

Energy-Efficient Platform Designs for Real-World Wireless Sensing Applications

Pai H. Chou and Chulsung Park

Center for Embedded Computer Systems, University of California, Irvine, CA 92697-2625, USA

Abstract—Real-world wireless sensing applications demand system platforms with a wide range of size, cost, power consumption, connectivity, performance, and flexibility requirements. These goals cannot be achieved without understanding the nature of the sensing functions in the first place, which can be classified into passive vs. active sensing, event detection vs. data acquisition, and real-time monitoring vs. data logging. This paper discusses platform design techniques for supporting these design goals through the trade-offs of sensing devices, wireless interfaces, and computation and control units. We also cover power subsystem design for supply-aware optimizations, including load/supply matching, power de-fragmentation in multi-supply systems, and use of supercapacitors. To evaluate these system platforms, we describe an emulation-based benchmarking methodology to quantify fitness metrics.

I. INTRODUCTION

Wireless embedded sensing systems (WESS) are one of the major driving forces behind many of the recent innovations in science and engineering. Their applications range from defense and environmental monitoring to health and business. WESSs not only enable scientific and medical researchers to collect data in brand new ways, but they also make a rich source of research topics and educational activities in multiple disciplines. Real-world wireless sensing applications are quite diverse, and they impose a wide range of constraints on the system platforms, including the size, cost, power availability, wireless connectivity, performance, memory, storage, and flexibility.

End users of a real-world application must often face the question of whether to purchase and use one of the existing sensor platforms or to build their own new platform. Existing platforms range from *sensor nodes* on the low end, such as the Mica2 mote [1] with an 8-bit microcontroller (MCU) running a thin software layer, to *sensor computers* on the high end, such as the Stargate [2] with a 32-bit CPU running Linux. The choice among platforms is not always obvious. Platform vendors today resort to citing power or data rate figures from data sheets for components such as MCUs or RF transceivers to characterize their system performance. However, this can be very misleading because they can exhibit very different characteristics when functioning as parts of a system. Unlike general-purpose computers, the equivalent of SPEC benchmark suite does not exist today for the purpose of quantitative evaluation; even if it does, there is no easy way to execute such a benchmark. The lack of generally accepted quantifiable metrics for WESS platforms has made it nearly impossible to measure progress in this field.

Before defining metrics for quantitative evaluation, it is

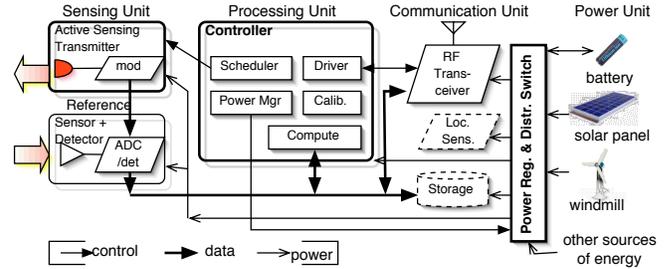


Fig. 1. Generalized architecture of a wireless embedded sensing system.

necessary to first classify WESSs according to their functions. Fig. 1 shows a general block diagram that encompasses the major components in a WESS for the purpose of classification. Each component can be implemented in a slightly different way, according to the required functions of the WESS's target application. WESSs can be categorized into

- *Event detection vs. Data acquisition vs. Data aggregation*
- *Passive vs. Active sensing*
- *Data logging vs. Real-time monitoring*

A. Event Detection, Data Acquisition, Data Aggregation

Examples of *event detectors* include smoke detectors, motion detectors, and light detectors, whose purpose is to determine the presence of smoke, motion, and light above a chosen threshold, as opposed to their actual values. On the other hand, *data acquisition* sensors must report the *magnitude* of the event being monitored, e.g., a seismographer or a thermometer. Hybrid schemes are also possible. For example, a seismographer may be normally idle, and vibrations above a certain threshold will wake it up for data acquisition functions. On the other hand, other applications such as a vehicles's line-of-bearing computation [3] need only aggregated data, which is always less voluminous. Such a WESS first collects data from sensing devices, stores them in the local memory, and runs aggregation algorithms. Then, the aggregated result will be transmitted. This scheme can save transmission power but requires more hardware resources such as a faster microprocessor and larger amount of memory for running relatively complex aggregation algorithms.

B. Passive vs. Active Sensing

A *passive* sensor measures the readily observable signals on the subject. For a passive sensing system, all sensor devices

look alike, as long as they have a voltage interface. As a result, some WESS platforms make sensor devices part of interchangeable modules that can be plugged in to the expansion slot. The primary difference is the sampling rate, bit-resolution, and the number of output signals from sensor devices to sample.

An *active* sensor emits a signal directed at the subject and measures its reflection. The signal may be an impulse, sinusoidal, or some constant intensity light, sound, or electromagnetic wave. The sensor device may still output a voltage or a current, but its value over time must be further interpreted in order to extract any information encoded in the reflection. An example is a sonar, which emits an impulse of sound and measures its echo to determine the distance from the subject. In this case, the *magnitude* of reflection is not important; it is the *time delay* of the echo that determines the distance.

C. Real-Time Monitoring vs. Data Logging

Virtually all wireless sensing applications require the collected data or detected event to be reported back to a control room or other automated response system. The difference is how urgently the reporting function must be executed. In case of fire detectors, intruder detection systems, or traffic monitoring, most likely the response should be near-real-time – that is, preferably instantaneous but tolerable up to a few seconds. For other applications such as habitat monitoring or Zebrant [4] where the goal is to collect statistics, then it is unnecessary to transmit data immediately. Instead, by logging the data or events in some nonvolatile memory and transmit in batch at a later time, it can potentially save energy through reduced communication overhead as well as better tracking of using harvested energy such as solar power.

D. Towards Quantitative Evaluation

Techniques for achieving energy efficiency can be quite different for the different classes of application requirements. A low power technique that is effective for one class of application may be counterproductive or even unacceptable in another class. This paper uses a number of case studies to highlight the energy issues in the context of different application requirements. Based on the proposed classifications, we discuss platform design techniques on both the consumption side and the supply side. Consumption-side issues include selection of sensing devices, wireless interfaces, computation and control units, and memory sizing and organization. Supply-side issues include load matching, power de-fragmentation, and new uses of supercapacitors in the power subsystem as battery replacement. We describe a new quantitative evaluation methodology based on emulation-driven measurement of actual WESSs. By translating performance figures into a value that quantifies the WESS's fitness to the application, we will be able to compare different WESSs in more systematic, objective ways.

II. CASE STUDIES OF SENSING APPLICATIONS

This section examines case studies of four wireless sensing applications. The application classes are by no means exhaustive, but we use them to illustrate the suitability of proposed

platforms for these applications. They are habitat monitoring, civil structural health monitoring, infant monitoring, and physiological state monitoring.

A. Habitat Micro-climate Monitoring

In the habitat monitoring application, sensor nodes are deployed as miniature weather stations. They collect climatic data including humidity, air pressure, temperature, radiation, and infrared, at a wide range of locations ranging from tree tops to underground settings [5]. Existing system platforms for this application achieve energy-efficient operations through low duty cycling. This is possible because climate conditions are not expected to change dramatically within a short period of time, and thus monitoring applications sleep most of the time to conserve energy, waking up periodically for very short runtimes. Thus, the design problem is mainly concerned with turning low duty cycling into opportunities for saving power.

A number of WESSs in the *sensor node* category exploit the low duty cycling requirements as the primary opportunity for saving energy. Energy efficiency is achieved primarily through software control by maximizing the sleep time of the MCUs and the RF modules. Most MCUs today support low-power sleep mode and other low-power modes, as will be discussed in §III-A. The radios of choice tend to be lower-frequency ISM bands because they have lower data rates but can transmit a longer distance for the same power, to be discussed in §III-B.

B. Structural Health Monitoring

Structural health monitoring refers to sensing and extraction of features that quantify the integrity of a structure and assess its ability to perform its intended function given aging and degradation over time. They are applicable to a wide range of systems from airplanes and many mechanical structures, and here we focus on civil structures such as highway bridges and skyscrapers. Proposed techniques analyze strain or ambient vibration on the order of milli-g (where g is a unit of acceleration that is equal to the earth's gravity). Damage can be detected by modal analysis, which entails transforming the time-domain data into frequency domain and extracting the dominant frequency component for comparison against that of a healthy structure. The sampling requirements are 200–1000 samples per second at 10–12 bits per sample in X, Y, Z axes. Time-domain analysis techniques may require a higher rate of 2K–10Ksps. In either case, WESSs for this application belongs to *data acquisition*, *real-time monitoring*, and *passive sensing*.

Unlike habitat monitoring, the much higher duty cycling requirement means there is little or no opportunity for dynamic power management. A Mica2 configured for such an application will operate for about 4–5 hours on its two AA batteries. Such a system will have no choice but to replenish the energy with alternative sources such as solar panels. The low, modem-speed radio driven by a software MAC will either overutilize the MCU or saturate the available bandwidth with just one node, because of the large amount of data generated at a higher sampling rate.

To address these problems, the authors constructed a system named DuraNode [6]. DuraNode maximizes solar-panel supply efficiency through load matching, as discussed in §IV. It uses the 802.11b wireless interface for the high bandwidth and low energy-per-bit, and the trade-offs with raw radios are discussed in §III-B. Another limiting factor is the noise problem in the sampled signal. Noise can be reduced by board-level design and use of regulators. Another source of noise is *jitter*, the deviation from a precise time distance between two consecutive samples, and this can translate into large noise in the frequency domain. Jitter reduction at the architecture level is discussed in §III-A.

C. Pre-term Infant Monitoring

The third case study involves monitoring the spontaneous motion of pre-term infants. One way to help them grow in weight and bone strength is to apply assisted exercise by moving their arms and legs as a way to stimulate their spontaneous movement. Their progress must be closely monitored to ensure the infants are not adversely assisted. Acceleration data at 10–100sps are usually required for this application. Commercial products such as ActiGraph are cordless data loggers that are good for adults, but weighing over 17g, they are too large and heavy for pre-term infants; besides, they cannot support real-time monitoring. Although the smallest Mica “DOT” is just slightly larger than a US quarter coin, after adding the expansion module for accelerometers and the coin battery, the DOT also becomes too big for the infant.

To solve these problems, we designed a WESS called Eco [7]. With a volume of 1.008cm³ and a weight of 3.6g including two lithium coin batteries, Eco fits the size and weight constraints. Eco achieves power-efficient short-distance wireless communication using the 2.4GHz radio transceiver on the nRF24E1 MCU, to be described in §III-A. Although there are few opportunities for dynamic power management, energy-efficiency can be achieved through trade-offs between the transmission power and the bit-error rate (§III-B).

D. Physiological State Monitoring

The fourth application is for monitoring physiological state by optical spectroscopy, which entails beaming light into live tissue and measuring the backscattered light. One approach, called frequency domain photon migration (FDPM), characterizes tissue content by the *scattering* and *absorption* coefficients (μ'_s, μ_a) over different optical wavelengths λ in the near-infrared (600nm–1000nm) regime [8]. *Time-resolved* techniques have been proposed to further improve the accuracy of the detection by using broadband, intensity-modulated light instead of constant-intensity light, since this enables the clearer separation between the scattering and absorption events by their timescale [9]. This technique has been applied to non-invasive detection of breast cancer, internal body temperature, glucose level, and many other physiological signs.

This is an example of *active sensing*, because it detects how the subject reacts to the signal it emits. None of the available WESS platforms meet the requirements of this application, and

the authors’ laboratory built one called the Mini-FDPM [10]. For active sensing, the system must perform real-time control to coordinate both the emitter and the detector. The “sensor” is actually a comparator chip, AD8302, which compares the reference sinusoidal with that from the photo detector and outputs two analog signals whose voltages are proportional to the phase and amplitude difference. Two ADCs then sample these two signals at 100Kbps until they stabilize for each modulation frequency. As with many other active sensors, the dominant power consumption is in the actuation part. The Mini-FDPM outputs 21dBm of power to the laser diode and two of the largest sources of power waste are impedance mismatch and optical coupling. Energy efficiency therefore must be achieved in the analog and optical domains. Using a *sensor computer* is not necessarily the right answer.

III. ENERGY CONSUMPTION

This section discusses techniques for achieving energy efficiency on the *consumption* side. The power consuming side of a WESS system architecture consists of the MCU, sensor devices, and RF, plus the actuator if it is an active sensor. The RF and the actuator normally consume the most power. Unfortunately, there is usually not much that the system designer can do to reduce the actuation power other than turning them off to the extent allowed. Even though the MCU is usually not the highest power consumer, it turns out that choosing the right MCU, and more generally the processing elements (PE) and the associated architecture, can make the greatest difference in improving *system-level* energy efficiency. This section discusses the selection of PE and wireless interfaces for efficient energy consumption.

A. Processing Elements

To define a system architecture, designers typically starts by selecting an MCU. The choices can be based on considerations for performance, software, built-in I/O interfaces, and power management capabilities.

1) *Performance*: We divide WESS architectures into sensor nodes and sensor computers. A *sensor node* normally uses an 8-bit or 16-bit MCU with limited memory, and the primary purpose is to control the other modules in the system. On the other hand, *sensor computers* are modeled after general purposes computers with a 32-bit CPU running an OS. For example, Stargate [2], WINS [11], μ AMPS [12], and PASTA [13] contain a 400MHz XScale or 133–233MHz StrongARM CPU. These powerful CPUs are used to support Linux or a real-time OS. When general-purpose MCUs cannot deliver the performance (e.g., due to the lack of floating point support in hardware), then special-purpose PE can be introduced. For instance, the iBadge [14] uses not only an ATmega but also a TI DSP C5416. The decision should be made based on analysis of the required features as discussed in the case study section earlier.

2) *Software Availability*: The available software base can also influence the consideration. For example, many researchers choose to write their code in nesC and run it on

TinyOS [15]. If TinyOS and the compiler tools have been ported to a particular MCU, then choosing that MCU will enable the designers to get access to the software made available for that platform. TinyOS supports cooperative multitasking and task scheduling. The BTnode also provides lightweight system software for device drivers and the dispatcher [16].

3) *Integrated I/O*: Many low-power 8-bit MCUs have built-in I/O interfaces. For example, the Atmel ATmega 128L MCU [17] used by the Mica2 and BTnode has a built-in 8-channel 10-bit ADC, an analog comparator, PWM channels, SPI, dual USARTs, and general purpose I/O pins. For sensor nodes that are size-constrained, having extra I/O resources will waste valuable board area, while not having sufficient I/O will require external components and extra power for the glue logic. Previously mainly on 32-bit MCUs but more recently even lower-end MCUs such as the TI MSP 430 [18] used in the Telos [19] start supporting DMA, enabling the ADC to write to memory directly. Many 16-bit and 32-bit MCUs have higher speed and more complex I/O interfaces. For instance, the 68HC12 has one version with built-in Fast Ethernet (100Mbps), enabling it to support a TCP/IP protocol stack. One of the most interesting trends is the integration of RF and other wireless modules with 8-bit MCUs. For instance, the Nordic nRF24E1 [20] contains a 2.4GHz RF transceiver, 9-input 12-bit ADC at 1000Kbps, and an 8501 MCU core all in a package that is 6mm on each side. The Atmel AT86RF210 [21] is slightly larger at 7mm on each side but includes an AES encryption engine and protocol stack support for a ZigBee transceiver.

4) *Power*: The ATmega 128L has six sleep modes and software settable frequency from 0–8MHz. It has a 128K program flash, 4K internal SRAM or 64K external memory. The MSP 430 has even lower current consumption, at 280 μ A/MIPS and 1.1 μ A standby. The NEC 78K0S family (μ PD7894xx) uses a novel, 3-tier clocking scheme that enables even finer-grain control of the CPU and I/O controllers. It draws 0.6mA of current in full-on mode at 5MHz, 0.9 μ A at 32KHz, and 0.05 μ A in stop mode while retaining memory content. Among 32-bit CPUs, the StrongARM 1100 used by the WINS runs at 133MHz draws up to 300mW, <200mW typically, <40mW when idle, and <0.8mW in sleep mode. However, its sleep power is much closer to the full-on power of an 8-bit MCU, whose sleep power is 2–3 orders of magnitude lower. The Atmel AT91 [22] with its 32-bit embedded ARM7 core has lower inactive power < 100 μ W, though the active power is still 90mW at its slowest speed. The XScale core can go down to 50mW at 150MHz, though it is not its most power efficient (joules per MIPS) operating point.

For sporadic event detection or low-rate data acquisition, power management can be accomplished by executing a sleep instruction. The MCU can be waken up by either a built-in voltage comparator tied to an interrupt, or by a timer that may be either on-chip or off-chip.

5) *Memory*: MCUs like any computer require both program memory and data memory. Programs usually reside in flash memory, which may be on-chip or off-chip. They are writable

by the MCU itself either in bootloader mode or explicitly programmed with this feature for in-field software upgrade. RAM sizes are usually much smaller because of the high power consumption, but RAM is faster than flash. In fact, although it is possible to run many MCU programs directly out of flash, one can see significant speedup by loading program from flash into RAM and run from RAM.

6) *Multi-MCU Architectures*: A multi-MCU architecture may be particularly suitable for *Real-Time Data Acquisition* applications, which necessitate accurate/fast data sampling capability as well as very frequent data reporting. In addition, each WESS may also need to form an ad-hoc network or relay packets from its neighbor. One may consider a high-end MCU at a faster clock to meet the higher performance demand, but this will result in much higher power consumption. Instead, the authors developed a dual-MCU architecture, which consists of two low-power MCUs and one shared FIFO memory. Higher performance/power ratio is achieved by distributing workload onto two MCUs. In the DuraNode example [6], one MCU is responsible for the sensing control, while the other performs the networking tasks such as controlling the RF interface and running networking algorithms. The FIFO also enables the data to be buffered and transmitted in batch for amortizing communication overhead and possibly creating opportunities for DPM on the RF.

B. Wireless Connectivity

Wireless is usually the highest power consumer in passive sensing systems, or second highest after actuators in active sensing systems. Most WESSs use radios for wireless communication. Infrared (IR) can have a peak bandwidth of 4Mbps. It is inexpensive, does not cause and is immune to electromagnetic interference that may be present in the sensing environment. IR is used by the MIT Pushpin [23], and it works particularly well for very dense, mainly indoors deployment. One advantage is that nodes in adjacent rooms do not interfere with each other. However, IR is prone to interference by fluorescent light and sun light. The overwhelming majority of sensor platforms thus use RF. The choice of RF interface is determined by the requirements on transmission range, data rate, usage pattern and purpose, and the power and energy budget. We focus our discussion on the choice of radios and their interactions with the battery.

1) *Choice of Radios*: Today's approaches can be divided into three groups: custom radio, commercial off-the-shelf (COTS) transceiver with custom MAC, and COTS transceiver/MAC modules. Some researchers implement custom radios for ultra low power, better integration, or scalability, while others choose commercial off-the-shelf interfaces. With a few exceptions [24], most radios use the license-free ISM (industrial, scientific, and medical) bands, although FCC imposes some restrictions. For instance, FCC 15.231 [25] limits periodic data transmission in the 260–470MHz band to be under one second per hour. Higher data rates are allowed in the 902–928MHz, 2400–2483MHz, 5725–5875MHz, and 24–24.25GHz bands, though not exceeding 1mW; or up to 1W if

spread spectrum is used [26].

Some COTS radios are transceiver only. For instance, the Mica2 uses the Chipcon CC1000 433MHz transceiver at up to 38.4kbps. MAC processing is done in software on the 8-bit MCU. This gives the software direct control over all aspects of the radio usage, from immediate acknowledgment and low-power listening to ad hoc networking protocol. However, this also limits the data rate for data acquisition functions. For instance, even though the built-in ADC on the Mica2 can sample substantially faster, it cannot sustain the same high data acquisition rate.

Many sensor nodes use commercial off-the-shelf radios such as 802.11 and BlueTooth. The UCLA iBadge [14] and ZTH BTNode [27] both use BlueTooth, a 2.4GHz radio standard (IEEE 802.15.1) commonly found in keyboard, mouse, and phone-to-computer or phone-to-earpiece connections. Class 1 BlueTooth can transmit up to 100m at up to 721kbps, while Classes 2 and 3 are up to 10m. BlueTooth devices can form a *piconet* with one master and up to seven active slaves while the other additional slaves are “parked,” and multiple piconets can form a *scatternet* with multi-hop communication. Some researchers found BlueTooth to have better energy per bit than the Mote’s custom radio but consume much higher power during inactivities [28]. ZigBee [29] is a new standard on top of IEEE 802.15.4. It addresses some of BlueTooth’s shortcomings in low duty-cycle applications and is used by the Telos [19]. IEEE 802.11* (a, b, g) radios are even lower energy per bit than BlueTooth, but they require higher power and have higher data rates. They are used in iPaq-based prototype nodes and sensor gateways (“macro Motes”) that relay data for nearby sensor nodes [2], [30].

One radio that provides a good mix of support is the nRF2401 radio chip. It contains two receivers and one transmitter in the 2.4GHz band with 1MHz frequency bands whose center frequencies must be at least 8MHz apart to avoid interference. The receivers can listen on two channels or be in transmit mode but not receive and transmit at the same time. The chip supports either direct mode – which is direct access to the radio without MAC – or in shockburst mode, which is Nordic’s own MAC that buffers up the payload and wraps its own header and checksum before sending it at 1Mbps. This way, the radio can use shockburst mode for data payload, or direct mode for precise clock synchronization. Alternatively one of the channels can be dedicated to listening on a beacon signal.

2) *Radio Power*: Because radio is usually the largest power consumer, optimizing radio power can result in significant improvement at the system level. Many radios allow the user to set the power level. For instance, the CC1000 has 23 levels of transmission power, from -20 dBm to $+10$ dBm. Setting the transmission power high will result in higher SNR (signal-to-noise ratio) and lower BER (bit-error rate), but at the same time it could increase interference with farther radios. More importantly, drawing higher current than the rated current of the given battery will decrease the battery efficiency, due to the *rate capacity* effect. On the Mica2DOT sensor node, the

CR2354 lithium coin battery has a rated current of 0.2mA. Experimental results have shown that the optimal transmission power is between 4–6 dBm; at 10 dBm, the battery efficiency drops down to 26–35% of the optimal [31].

For enhancing transmission distance, one can use more transmission power, alternative antenna design and frequency selection schemes. However, raising the transmission power is not always a viable option due to energy constraints. PCB antennas have worked well for 2.4GHz radios in direct line of sight, although they are still big. Chip antennas can be a good compromise: they can be made much smaller than PCB antennas while delivering better range than other alternatives of the similar size. For example, AN9520 [32] from RainSun measures $10 \times 2 \times 1$ mm, but its gain reaches 1.5dBi. The smallest size is only $3.2 \times 1.6 \times 0.9$ mm (AN3216 [33]) and its gain is 0.5dBi.

IV. ENERGY SUPPLIES

For WESS platforms, energy optimization cannot be done on the consumption side alone; the supply side must also be considered for several reasons. First, different sensing devices may require multiple voltages, from 3.3V to 5V, 9V, and 12V [34], and these require additional power regulation or conversion circuitry. Second, currently batteries are the primary energy source, but their capacity is finite. Thus, there is a growing need for recharging by harvesting energy from sunlight, wind, or other sources in the environment to support extended operations. Third, batteries are rated for some nominal current; drawing current above this level can significantly degrade the battery efficiency due to the rate capacity effect [35], [36]. Some researchers have proposed *battery friendly* discharge patterns, though the use of switching regulators may dampen their effectiveness.

Ideally, one would like to separate the issues into a power module, but a WESS platform that is unaware of the state of its power subsystem will operate in energy-inefficient regions most of the time. This section discusses a number of issues related to supply-side power management for WESS platforms. We first review the non-ideal features with power sources and ways to improve their efficiency. Then, we state the power fragmentation problem in multi-supply systems and a power-routing switch as a solution. We also discuss the emerging trend of using supercapacitors as an increasingly important element in the power subsystems of WESSs.

A. *Maximum Power Point Tracking (MPPT)*

WESSs are powered by mostly non-ideal sources. The problem becomes even more complex with other energy sources. Ambient sources such as solar and wind tend to have a much wider dynamic range and unpredictable availability. Even with perfect predictability, a solar panel has a wide range of impedance values. *Impedance mismatch* wastes available power by drawing current at a suboptimal level. If not impedance matched, then the system can easily get <1 W of power even though up to 3W may be available at the given light intensity. Impedance matching can be accomplished by

a combination of a power distribution switch (§IV-B) and dynamic power management. For a given light intensity of a solar panel, the voltage/current level that maximize the output power is called the *maximum power point* (MPP).

Numerous methods proposed to date for tracking the solar cell include the hill climbing method, short-circuit method, and open-circuit voltage method [37]–[40]. The hill climbing method of sweeping solar voltage while measuring the current requires a great deal of circuitry in the form of a DSP, FPGA, or MCU to calculate the MPP. The power needed to run those chips is too high to keep the converter efficient. The short circuit method entails shorting the solar cell and measuring the short circuit current, which directly determines the MPP [41]. Finally, the open-circuit voltage method simply requires disconnecting the solar cell from any load and measuring the open-circuit solar voltage, which again is directly related to the MPP [42], [43]. The latter two methods are not as accurate as the hill climbing method, but the complexity and overhead power make the former inefficient.

Once the MPP is found, then the problem becomes how to make the system draw the optimal current. One way is to keep the system's duty cycle the same while varying the power drawn from the solar panel or battery [44]. Another approach is to vary the system's duty cycle [45] or algorithms [46], to the extent allowed by the application.

B. Systems with Multiple Power Sources

Some sensor nodes now include multiple power sources to replenish the charge over time. They include solar panels, windmills, and other energy scavenging mechanisms. For instance, ZebraNet [4] contains a solar array (Unisolar USF5) that generates up to 5W, in addition to 14 Sony Li-ion polymer cells. It requires this much power mainly for long distance communication, even though the duty cycle is relatively low. Helimote [47] uses two solar cells to charge NiMH batteries in an attempt to sustain operation eternally, for the purpose of battery sizing, and for steering power-aware scheduling. The PicoBeacon [24] 1.9GHz radio transmitter operates on energy scavenged from a photovoltaic and a piezoelectric generator. Vibration to electric energy converters based on MEMS can generate up to $8\mu\text{W}$ of power [48], although it will be some time before high duty-cycle sensor nodes can take advantage of such technologies. Thermoelectric converters have also been demonstrated although they tend to require a large ($> 50^\circ\text{C}$) temperature differential. Among these sources, windmills and vibration-based energy harvesters output AC power, which normally require rectifiers for conversion into DC power.

In all cases, a new problem that arises in multiple power source (MPS) systems is *power fragmentation*. Not all sources such as solar and wind may be available at all times; when they are, they might not be powerful enough to drive the entire system. Furthermore, it is not possible to “add” power from these weak sources to make a strong source. When the ambient sources cannot supply sufficient power, then the rechargeable battery must provide the power, and the ambient power must be discarded. Such wasted power can be utilized if

the system is partitioned into subsystems that require a much lower supply threshold each. Partitioning the power supplies between sensors and digital parts can also help reduce noise.

One way to reduce power fragmentation is a combination of a power distribution switch and a source-consumption matching algorithm that maximizes the total utility of the available power from these ambient power sources [44]. The switch contains power transistors that enable each power source to be connected to a set of power consumers. A sensor on each power source reports the current supply condition and the algorithm finds a connection scheme between subsystems and power sources to maximize power utility. This scheme has been shown to reclaim 25%–50% of the power that would be wasted by conventional designs. By making more efficient use of the available power, the system can be designed to be more compact, operate for longer time, and even slow down the aging process of batteries due to frequent recharge cycles.

C. Supercapacitors

The battery is the primary limiting factor that prevents the node from operating maintenance-free for more than several years at non-trivial data rates. One solution that slows down battery aging is to place a supercapacitor in parallel with the battery so that transient power is delivered by the capacitor rather than the battery [49]–[51]. Fewer and shorter current pulses drawn from the battery allow more efficient use of battery capacity and increase the number of charge cycles possible. Supercapacitors have received wide attention recently due to their power density, low equivalent series resistance (ESR), and very low leakage current [52]. A typical supercapacitor offers more than half a million charge cycles and a 10-year operational lifetime until the capacitance is reduced by 20%. One could not simply replace a battery with a supercapacitor because of the very different electrical characteristics and efficiency considerations. Prometheus [45] uses two 22F supercapacitors as the primary buffer charged by the solar panel, and a lithium polymer battery as the secondary buffer, and the charging is under software control. However, it is possible to do better by eliminating the battery entirely. Given that a 350F capacitor with a capacity of 240 mAh from Maxwell Technologies costs \$30 in single quantity, supercapacitors are also becoming competitive with rechargeable Li-ion batteries.

There has been little prior work on efficiently charging a capacitor using optimal solar power. If placed in supply rails, a capacitor in the hundreds of farads will appear as a short, bringing the supply rail voltage down to the capacitor voltage. When attempting to charge a supercapacitor with a solar cell, it will not do so efficiently, because it will not be at the MPP (§IV-A). A typical voltage regulator is not suitable for charging the supercapacitor, because the feedback voltage will signal to the regulator that the output voltage is too low and short the input to the output. Pulse power applications use resonant converters to efficiently and quickly charge capacitors [53], [54], but they require an AC voltage that cannot be generated by the solar panel, although they can work with

a windmill or vibration-based generators that output AC. An MCU-controlled, feed-forward, pulse frequency modulated regulator that charges the supercapacitor at the MPP for solar cells has been demonstrated [55], enabling uninterrupted, high duty-cycle operation without a battery.

V. METRICS AND QUANTITATIVE EVALUATION

Benchmarks and metrics for quantitative evaluation are essential in advancing an engineering field. End users may be concerned about the *sensitivity* (high true-positives) and *selectivity* (low false-positives) of the sensing system. For system designers, the metrics for a WESS should be an indicator for its *fitness* to the application under consideration. For instance, some WESSs may be more fit for data acquisition while others for event detection, where fitness may be defined in terms of power efficiency, latency and throughput, data noise and jitter, etc. Each metric in turn may be further qualified by if it is solar powered and how well it performs MPPT; by the event density; and by the wireless interference. However, another problem is that, even if a set of benchmarks were available, it would still be difficult to reproduce environmental inputs to test these WESSs. To accomplish this, we describe an emulation-driven benchmarking approach.

A. Fitness Metrics

For each application A , each metric m_i defines a measurable quantity $c_i^A(W)$ of the subject WESS W and $c_i^A(I)$ of an ideal WESS I . The fitness f_i^A of W with respect to metric m_i is the ratio to the ideal (or a reference platform):

$$f_i^A(W) = \frac{c_i^A(W)}{c_i^A(I)} \quad (1)$$

the overall fitness is therefore the *geometric mean* of the respective fitness functions:

$$f^A(W) = \sqrt[N]{\prod_{i=1}^N f_i^A(W)} \quad (2)$$

Different metrics can be weighted by simply raising the ratio to the power of the weight. The fitness function has a range $0 \leq f \leq 1$ where 1 is ideal, and 0 means the WESS fails to perform what is required by the application, including size, missing deadlines, exceeding the max power constraint, etc. As new applications arise, additional metrics can be added.

B. Emulation-Driven Benchmarking

We believe the only realistic way to evaluate WESS architectures is by measurement on the actual system. Estimates based on datasheet figures of the components can be very misleading. For instance, we have found that the measured battery lifetime of a sensor node to be only 26–35% if it does not manage its RF power carefully. Although simulators exist, the only way to consider all effects is to run the WESS in the intended operating environment. Unfortunately, this poses several challenges. Unlike SPEC benchmark programs, WESSs cannot read inputs (signals to sense, ambient power profile, etc) from a file, and it is difficult to exactly reproduce

all these environmental conditions that comprise the input to WESSs. Even if possible, there is the practical issue of duration: some WESSs are designed for months or years of operation until the battery is fully drained. It would be impractical then to test these systems for so long.

To address these problems with benchmarking WESSs, we propose an *emulation-driven measurement* technique. The idea is to collect all relevant environmental profiles such as solar, wind, and signals of interest, save their digitized representation in a file, and “play back” these conditions in real time for the WESS under test, by supplying the power to the system; for other signals we may either re-create the physical conditions for sensing purposes or just provide the electrical form of the signal. Such a framework is being built by extending our existing battery emulator and power profiler called B# (B-sharp) [56]. B# is a battery emulator and a power profiling instrument. It emulates the behavior of a battery by measuring the current load, calling a battery simulation program in real time to compute the voltage response, and controlling a linear regulator to mimic the voltage output of a battery. B# gives users controllability and reproducibility: they have complete control over the charge state of the battery by charging this virtual battery instantly or set it to any arbitrary charge state – fully charged, half discharged, near depletion – all by simply changing a few bits in the simulation program. We extended it with a model for solar panels, and we plan to add models for wind generator and fuel cells. B# can be extended to support emulation of other non-power events by simply adding more digital-to-analog converters synchronized to the same clock.

VI. CONCLUSIONS

This tutorial highlights issues to consider when choosing an existing WESS platform or building one’s own. For the design automation community, opportunities are in system-level design tools that produce WESS platforms with high affinity to the specific applications. The recent trend towards modular platforms indicates the need for more customization by composing reusable blocks, but for deployment, they must be much more tightly integrated on-board or on-chip. Also needed are tools that aid integration of WESS platforms with each other to form a network or with an existing infrastructure. Many challenges remain in the cross-disciplinary optimization. As energy efficiency underscores all aspects of a WESS platform design, optimization techniques now span both sides of power consumption and supply. Ultimately, these innovations must be evaluated together *by measurement* as an entire system, rather than citing datasheet figures. We believe a well-established quantitative evaluation methodology of whole systems by measurement will be crucial in measuring true progress of WESS platform designs as a field.

Acknowledgments: This work is sponsored by NSF CCR-0205712, CMS-0509018, and NSF CAREER CNS-0448668. We thank Jinfeng Liu, Jiwon Hahn, and Farhan Simjee for their input.

REFERENCES

- [1] “MPR400/410/420 MICA2 Mote,” <http://www.xbow.com/Products/productsdetails.aspx?sid=72>.

- [2] "Stargate X-Scale processor platform," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0049-01_B.STARGATE.pdf.
- [3] "PASTA: Power aware sensing tracking and analysis," <http://pasta.east.isi.edu/>.
- [4] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proc. ASPLOS*. ACM Press, 2002, pp. 96–107.
- [5] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *CACM*, vol. 47, no. 6, pp. 34–40, Jun. 2004.
- [6] H. Chung, C. Park, Q. Xie, P. H. Chou, and M. Shinozuka, "DuraNode: wireless network sensor for structural safety monitoring," in *SPIE Nondestructive Detection and Measurement for Homeland Security III Conference*, Mar. 2005.
- [7] C. Park, J. Liu, and P. H. Chou, "Eco: an ultra-compact low-power wireless sensor node for real-time motion monitoring," in *Proc. IPSN*, May 2005.
- [8] B. J. Tromberg, N. Shah, R. Lanning, A. Cerussi, J. Espinoza, T. Pham, L. Svaasand, and J. Butler, "Non-invasive *In Vivo* characterization of breast tumors using photon migration spectroscopy," *Neoplasia*, vol. 2, no. 1–2, pp. 26–40, January–April 2000.
- [9] T. H. Pham, O. Coquoz, J. B. Fishkin, E. Anderson, and B. Tromberg, "Broad bandwidth frequency domain instrument for quantitative tissue optical spectroscopy," *Review of Scientific Instruments*, vol. 71, no. 6, pp. 2500–2513, June 2000.
- [10] K.-S. No and P. H. Chou, "Mini-FDPM and heterodyne Mini-FDPM: Handheld non-invasive breast cancer detectors based on frequency domain photon migration," *IEEE Transactions on Circuits and Systems*, no. 11, November 2005.
- [11] "WSN wireless sensing networks," <http://wins.rsc.rockwell.com>.
- [12] D. D. Wentzloff, B. H. Calhoun, R. Min, A. Wang, N. Ickes, and A. P. Chandrakasan, "Design consideration for next generation wireless power-aware microsensor nodes," in *Proc. VLSI'04*, 2004, pp. 361–367.
- [13] B. Schott, M. Bajura, J. Czarnaski, J. Flidr, T. Tho, and L. Wang, "A modular power-aware microsensor with >1000X dynamic power range," in *Proc. IPSN*, May 2005.
- [14] M. Amundson, "Smart kindergarten enhances childhood learning iBadge tracks, analyzes interactions in classroom," *UCLA Engineering - Spring 2002*, 2002.
- [15] <http://www.tinyos.net>.
- [16] "BTnode system software," http://www.btnode.ethz.ch/support/btn_api/index.html.
- [17] "ATMEL ATmega128 microcontroller," http://www.atmel.com/dyn/products/product_card.asp?part_id=2018.
- [18] "MSP430F G437," <http://www.ti.com/sc04084>.
- [19] <http://www.moteiv.com/>.
- [20] "nRF24E1 transceiver / MCU / ADC," <http://www.nvlsi.com/index.cfm?obj=product&act=display&pro=79>.
- [21] "AT86RF210 Z-Link transceiver," http://www.atmel.com/dyn/resources/prod_documents/doc5033.pdf.
- [22] http://www.atmel.com/dyn/resources/prod_documents/DOC1743.PDF.
- [23] J. Lifton, M. Broxton, and J. A. Paradiso, "Experiences and directions in Pushpin computing," in *Proc. IPSN*, May 2005.
- [24] S. Roundy, B. P. Otis, Y.-H. Chee, J. M. Rabaey, and P. Wright, "A 1.9GHz RF transmit beacon using environmentally scavenged energy," in *Proc. ISLPED*, 2003.
- [25] "Section 15.231 periodic operation in the band 40.66–40.70 MHz and above 70 MHz," http://a257.g.akamaitech.net/7/257/2422/05dec20031700/edocket.access.gpo.gov/cfr_2003/octqtr/pdf/47cfr15.227.pdf, p. 745.
- [26] Office of Engineering and Technology, Federal Communications Commission, "Understanding the FCC regulations for low-power, non-licensed transmitters," http://www.fcc.gov/Bureaus/Engineering_Technology/Documents/bulletins/oet63/oet63rev.pdf, February 1996.
- [27] "BTnodes - a distributed environment for prototyping ad hoc networks," <http://www.btnode.ethz.ch/>.
- [28] M. Leopold, M. B. Dydensborg, and P. Bonnet, "Bluetooth and sensor networks: A reality check," in *Proc. SenSys'03*, November 2003, pp. 103–113.
- [29] "ZigBee Alliance," <http://www.zigbee.org/>.
- [30] "Heterogeneous sensor networks," <http://www.intel.com/research/exploratory/heterogeneous.htm>.
- [31] C. Park, K. Lahiri, and A. Raghunathan, "Battery discharge characteristics of wireless sensor nodes: An experimental analysis," in *IEEE Conf. on Sensor and Ad-hoc Communications and Networks (SECON)*, Sep. 2005.
- [32] "AN9520 multilayer chip antenna for 2.4GHz wireless communication," <http://www.rainsun.com/web-download/AN9520.pdf>, 2004.
- [33] "AN3216 multilayer chip antenna for 2.4GHz wireless communication," <http://www.rainsun.com/Products/LTCC/AN3216/AN3216.pdf>, 2004.
- [34] A. Y. Benbasat and J. A. Paradiso, "A compact modular wireless sensor platform," in *Proc. IPSN*, May 2005.
- [35] M. Doyle, T. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of the Electrochemical Society*, vol. 140, no. 6, pp. 1526–1533, June 1993.
- [36] T. Fuller, M. Doyle, and J. Newman, "Relaxation phenomena in lithium-ion insertion cells," *Journal of the Electrochemical Society*, vol. 141, pp. 982–990, 1994.
- [37] K. Hussein, I. Muta, T. Hoshino, and M. Osakada, "Maximum photovoltaic power tracking: an algorithm for rapidly changing atmospheric conditions," in *IEE Proceeding Generation, Transmission and Distribution*, Jan. 1995, pp. 59–64.
- [38] C. Hua and C. Shen, "Study of maximum power tracking techniques and control of DC/DC converters for photovoltaic power system," in *IEEE-PESC. Conf. Rec.*, 1998, pp. 86–93.
- [39] E. Koutroulis, K. Kalaitzakis, and N. Voulgaris, "Development of a microcontroller-based, photovoltaic maximum power point tracking control system," *IEEE Trans. Power Electronics*, vol. 16, pp. 46–54, Jan. 2001.
- [40] Y. Kuo, T. Liang, and J. Chen, "Novel maximum-power-point-tracking controller for photovoltaic energy conversion system," *IEEE Trans. Ind. Electronics*, vol. 48, pp. 594–601, June 2001.
- [41] T. Noguchi, S. Togashi, and R. Nakamoto, "Short-current pulse-based maximum-power-point tracking method for multiple photovoltaic-and-converter module system," *IEEE Trans. Ind. Electronics*, vol. 49, pp. 217–223, Feb. 2002.
- [42] J. Enslin, M. Wolf, D. Snyman, and W. Swiegers, "Integrated photovoltaic maximum power point tracking converter," *IEEE Trans. Ind. Electronics*, vol. 44, pp. 769–773, Dec. 1997.
- [43] D.-Y. Lee, H.-J. Noh, D.-S. Hyun, and I. Choy, "An improved MPPT converter using current compensation method for small scale pv-applications," in *IEEE 18th Annual Applied Power Electronics Conf. and Expo.*, 2003.
- [44] C. Park and P. H. Chou, "Power utility maximization for multiple-supply systems by a load-matching switch," in *Proc. ISLPED*, August 2004.
- [45] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proc. IPSN*, May 2005.
- [46] D. Li and P. H. Chou, "Maximizing efficiency of solar-powered systems by load matching," in *Proc. ISLPED*, August 2004.
- [47] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *SIGMETRICS/Performance'04*, 2004, pp. 223–234.
- [48] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, A. P. Chandrakasan, and J. H. Lang, "Vibration-to-electric energy conversion," *IEEE Transactions on VLSI Systems*, vol. 9, no. 1, pp. 64–76, February 2001.
- [49] P. Enjeti, J. Howze, and L. Palma, "An approach to improve battery runtime in mobile applications with supercapacitors," in *IEEE 34th Annual Power Electronics Specialists Conference*, 2003.
- [50] T. Smith, J. Mars, and G. Turner, "Using supercapacitors to improve battery performance," in *Power Electronics Specialists Conference*, 2002.
- [51] L. Gao, R. Dougal, and S. Liu, "Active power sharing in hybrid battery/capacitor power sources," in *Eighteenth Annual IEEE Applied Power Electronics Conference and Exposition*, 2003.
- [52] Maxwell Technologies, "BCAP0350 datasheet," <http://www.maxwell.com>.
- [53] H. Pollock, "High efficiency, high frequency power supplies for capacitor and battery charging," in *IEE Colloquium on Power Electronics for Demanding Applications*, Apr. 1999, pp. 901 – 910.
- [54] R. Nelms and J. Schatz, "A capacitor charging power supply utilizing a ward converter," *IEEE Trans. Ind. Electronics*, vol. 39, pp. 421–428, Oct. 1992.
- [55] F. Simjee and P. H. Chou, "Everlast: Long-life, supercapacitor-operated wireless sensor node," University of California, Irvine, Tech. Rep., 2005.
- [56] C. Park, J. Liu, and P. H. Chou, "B#: a battery emulator and power profiling instrument," *IEEE Design and Test of Computers*, vol. March-April, pp. 150–159, 2005.