sensor network architectures and event-driven applications. We have proposed lucid dreaming, a hardware–software technique that remedies this mismatch and characterized the situations in which the technique is appropriate. We have designed, built, and tested an implementation (the Shake 'n Wake board
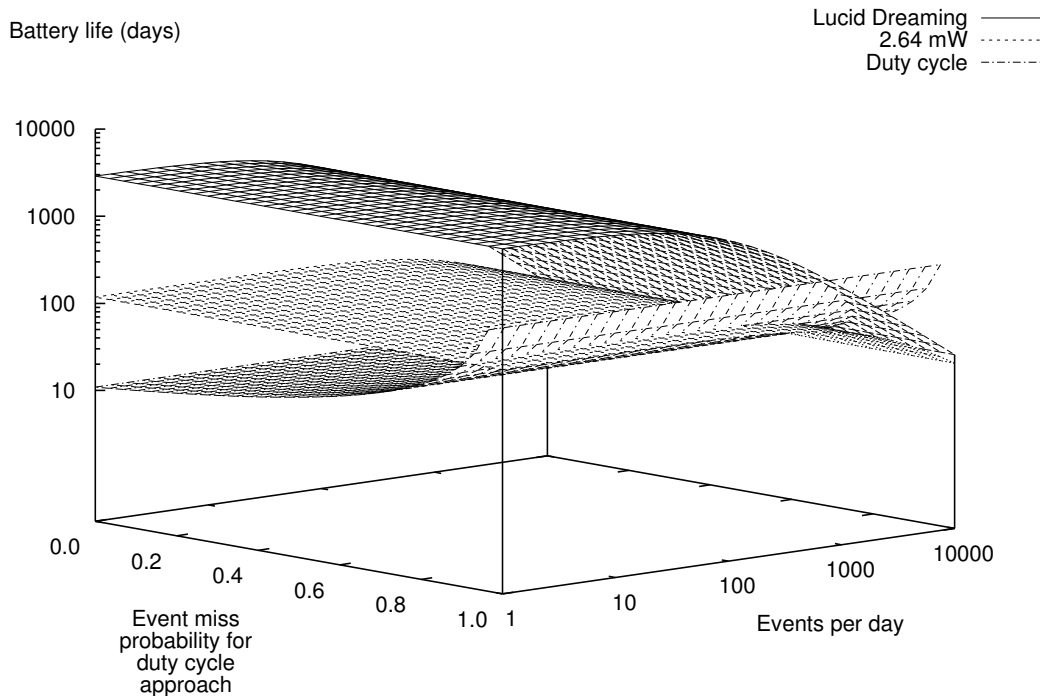
Figure 5: Battery life as a function of event miss probability and F_{DC}.

and software) of our technique for use in structural integrity monitoring of buildings and bridges that reduces power consumption to 1/245 that required by existing approaches. This implementation is compatible with Crossbow MICAz and MICA2 motes.

We plan to expand the capabilities of Shake 'n Wake by using ultra-low-power asynchronous finite state machines to support more complex event detection functions. More broadly, we plan to expand Shake 'n Wake into a general-purpose analog toolbox from which power and rate critical portions of the sensor network application can be constructed.

For applications similar to that described in Section 3, the electronic Gerber format printed circuit board specifications are available from the authors. For applications running on host platforms other than the Crossbow MICA2 and MICAz, or applications with sensing parameters that differ greatly, we hope that the schematic depicted in Figure 4 and described in Section 4 provide a useful starting point to other researchers and designers.

7. REFERENCES

[1] ABRACH, H., BHATTI, S., CARLSON, J., DAI, H., ROSE, J., SHETH, A., SHUCKER, B., AND HAN, R. MANTIS: System support for MultimodAl NeTworks of In-situ Sensors. In *Proc. Int. Wkshp. Wireless Sensor Networks and Applications* (Sept. 2003), pp. 50–59.

[2] AGARWAL, Y., SCHURGERS, C., AND GUPTA, R. Dynamic power management using on demand paging for networked embedded systems. In *Proc. Asia & South Pacific Design Automation Conf.* (Jan. 2005), pp. 755–759.

[3] AKKAYA, K., AND YOUNIS, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks 3*, 3 (May 2005), 325–349.

[4] AL-KARAKI, J., AND KAMAL, A. Routing techniques in wireless sensor networks: A survey. *IEEE J. Wireless Communications 11*, 6 (Dec. 2004), 6–28.

[5] ANASTASI, G., CONTI, M., FALCHI, A., GREGORI, E., AND PASSARELLA, A. Performance measurements of more sensor networks. In *Proc. Int. Wkshp. on Modeling, Analysis, and Simulation of Wireless and Mobile Systems* (Oct. 2004).

[6] Automated crack measurement project. http://www.iti.northwestern.edu/acm.

[7] BARZILAI, A. *Improving a Geophone to Produce an Affordable Broadband Seisometer*. PhD thesis, Department of Mechanical Engineeering, Stanford University, Jan. 2000.

[8] BRINCKER, R., LAGO, T., ANDERSEN, P., AND VENTURA, C. Improving the classical geophone sensor element by digital correction. Tech. rep., Pinocchio Data Systems, Feb. 2005.

[9] CROSSBOW TECHNOLOGY INC. *MICAz Wireless Measurement System Datasheet*, 2006. Document Part Number 6020-0060-03 Rev A.

[10] DOWDING, C. H., AND MCKENNA, L. M. Crack response to long-term and environmental and blast vibration effects. *J. Geotechnical and Geoenvironmental Engineering 131*, 9 (Sept. 2005), 1151–1161.

[11] DOWDING, C. H., OZER, H., AND KOTOWSKY, M. Wireless crack measurment for control of construction vibrations. In *Proc. Atlanta GeoCongress* (Feb. 2006).

[12] DUTTA, P., GRIMMER, M., ARORA, A., BIBYK, S., AND CULLER, D. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *Proc. Int. Conf. on Information Processing in Sensor Networks* (Apr. 2005).

[13] GAY, D., LEVIS, P., VON BEHREN, R., WELSH, M., BREWER, E., AND CULLER, D. The nesC language: A holistic approach to networked embedded systems. In *Proc. Programming Language Design and Implementation Conf.* (June 2003).

[14] HILL, J., SZEWCZYK, R., WOO, A., HOLLAR, S., CULLER, D., AND PISTER, K. System architecture directions for networked sensors. In *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems* (Nov. 2000).

[15] KOTOWSKY, M., AND OZER, H. Wireless data acquisition. Crossbow Smart Dust Challenge, 2004. http://www.iti.northwestern.edu/research/projects/ dowding/micro.html.

[16] KUBISCH, M., KARL, H., WOLISZ, A., ZHONG, L. C., AND RABAEY, J. Distributed algorithms for transmission power control in wireless sensor networks. *Wireless Communications and Networking 1* (Mar. 2003), 558–563.

[17] KURATA, N., JR., B. F. S., RUIZ-SANDOVAL, M., MIYAMOTO, Y., AND SAKO, Y. A study on building risk monitoring using wireless sensor network MICA mote. In *Proc. Int. Conf. on Structural Health Monitoring and Intelligent Infrastructure* (Nov. 2003), pp. 353–357.

[18] LEVIS, P., MADDEN, S., GAY, D., POLASTRE, J., SZEWCZYK, R., WOO, A., BREWER, E., AND CULLER, D. The emergence of networking abstractions and techniques in TinyOS. In *Proc. Symp. Networked Systems Design and Implementation* (Mar. 2004).

[19] LYNCH, J. P., LAW, K. H., KIREMIDJIAN, A. S., KENNY, T. W., CARRYER, E., AND PARTRIDGE, A. The design of a wireless sensing unit for structural health monitoring. In *Proc. Int. Wkshp. on Structural Health Monitoring* (Sept. 2001).

[20] MITCHESON, P. D., YATES, D. C., YEATMAN, E. M., GREEN, T. C., AND HOLMES, A. S. Modelling for optimisation of self-powered wireless sensor nodes. In *Proc. Int. Wkshp. Wearable and Implantable Body Sensor Networks* (Apr. 2005).

[21] OZER, H. Wireless crack measurement for control of construction vibrations. Master's thesis, Department of Civil and Environmental Engineering, Northwestern University, July 2005.

[22] POLASTRE, J., SZEWCZYK, R., AND CULLER, D. Telos: enabling ultra-low power wireless research. In *Proc. Int. Symp. Information Processing in Sensor Networks* (Apr. 2005).

[23] POLASTRE, J., SZEWCZYK, R., SHARP, C., AND CULLER, D. The mote revolution: Low power wireless sensor network devices. In *Proc. Symp. High Performance Chips* (Aug. 2004).

[24] RABEY, J. M., AMMER, M. J., DA SILVA JR., J. L., PATEL, D., AND ROUNDY, S. PicoRadio supports ad hoc ultra-low power wireless networking. *IEEE Computer* (July 2000), 42–48.

[25] RAGHUNATHAN, V., SCHURGERS, C., AND SRIVASTAVA, S. P. M. B. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine 19*, 2 (Mar. 2002), 40–50.

[26] ROUNDY, S., WRIGHT, P. K., AND RABEY, J. A study of low level vibrations as a power source for wireless sensor nodes. *Computer Communications 26* (Oct. 2003).

[27] SCHOTT, B., BAJURA, M., CZARNASKI, J., FLIDR, J., THO, T., AND WANG, L. A modular power-aware microsensor with $> 1000\times$ dynamic power range. In *Proc. Int. Symp. Information Processing in Sensor Networks* (Apr. 2005), pp. 469–474.

[28] SINHA, A., AND CHANDRAKASAN, A. Dynamic power management in wireless sensor networks. *IEEE Design and Test of Computers* (Mar. 2001), 62–74.

[29] XU, N., RANGWALA, S., CHINTALAPUDI, K. K., GANESAN, D., BROAD, A., GOVINDAN, R., AND ESTRIN, D. A wireless sensor network for structural monitoring. In *Proc. Conf. on Embedded and Networked Sensor Systems* (Nov. 2004).

[30] ZHENG, R., HOU, J. C., AND SHA, L. Asynchronous wakeup for ad hoc networks. In *Proc. Int. Symp. Mobile Ad Hoc Networking and Computing* (June 2003), pp. 35–45.