

Research Article

An Efficient Renewable Energy Management and Sharing System for Sustainable Embedded Devices

Ting Zhu¹ and Chang Zhou²

¹Department of Computer Science, State University of New York, Binghamton, NY 13905, USA

²College of Information Engineering, China Jiliang University, Hangzhou, Zhejiang 310018, China

Correspondence should be addressed to Ting Zhu, tzhu@binghamton.edu

Received 23 June 2011; Revised 26 October 2011; Accepted 6 November 2011

Academic Editor: Qinru Qiu

Copyright © 2012 T. Zhu and C. Zhou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present the design and evaluation of a capacitor-driven efficient renewable energy management and sharing system, called REMS system, that is designed to replace battery-based energy storage units. We first identify potential applications and requirements of the REMS system. Then we describe the design of REMS system which aims at minimizing the energy leaked away and providing accurate and fine-grained energy sharing between embedded devices. We extensively evaluate our system under different real-world settings. Results indicate that our charging and discharging control can effectively minimize the energy leaked away. Moreover, the energy sharing design can efficiently share renewable energy among multiple devices and significantly extend the lifetime of a system.

1. Introduction

With the increasing need for ubiquitous computing, embedded devices are heavily involved in a lot of applications such as data collection [1, 2], weather condition monitoring [3], road service monitoring [4], online game [5], VoIP service [6], mobile payment [7], and social activities [8]. Due to the mobility requirement, embedded devices are normally equipped with limited and rechargeable energy storage units, such as rechargeable batteries.

However, the electro-chemical conversion inside the rechargeable batteries during the charging stage significantly reduces the energy efficiency to as low as 6% [9]. Moreover, rechargeable batteries have limited recharge cycles due to cyclic memory and crystalline formation (e.g., a Li-ion battery has 500 cycles and NiMH 300 cycles), which translates to a lifetime of 2 to 3 years. Given the huge amount of the embedded devices on the market, the relative short lifetime of rechargeable batteries results in a huge amount of disposed batteries each year. The chemical materials (e.g., nickel, mercury, etc.) inside these batteries can cause very serious environmental problem. Moreover, the innovation in lithium-ion battery capacity has essentially stalled for more

than 4 years, with capacity improvement increasing by less than 5% per year [10].

Compared with rechargeable batteries, ultracapacitors possess a set of advantages: they (i) have high charge efficiency (i.e., more than 90%); (ii) have more than 1 million recharge cycles which translates to a lifetime of more than 10 years; (iii) can be charged very fast; (iv) are made from carbon nanotubes which does not have toxic-waste-disposal problems when the capacitors reach the end of their lifetime. Furthermore, recent advances in ultracapacitor technology make it possible to use ultracapacitors as the *only* energy storage device. For instances, the research groups at MIT [11] and the University of Maryland [12] have announced nanotube-based ultracapacitors, which can provide energy storage densities comparable to those of batteries. In 2006, a US patent [13] was issued for an electrical energy storage unit using an ultracapacitor that has an energy/weight value of about 342 W·h/kg, twice that of Li-ion batteries. The largest capacitance available on the market increased from 3,000 F [14] to 5,000 F [15] in 2009. Powered by a 5,000 F capacitor, a mobile phone can work for more than 11 days under standby mode with a single initial charge of the capacitor. It is highly possible that in the near future we will witness a paradigm

shift from a battery-based to an ultracapacitor-based design for all kinds of embedded devices. Therefore, it is essential to explore this frontier in advance.

To utilize the aforementioned advantages of ultracapacitors, we designed a capacitor-driven renewable energy management and sharing system, called *REMS* system. The system can be charged within seconds and have high charge efficiency. Moreover, the systems can conduct fine-grained and efficient energy sharing among themselves. Given limited energy inside each individual embedded device, we believe that energy sharing between embedded devices to efficiently utilize the energy is the future trend. On October 22, 2009, the International Telecommunication Union announced a standard for a universal charger for mobile handsets [16], which provides a general hardware interface for applying our energy sharing approach.

In summary, our contributions are as follows.

- (i) To our knowledge, energy sharing is a new concept that can effectively utilize the limited energy and improve the performance of multiple embedded devices as a whole. With fine-grained and efficient energy sharing, energy can quantitatively flow back and forth from one device to another device.
- (ii) Our hardware and control layer design is simple and general. It can easily be extended to support n capacitors. Furthermore, our design affords simultaneous energy sharing among multiple embedded devices.
- (iii) We evaluate our system extensively in multiple real-world testbeds and simulation. The results indicate that our system can effectively minimize the energy leaked away and provide accurate and efficient energy sharing among multiple systems.

The paper is organized as follows. Section 2 examines why energy sharing is necessary for embedded devices and discusses the need and the issue for capacitor-driven energy storage system. Section 3 gives an overview of the system architecture. Hardware design, control and network layers are presented in Sections 4, 5 and 6, respectively. System implementation and evaluation are detailed in Sections 7 and 8. Related work is discussed in Section 9. Finally, Section 10 concludes the paper.

2. Motivation

Although ultracapacitors have some advantages over rechargeable batteries, they also have a major limitation (i.e., leakage) which introduce a design challenge for capacitor-driven energy storage system. In this section, we (i) investigate the leakage property of ultracapacitors, (ii) show the benefit of the capacitor-driven design by studying two real-world scenarios, and (iii) discuss the requirements of enabling such benefit.

2.1. The Challenge for Capacitor-Driven Design. Compared with rechargeable batteries, ultracapacitors have higher leakage current which is only crudely specified in the data sheets and introduce a challenge to the system design. To

further investigate the leakage property of the ultracapacitor, we conducted experiments over a period of 6 weeks using ten types of ultracapacitors, ranging from 22 F to 3000 F. After an ultracapacitor was fully charged, we isolated it and continuously monitored its remaining voltage over 1000 hours.

Figure 1(a) shows that leakage is more severe for larger ultracapacitors than for smaller ultracapacitors. For example, after 1000 hours, the energy leaked away inside a 3,000 F capacitor is about 83 times of the energy leaked away inside a 22 F. Figure 1(a) also reveals that the ultracapacitors have very large amount of energy (e.g., 22.9% of total energy for 3000 F) leaked away during the first 10 hours, but little energy (e.g., 21.7% for 3000 F) leaked away during the rest 990 hours. These results indicate that ultracapacitors have leakage performance that is comparable to that of batteries (e.g., 15% per month for NiMH battery [17]) when voltage is controlled under an appropriate level.

However, ultracapacitors suffer severe energy leakage when it is charged to the limit. We investigated the correlation between the voltage and leakage of ultracapacitors. Figures 1(b) and 1(c) show that when the voltage of ultracapacitors approaches its limit, the value of leakage power increases significantly. For example, when the voltage of the 3000F ultracapacitor increases to 2.7 V (shown in Figure 1(c)), its leakage power is 145.2 mW, which is 5 times of the power needed for powering a Cingular 2125 smartphone to work under GSM Idle scenario [6].

Our empirical study concludes that (i) ultracapacitor is a good substitute for battery and (ii) energy leakage in ultracapacitors should not be overlooked.

2.2. The Benefit for Capacitor-Driven Design. Except the leakage issue, capacitor-driven energy storage system enables a new style of energy management: *energy sharing* which allows energy flow back and forth among multiple capacitors which belong to different energy storage systems. Energy sharing will be of benefit to a lot of applications, such as wearable computing [18, 19], portable weather monitoring system [3], and robot-sensor networks [1, 20, 21]. Due to the space constraint, we only study two real-world scenarios in this section.

Scenario 1. In the wearable computing application, Dr. Abowd's team [22] has empirically studied the expected harvesting power from six body locations. Their results indicate that different devices have significantly different harvesting power (e.g., $115 \pm 106 \mu\text{W}$, $1.01 \pm 0.46 \text{ mW}$, and $4.9 \pm 3.63 \text{ mW}$ for the wristwatch, the cell phone, and the shoe, resp.). Moreover, the harvesting power varies over time.

In this scenario, we see the temporal and spatial diversity of the harvesting power. However, the energy harvested by these devices may not meet their energy consumption requirement. For example, the cell phone may have the highest energy consumption, while the shoe may have the lowest energy consumption. If we enable energy sharing among these devices, the energy generated from the shoe can be utilized to power the cell phone. Moreover, the

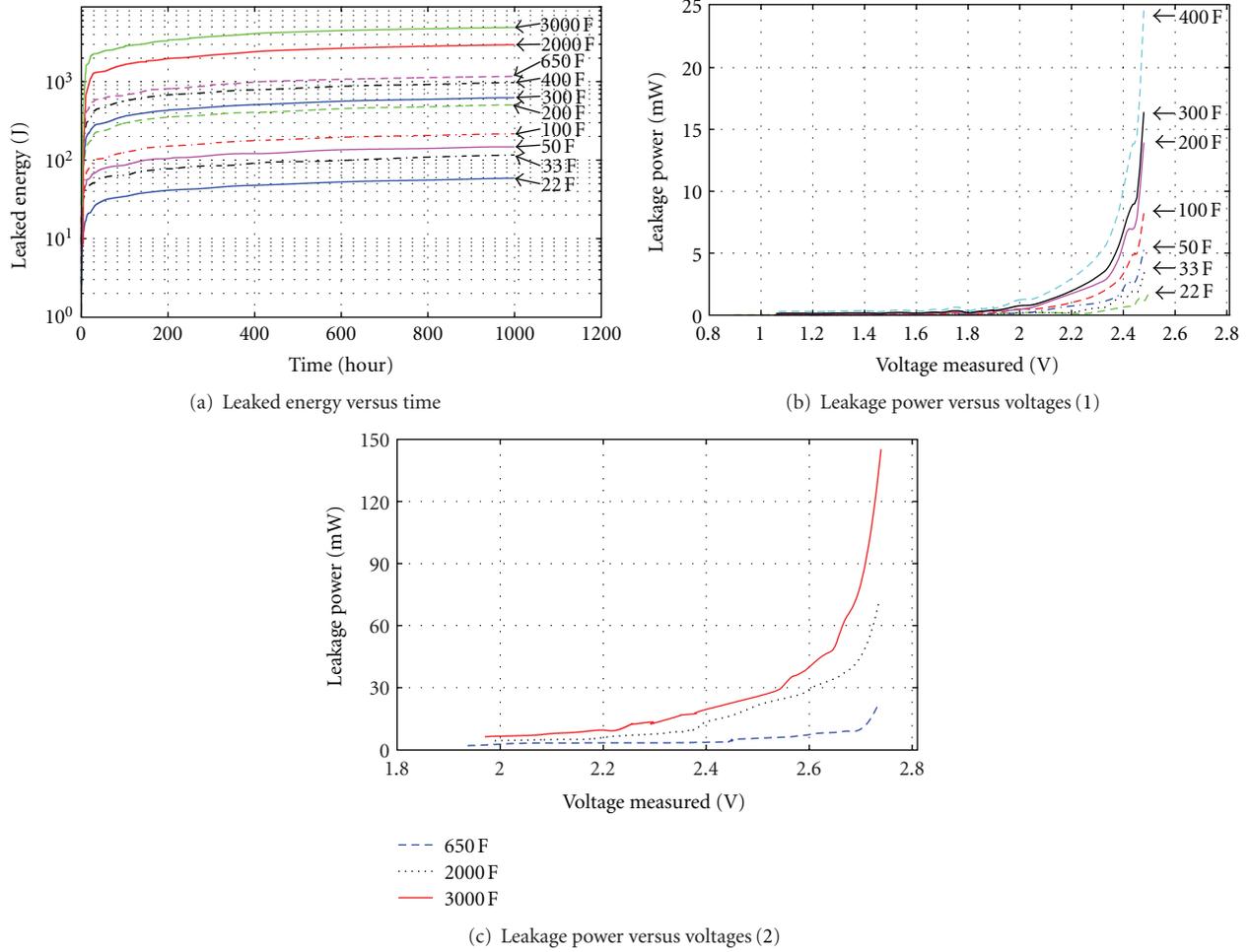


FIGURE 1: Leakage property of ultracapacitor.

temporal diversity of the harvesting power also requires the wristwatch to share energy with the cell phone during the period when only the wristwatch can harvest energy. Given the low harvesting power, efficient and fine-grained energy sharing is needed to maintain the aliveness of all the devices.

Scenario 2. The Plant Care project [23] is an investigation of practical ubiquitous systems. In this system, a robot is employed to water the plants and to calibrate and recharge deployed sensors. To achieve sustainability, a power-line is applied to charge the robot periodically.

In this scenario, if the sensors and the robot are equipped with energy harvesting circuits (e.g., solar panels or wind generators), the system will keep on working even when a winter storm knocked out the power-line. Due to the location constraint, different sensors may have different harvesting power. If the energy can be shared among the sensors and the robot, the performance of the whole system can be balanced. Since the robot needs to interact with a lot of sensors during the dry season, the time duration of the energy sharing between the robot and sensors should be short.

The two scenarios discussed above illustrate the benefit of energy sharing feature and also indicate the requirements of the energy sharing: (i) highly efficient, (ii) fast, and (iii) fine-grained. Theoretically, the energy sharing concept can be applied to the battery-based systems, but the performance of these systems is significantly affected by the limitations of batteries: (i) low charge efficiency (as low as 6% [9]), (ii) limited charge current, and (iii) inaccurate remaining energy prediction which results in coarse granularity of energy sharing. In contrast, capacitors do not have all of the above limitations of the batteries, thus they meet the requirements of the energy sharing.

3. System Overview

The empirical study in Section 2 indicates that energy leakage is a critical hurdle that hinders designers exploiting the benefit of ultracapacitor. To address this hurdle, we propose a three-layer design as shown in Figure 2. The design goal of our system is to minimize the energy leaked away and efficiently utilize and share the energy among multiple embedded devices which form a network.

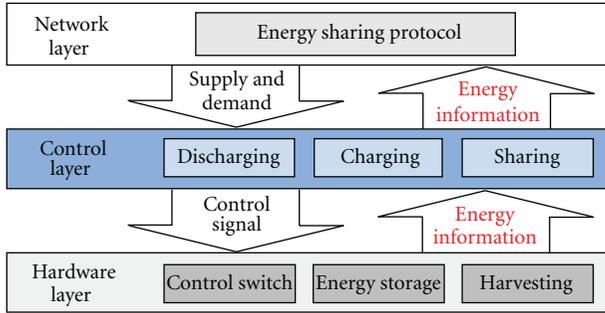


FIGURE 2: Overview of system architecture.

- (1) The hardware layer uses (i) multiple ultracapacitors to form a capacitor array as the *only* energy storage unit to power the system, and (ii) a series of control switches to control different types of connections between the ultracapacitors (i.e., in parallel, serial, or hybrid). To achieve fast and efficient energy sharing, we use N-MOSFETs as switches which support large current, low energy consumption, and extremely short switching time (in the level of *ns*). This layer also provides an energy harvesting circuit (e.g., solar panel or wind generator) to harvest the environmental energy. Moreover, this layer (i) monitors the remaining energy inside each ultracapacitor and (ii) samples the harvesting power from the energy harvesting circuit. These two pieces of information are then fed into the control layer.
- (2) The control layer serves as a bridge between the hardware layer and network layer. It plays two important roles: (i) it calculates the energy leakage rate online based on the remaining energy inside each ultracapacitor and then forwards the leakage information (together with the remaining and harvesting energy) to the network layer; (ii) based on the leakage model and the energy supply and demand request from the network layer, it controls the working state (i.e., discharging, charging, or energy sharing) of the energy storage system by sending the control signal to the hardware layer to control the on/off state of the control switches. Since the control layer has all of the energy information (including leakage information), it can conduct fine-grained control (including energy sharing).
- (3) In the network layer, a running energy sharing protocol needs to effectively utilize the energy among multiple devices. In each embedded device, the energy sharing protocol calculates the difference between the actually available energy and the expected required energy by using the energy information forwarded from the control layer. Based on the energy difference, the energy sharing protocol decides the amount of energy needed to be shared among embedded devices and then determines the working state of the energy storage system by sending the quantitative

energy supply and demand request to the control layer.

4. Hardware Layer

The design goal of the hardware layer is to provide a general, simple, and stable energy storage system with minimum leakage. To achieve this design goal, we designed the capacitor array circuit and built the hardware platform (i.e., Energy Mule).

4.1. Capacitor Array Circuit Design. A straightforward choice for energy storage is to use a single large ultracapacitor. However, this suffers from a slow boot-up time (to reach operational voltage), high remaining energy (detailed in Section 5.3), and inflexibility in fine-grained control (due to the limited granularity of the A/D converter inside sensor devices). For example, MICAz uses a 10-bit A/D converter. When being charged or discharged with the same amount of energy, a single large capacitor's voltage value does not change as significantly as a small capacitor, which causes inaccurate charging and discharging control (detailed in Sections 5.2 and 5.3). In our design, we build the Energy Mule using an array of ultracapacitors that have different levels of capacitances. This capacitor array is designed to satisfy the following three requirements.

(i) *Generality.* The circuit should be easily extended to n capacitors. Moreover, the capacitors can be connected in different types of connections (e.g., series, parallel, or hybrid) by using this circuit.

(ii) *Simplicity.* It should provide minimum number of control interface to the embedded device.

(iii) *Stability.* It needs to have stable and simultaneous charging and discharging capability. In other words, during the charging state the circuit still needs to output energy to power the embedded device. Moreover, it is important to stably and smoothly change the connection type (e.g., from parallel to series) of the capacitors during the charging/discharging state.

The circuit we designed successfully achieves the above three design goals. Figure 3(a) shows the circuit for two capacitors which can be easily extended to support n capacitors (shown in Figure 3(b)). Moreover, this circuit can achieve different combinations of connections (shown in Table 1). Here we use 0/1 to indicate the on/off state of the switches. All of these switches can be controlled by an embedded device through the serial-to-parallel control signal converter (shown in Figure 4) which introduces minimum number of control interface to the embedded device. The output of the circuit is connected to a step-up DC/DC converter. Since a DC/DC converter normally has a very wide input voltage range (e.g., 0.7V \sim 5V for MAX1676), our circuit can still provide a constant voltage from the output of the DC/DC converter when the capacitors are changing their connections (e.g., from parallel to series).

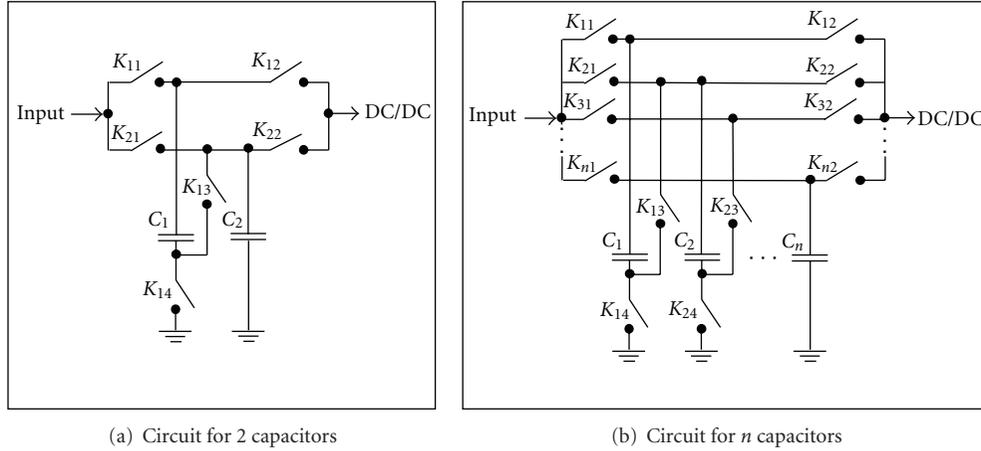


FIGURE 3: Capacitor array circuit.

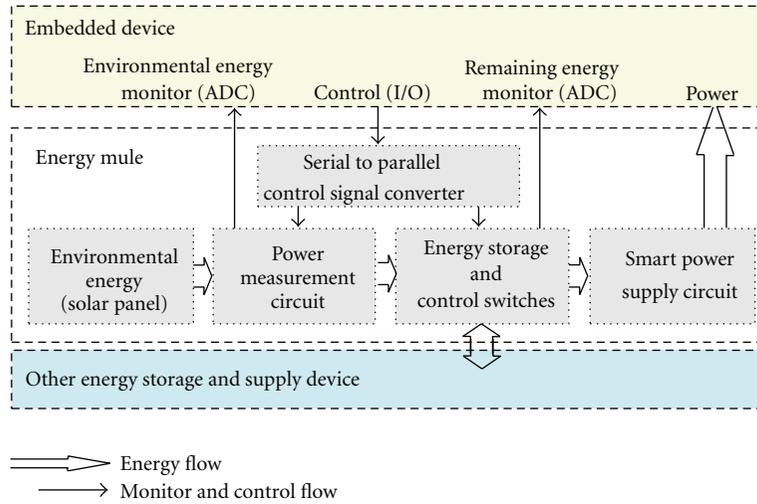


FIGURE 4: Energy mule hardware architecture.

4.2. *Hardware Platform.* Energy Mule is an add-on power board that supports multiple ultracapacitors to form an array as the only energy storage unit to overcome the intrinsic limitations of batteries (e.g., energy uncertainty, limited recharge cycles, low conversion efficiency, and environment unfriendliness). The architecture of Energy Mule is shown in Figure 4. It consists of (i) the solar panels and peripheral circuit for energy harvesting; (ii) the harvesting power measurement circuit; (iii) the ultracapacitor-based energy storage and the control switches which can directly share energy with the other energy storage and supply device; (iv) the serial to parallel control signal converter; (v) the smart power supply circuit with a DC/DC converter for powering the embedded device which is attached to the Energy Mule board.

5. Control Layer

By interacting with the hardware layer, the control layer achieves the following two design goals: (i) effectively

controls the energy utilization of a single embedded device to minimize the energy leaked away inside the energy storage system of this device; (ii) efficiently shares energy between the embedded device itself and its neighboring devices while maintaining its own aliveness. In summary, the control layer contains three types of control: (i) charging, (ii) discharging, and (iii) energy sharing. To achieve effective control, leakage model is very important.

5.1. *Energy Leakage Model.* Based on our empirical study in Section 2.1, we formulate a leakage model to characterize the relationship between the voltage in the ultracapacitor and the leakage power. The embedded device builds the leakage model by storing the value of the leakage power P_L and the corresponding capacitor's voltage value V_c in its memory. Figure 5(a) shows the empirical leakage data of the leakage model for diverse capacitors. To express the model mathematically and reduce the storage space for the model, we use a piecewise linear approximation of the leakage curve. We define *turning points* as the points in the curve with

TABLE 1: Different types of connections.

Connection types	K_{11}	K_{12}	K_{13}	K_{14}	K_{21}	K_{22}
Discharge C_1 , Disconnect C_2	0	1	0	1	0	0
Discharge C_1 , Charge C_2	0	1	0	1	1	0
Charge and Discharge C_1 simultaneously	1	1	0	1	0	0
Discharge C_2 , Disconnect C_1	0	0	0	1	0	1
Discharge C_2 , Charge C_1	1	0	0	1	0	1
Charge and Discharge C_2 simultaneously	0	0	0	1	1	1
Discharges C_1, C_2 in parallel	0	1	0	1	0	1
Charge and Discharge C_1, C_2 in parallel	1	1	0	1	1	1
Discharges C_1, C_2 in serial	0	1	1	0	0	0
Charge and Discharge C_1, C_2 in serial	1	1	1	0	1	0

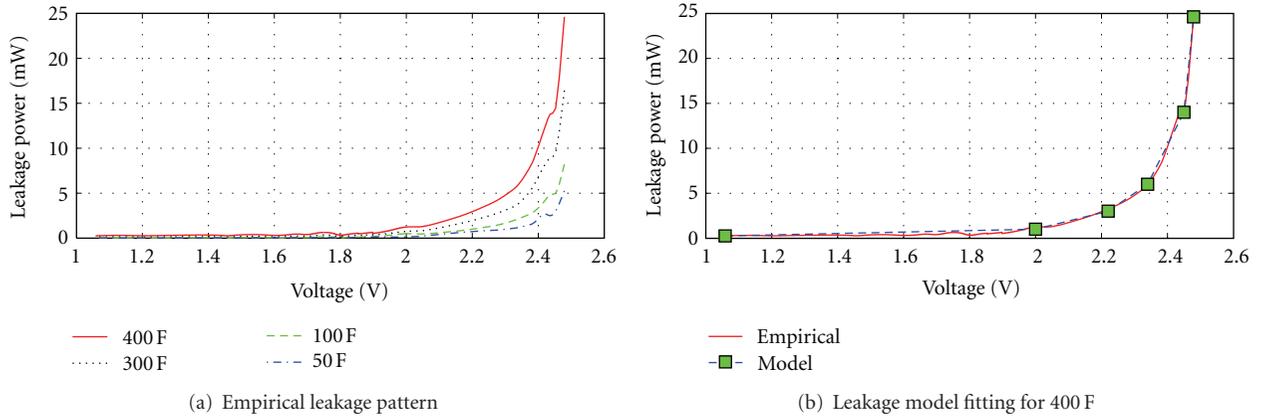


FIGURE 5: Leakage model.

considerable slope change, which are used to decide the start and end points of line segments. Therefore, the whole leakage model can be represented by a set of linear functions, as follows:

$$P_L(V_c) = \begin{cases} a_1 \cdot V_c + b_1; & V_1 \leq V_c < V_2 \\ a_2 \cdot V_c + b_2; & V_2 \leq V_c < V_3 \\ \vdots & \vdots \\ a_n \cdot V_c + b_n; & V_n \leq V_c \leq V_{n+1}. \end{cases} \quad (1)$$

Here V_1, V_2, \dots, V_{n+1} are the remaining energy values corresponding to the turning points of the segments. a_1, a_2, \dots, a_n and b_1, b_2, \dots, b_n are the coefficients for each line segment. Figure 5(b) shows an example for the line-segment-based modeling of the leakage power for a 400F capacitor. The square dots represent the turning points. The model we built accurately matches the empirical results.

5.2. Charging. The design goal of the charging control is to efficiently charge the capacitor array based on the leakage model of each capacitor so that the amount of energy leaked away is minimized. As an example, Figure 6 shows the charging control of two capacitors (C_1 and C_2). Initially, switches K_{11} and K_{21} are turned off. When the charging

control is conducted, the control layer measures the voltage values (V_{C_1} and V_{C_2}) of these two capacitors and calculates the corresponding leakage powers (P_{L_1} and P_{L_2}) based on the leakage model. If P_{L_1} is less than or equal to P_{L_2} (shown in Figure 6(a)), the control layer decides to charge C_1 by turning on K_{11} . In the case that P_{L_1} is greater than P_{L_2} (shown in Figure 6(b)), C_2 is charged. The charging control repeats with time interval T until there is no external energy available or all the capacitors are fully charged.

As an all-purpose energy storage system, it is important that the system can be charged by different energy sources (e.g., environmental energy or DC power supply). However, these diverse energy sources introduce a challenge to the design of charging control, because their charge current varies over four decades (ranging from less than 1 mA for environmental energy to more than 10 A for DC power supply). If the charging control is conducted with a fixed small time interval (in favor of the large charge current), unnecessary voltage measurements and comparisons are conducted when the capacitor is charged with a small current and its voltage increases slowly. Whereas, if the time interval is large (in favor of the small charge current), more energy is leaked away when the charge current is large.

To minimize the energy leaked away and reduce the control overhead, we propose the adaptive charging control

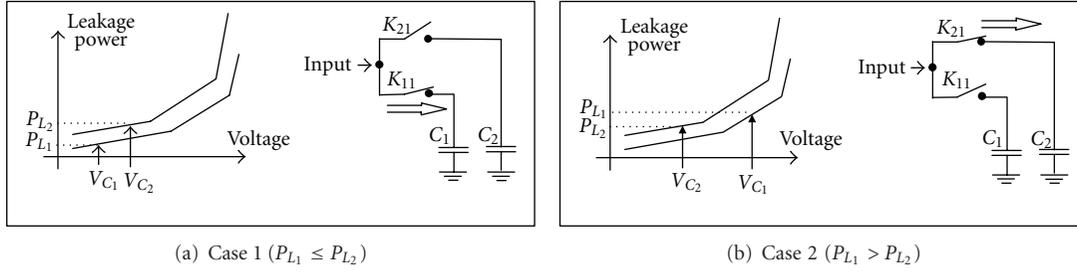


FIGURE 6: Charging control example.

```

(1) for  $i \leftarrow 1$  to  $n$  do
(2)    $K_{i1} \leftarrow 0$ ;
(3)   Calculate  $P_{L_i}$  based on measured  $V_{C_i}$ ;
(4) end
(5)  $m \leftarrow \text{GetMinCapId}()$ ,  $K_{m1} \leftarrow 1$ ;
(6) Set timer with the smallest time interval  $T$ ;
(7) while timer fired do
(8)   for  $i \leftarrow 1$  to  $n$  do
(9)     Measure  $V_{C_i}$  and update  $P_{L_i}$ ;
(10)  end
(11)  if  $m = \text{GetMinCapId}()$  then
(12)     $T \leftarrow 2T$ ;
(13)  else
(14)     $K_{m1} \leftarrow 0$ ,  $m \leftarrow \text{GetMinCapId}()$ ;
(15)     $K_{m1} \leftarrow 1$ ;
(16)  end
(17)  if Capacitors Full/No External Energy then
(18)    break;
(19)  else
(20)    Set timer with interval  $T$ ;
(21)  end
(22)end

```

ALGORITHM 1: Adaptive charging control.

which conducts the charging control operation with an adaptive time interval based on the charge current. Algorithm 1 illustrates the control algorithm for n capacitors. Initially, the controller turns off all the switches and calculates the leakage power of all the capacitors based on the measured voltage values of these capacitors (Lines 1 to 4). Then, the controller turns on the switch which connects to the capacitor with the minimum leakage power and sets the timer with the smallest time interval T (Lines 5 and 6). Since we allow simultaneous charging and discharging, when the timer fired, the controller needs to recalculate the leakage power of all the capacitors based on the newly measured voltage values of these capacitors (Lines 8 to 10). If the capacitor with the minimum leakage power is the same as the one in the previous cycle (indicating the current time interval T is too short), the time interval is doubled (Lines 11 and 12). Otherwise, the controller turns off the switch that connects to the previous capacitor and turns on the switch that connects to the capacitor which currently has the minimum leakage power (Lines 13 to 16). The above control procedure repeats every time interval T , until all of the capacitors are fully

charged or there is no external energy available (Lines 17 to 22).

5.2.1. Complexity and Feasibility Analysis. In the worst case, the timer is fired for k times and the control algorithm measures capacitors' voltage every time the timer is fired. Therefore, the complexity of the control algorithm is $O(kn)$, where n is the number of capacitors. In the algorithm, the smallest time interval T is 100 ms. It will exponentially increase if the capacitor with the minimum leakage power is the same as the one in the previous cycle. The value of time interval T can reach 1 hour within 16 iterations. Therefore, it is feasible for both indoor charging with DC power supply and outdoor charging with energy harvesting devices such as solar panels.

5.3. Discharging. This section discusses how to smartly discharge the capacitors to achieve the following two design goals: (i) the energy leaked away is minimized during the discharge stage, (ii) the remaining energy inside the capacitor

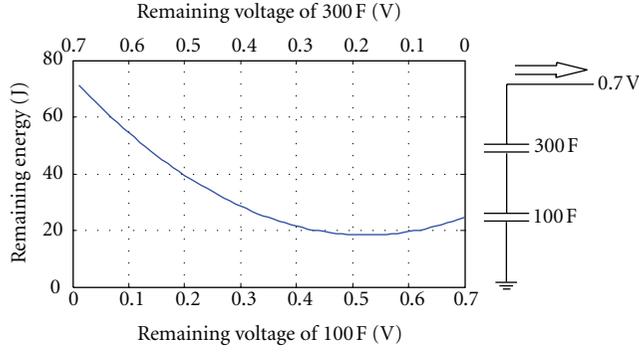


FIGURE 7: Different voltage combinations.

array is minimized at the end of the discharging process so that the lifetime of the system is maximized before the next charging process.

In order to minimize the energy leaked away, the controller uses adaptive discharging control algorithm which is very similar to the adaptive charging control algorithm, except that the capacitor with the highest leakage power needs to be discharged first.

Since the output of the capacitor array serves as the input of the DC/DC converter which has the minimum input voltage (e.g., 0.7 V for MAX1676), to minimize the remaining energy inside the capacitor array, the capacitors are connected in series so that the sum of their voltages is greater than the minimum input voltage of the DC/DC converter. However, for a fixed summation, different combinations of the capacitors' voltage value result in different remaining energy inside the capacitor array. As an example, Figure 6 shows that 100 F and 300 F capacitors are connected in series to provide the same sum of 0.7 V. However, the remaining energy is dramatically different for different combinations of voltage values of 100 F and 300 F capacitors. To minimize the remaining energy inside the series of capacitors, we need to determine the voltage value of each capacitor. The problem can be formulated as:

$$\begin{aligned} & \text{minimize} && V^T C V \\ & \text{subject to} && V \succcurlyeq 0 \\ & && \sum_{i=1}^n V_{c_i} = V_{\min}. \end{aligned} \quad (2)$$

Here, we define $C = \frac{1}{2} \text{diag}(c_1, c_2, \dots, c_n)$, $V^T = [V_{c_1}, V_{c_2}, \dots, V_{c_n}]$. V_{\min} is the minimum input voltage of DC/DC converter. This is a quadratic programming problem, and the analytic solution is as follows.

$$\begin{aligned} V_{c_1} &= V_{\min} \cdot c_{\text{total}} \cdot \frac{1}{c_1}; \\ V_{c_2} &= V_{\min} \cdot c_{\text{total}} \cdot \frac{1}{c_2}; \\ &\vdots \\ V_{c_n} &= V_{\min} \cdot c_{\text{total}} \cdot \frac{1}{c_n}. \end{aligned} \quad (3)$$

Here c_{total} is the total capacitance of all the capacitors in series and $c_{\text{total}} = 1/(1/c_1 + 1/c_2 + \dots + 1/c_n)$.

From the above, we get the voltage values of a series of capacitors with the minimum remaining energy. However, initially all of the capacitors have voltage values larger than the minimum input voltage of the DC/DC converter. In order to minimize the energy leaked away, only the capacitor with the highest leakage power is selected as the input of the DC/DC converter, and all the other capacitors are originally disconnected from each other. To achieve the minimum remaining energy, we need to decide when to connect all of these capacitors in series. If we connect them in series too early (or late), their voltage values may be different which causes the remaining energy to be not minimum. By investigating (3) carefully, one can find that the voltage values of each capacitor share a common factor $V_{\min} \cdot c_{\text{total}}$. Moreover, these voltage values are reverse proportional to their capacitance. Our further experiments revealed that if the capacitors are discharged in series with their initial voltage values reverse proportionally to their capacitance, their voltage value will maintain this relation until the end of the discharge. Furthermore, the capacitor with the largest capacitance has the smallest voltage value. For example, in Figure 7, if the initial voltage values of the 100 F and 300 F capacitors are 2.1 V and 0.7V, respectively, after they are discharged in series, their voltage value will maintain the relation in which the voltage value of 100 F is 3 times more than the voltage value of 300 F.

Guided by the above observation, we can calculate the minimum voltage values of these capacitors before they are connected in series:

$$\begin{aligned} V_{1_{\min}} &= V_{\min} \cdot \frac{c_i}{c_1}; \\ V_{2_{\min}} &= V_{\min} \cdot \frac{c_i}{c_2}; \\ &\vdots \\ V_{i_{\min}} &= V_{\min}; \\ &\vdots \\ V_{n_{\min}} &= V_{\min} \cdot \frac{c_i}{c_n}. \end{aligned} \quad (4)$$

Here we assume that capacitor c_i has the largest capacitance which results in the smallest voltage value. Since all of the parameters in (4) are determined before conducting the discharging control, (4) can be solved off line by using personal computer. In summary, the discharging control algorithm is straightforward and can be described as follows.

Initially, all of the capacitors are adaptively discharged based on which capacitor has the highest leakage power. If the voltage value of the capacitor reaches the minimum voltage value calculated from (4), this capacitor is excluded from the adaptive discharging. When all of the capacitors have the voltage values less than or equal to the voltage in (4), these capacitors are connected in series and conduct discharge until their total voltage value is less than the minimum input voltage of the DC/DC converter.

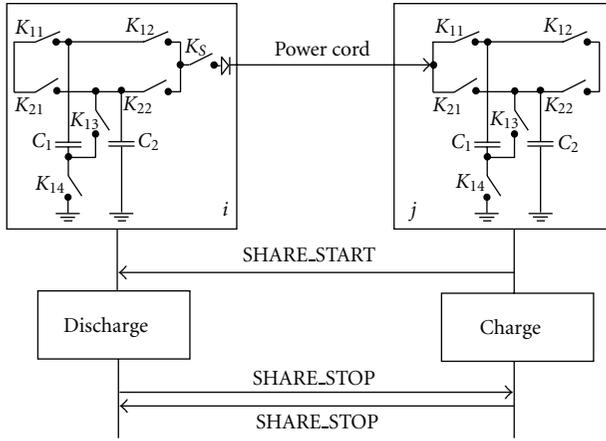


FIGURE 8: Energy sharing protocol.

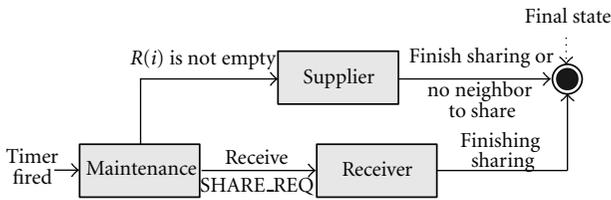


FIGURE 9: State machine diagram of network layer.

We note that there is a trade-off between minimizing the energy leaked away and minimizing the remaining energy. Here minimizing the remaining energy has higher priority than minimizing the energy leaked away. This is because when the capacitors start to be connected in series, their leakage power is very low.

5.4. Energy Sharing Protocol. Without losing generality, we assume that supplier i wants to share some energy with receiver j (shown in Figure 8). To achieve the design goal of being fast and highly efficient, instead of going through the DC/DC converter, the energy is directly shared between the capacitor arrays inside supplier i and receiver j . This is because the DC/DC converter consumes large amounts of energy, especially when its input voltage is low. Moreover, its output current is limited, which significantly increases the time duration of the energy sharing. As shown in Figure 8, the output of i is directly connected to the input of j through power cords. For high-efficiency purposes, we use the power cords instead of using the wireless charging interface.

The energy sharing protocol is a receiver-initiated protocol. After receiving the energy sharing control from the network layer, the receiver j sends out the *SHARE.START* packet to the supplier i and prepares for charging. Upon receiving the *SHARE.START* packet, i turns on the switch K_5 and conducts the discharging control. During the discharging control, the supplier i periodically monitors the amount of energy transferred by sampling the voltage of its capacitors and calculating its own remaining energy until the expected amount of energy is transferred. Then the supplier

turns off the switch K_5 and sends out the *SHARE.STOP* packet to receiver j telling j the end of energy sharing.

On the receiver j side, the charging control is conducted. j also periodically monitors the energy received by sampling the voltage of its capacitors and calculating its remaining energy. When the expected energy is received, j turns off the charging switches (i.e., K_{11} and K_{21} in Figure 8) and sends out the *SHARE.STOP* packet to i telling i that the energy sharing is terminated. Since j can accurately monitor the increased remaining energy inside its capacitor, it can terminate the energy sharing process with high accuracy. In our energy sharing protocol, both the supplier and the receiver can decide when to terminate the energy sharing process. Therefore, our control is robust to the failure of supplier and receiver.

6. Network Layer

As discussed in Section 5, the control layer is designed to minimize the energy leaked away inside a single device and efficiently share energy between two devices. In this section, we introduce the network layer design which tries to efficiently utilize the energy inside the whole network and prolong the network lifetime. Here the network lifetime is defined as the period of time during which all the devices inside the network have energy to execute tasks.

In order to maintain the aliveness of all the devices and minimize the energy leaked away, it is very important for the network layer to balance the remaining energy among these devices. Given the dynamic change of the environmental energy and different working patterns of the devices, the challenge here is to design a general and light-weight network layer that works without knowing the information of environmental energy and the energy consumption of the other devices. To overcome this challenge, the network layer is designed to be a simple finite state machine. As shown in Figure 9, every device inside the network is in one of four states at any time: (i) maintenance, (ii) supplier, (iii) receiver, and (iv) final. Transitions between the states are triggered by events.

Each node's timer fires periodically with time interval T . After the timer is fired, the node enters the maintenance state, in which all of its neighbors' information is periodically maintained. Here, two nodes are considered as neighbors if they are directly connected by using power cords. Whenever the node detects that the energy waste can be reduced by sharing energy with its neighbor, the node enters the supplier state and sends out the *SHARE.REQ* message to the neighbor, which causes the least amount of energy waste. When a node receives the *SHARE.REQ* message from its neighbor, it enters the receiver state in which the node negotiates with the supplier node about the energy sharing. After energy sharing negotiation, the nodes in the supplier state and receiver state enter the final state and wait for the timer fires again. In the rest of this section, we explain the operation in each state in detail.

6.1. Maintenance State. Since the network layer is designed to work under unpredictable environmental energy and different working patterns of the nodes, maintenance is needed to keep track of the remaining energy inside each node and its neighbors. Each node periodically measures its remaining energy with time interval T . Without losing generality, suppose node i is one of the nodes inside the network and its neighboring nodes are j, k, \dots , and n (shown in Figure 10). Assume node i 's remaining energy is E_{R_i} at time kT . Based on E_{R_i} , node i calculates the leakage power (P_{L_i}) of the ultracapacitors by using the leakage model which is discussed in Section 5.1. Similarly, node i 's neighboring nodes can also calculate their own remaining energy and leakage power. Then these nodes share their remaining energy and leakage power information among themselves by embedding this information into their local broadcast packets.

After node i receives the remaining energy and leakage power information from its neighbor (assume node j), node i decides whether it needs to share energy with node j by comparing the energy waste in two scenarios: (i) no energy sharing with node j and (ii) energy sharing. In the first scenario, the energy leaked away inside nodes i and j during the time interval T is the energy waste which can be calculated by using the following equation:

$$W_{i,j} = (P_{L_i} + P_{L_j}) \cdot T. \quad (5)$$

In the second scenario, after sharing the energy with node j , node i 's leakage power is decreased from P_{L_i} to P'_{L_i} . However, node j 's leakage power is increased from P_{L_j} to P'_{L_j} . Therefore, the energy waste is the summation of the energy loss during the sharing procedure and the energy leaked away inside nodes i and j during the time interval T after the energy sharing. The energy waste can be calculated by using the following equation:

$$W'_{i,j} = \Delta E[i \Rightarrow j] \cdot (1 - e_{i,j}) + (P'_{L_i} + P'_{L_j}) \cdot T. \quad (6)$$

Here $\Delta E[i \Rightarrow j]$ is the total amount of energy transferred from node i to node j , and $e_{i,j}$ is the energy sharing efficiency between i and j . Then $\Delta E[i \Rightarrow j] \cdot (1 - e_{i,j})$ is the energy loss during the sharing procedure. For example, if $\Delta E[i \Rightarrow j] = 100J$ and $e_{i,j} = 90\%$, the energy loss during the sharing procedure is $100 * (1 - 0.9) = 10J$.

If energy sharing wastes less amount of energy than no energy sharing (i.e., $W'_{i,j} < W_{i,j}$), node i will share energy with node j . Otherwise, energy sharing does not happen between i and j . The total energy gain during the energy sharing procedure can be calculated as follows:

$$G_{i,j} = W_{i,j} - W'_{i,j}. \quad (7)$$

Since node i may have multiple neighbors, i will send the *SHARE_REQ* message to its neighbor which results in the maximum energy gain. After sending out the *SHARE_REQ*, i enters the supplier state.

6.2. Supplier State. In the supplier state, a node needs to negotiate with its neighbors about the energy sharing. This

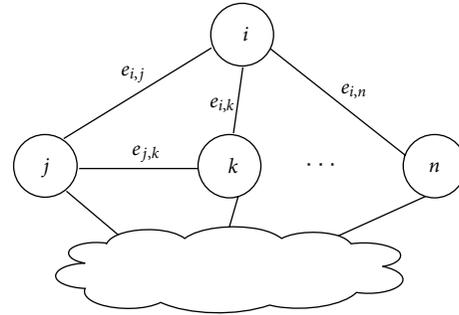


FIGURE 10: Energy sharing networks.

is because the node's initially selected neighbor may receive multiple *SHARE_REQ* messages from other nodes. For example, node i selects node k to share the energy which can achieve the maximum energy gain between node i and node k from node i 's perspective. However, node k may receive multiple *SHARE_REQ* messages from k 's neighbors. From node k 's perspective, sharing energy with node i may not achieve the maximum energy gain. Therefore, node k may deny node i 's energy sharing request. In this case, node i needs to exclude node k and send the *SHARE_REQ* message to its neighbor that can achieve the maximum energy gain. This procedure is repeated until node i either can not find its neighbor to share the energy or receive the *SHARE_ACCEPT* message from one of its neighbors.

If node i can not find a neighbor to share energy, i terminates the energy sharing process and enters the final state. In the case that node i receives the *SHARE_ACCEPT* message from one of its neighbors, i indicates its control layer to conduct the energy sharing control and enters the final state.

6.3. Receiver State. After a node receives the *SHARE_REQ* message from its neighbor, this node enters the receiver state. If this node receives multiple *SHARE_REQ* messages from its neighbors, the node will (i) send *SHARE_ACCEPT* message to the neighbor which can reduce the largest amount of energy waste and (ii) send *SHARE_DENY* messages to the other neighbors. After sending out these messages, the node indicates its control layer to conduct energy sharing control and enters the final state.

In our design, the node in the receiver state can only accept one energy sharing request. This has two advantages.

(i) *Accuracy.* Since the node's remaining energy and leakage power will change after energy sharing, it is not accurate to conduct the second energy sharing.

(ii) *Loop-Free.* The node can either receive or transfer energy for a single time, thus no loop will be created. Therefore, energy can be efficiently utilized.

6.4. The Detailed Protocol. Combining all the design components, the network layer energy sharing protocol can be specified by the pseudocode shown in Algorithm 2. The

```

(1) switch Event do
(2)   case Timer fired
(3)      $R(i) \leftarrow \phi, S(i) \leftarrow \phi, \forall j \in N(i), G_{i,j} \leftarrow 0;$ 
(4)     Measure  $E_{R_i}$  and calculates  $P_{L_i}$ ;
(5)     Share  $E_{R_i}$  and  $P_{L_i}$  information with neighbors;
(6)     for  $j \in N(i)$  do
(7)       Update  $G_{i,j}$  via Formula 5, 6, and 7;
(8)       if  $G_{i,j} > 0$  then  $R(i) \leftarrow R(i) + \{j\};$ 
(9)       Call Receptor Selection method;
(10)    case Receive SHARE_DENY from node  $j$ 
(11)       $R(i) \leftarrow R(i) - \{j\};$ 
(12)      Call Receptor Selection Method;
(13)    case Receive SHARE_ACCEPT from node  $j$ 
(14)      Conduct energy sharing control;
(15)    case Receive SHARE_REQ from node  $k$ 
(16)       $S(i) \leftarrow S(i) + \{k\};$ 
(17)      Wait until receives all the SHARE_REQ from neighbors;
(18)      From  $S(i)$ , select  $k$  which has the maximum  $G_{k,i}$ ;
(19)      Send SHARE_ACCEPT to  $k$ ;
(20)      Conduct energy sharing control;
(21)       $S(i) \leftarrow S(i) - \{k\};$ 
(22)      Send SHARE_DENY to all the nodes in  $S(i)$ ;
(23) Receptor Selection method:
(24) if  $R(i) \neq \phi$  then
(25)   From  $R(i)$ , select node  $j$  which has the maximum  $G_{i,j}$ ;
(26)   Send SHARE_REQ to  $j$ ;

```

ALGORITHM 2: Network layer energy sharing protocol.

design is simple and triggered by the events of either timer fired (Line 2) or receiving energy sharing messages (Lines 10, 13, and 15).

Lines 2 to 9 deal with the event of the timer fired and the node enters the maintenance state. Node i , which runs the protocol, maintains its receptor node set $R(i)$, supplier node set $S(i)$, and neighboring node set $N(i)$. After the timer fired, $R(i)$ and $S(i)$ are initially set to be empty, and the energy sharing gains between i and its neighbors are set to be 0 (Line 3). After sharing the remaining energy (E_{R_i}) and the leakage power (P_{L_i}) information with neighbors, i calculates the energy sharing gain from itself to all its neighbors by using Formula 5, 6, and 7 (Lines 5 to 7). If the neighbors result in positive energy sharing gains, i adds its neighbors into its receptor node set $R(i)$ (Line 8). After adding all the eligible neighbors into $R(i)$, i calls the *Receptor Selection* method (Lines 9). The *Receptor Selection* method tests whether the receptor node set $R(i)$ is empty. If $R(i)$ is not empty, the node with the maximum energy sharing gain will be selected and *SHARE_REQ* is sent to this node (Lines 23 to 26).

Lines 10 to 14 deal with the events when the node is in the supplier state. Lines 10 to 12 handle the event of receiving the *SHARE_DENY* message from a neighbor j . In this case, j is removed from the receptor node set $R(i)$, and a new receptor is selected by calling the *Receptor Selection* method. In the case that i receives the *SHARE_ACCEPT* message from a neighbor j , i indicates its control layer to conduct energy sharing control (Lines 13 and 14).

Lines 15 to 22 handle the event of receiving the *SHARE_REQ* message from a neighbor k when node i is in the receiver state. i adds node k into its supplier node set $S(i)$ (Line 16). After receiving all the *SHARE_REQ* messages from its neighbors, i sends the *SHARE_ACCEPT* message to the node which results in the maximum energy gain and sends the *SHARE_DENY* messages to all the other nodes in its supplier node set (Lines 17 to 22).

6.4.1. Complexity and Feasibility Analysis. The energy sharing protocol is triggered by the events of either timer fired or receiving energy sharing messages. In each case, there is a single loop. Therefore, the complexity of the algorithm is $O(N)$, where N is the number of neighbors a node has. In sensor networks, each node normally has limited number of neighbors. Moreover, the timer fires with long time interval (i.e., 5 minutes in our implementation). Thus, the overhead of the protocol is small. The impact of maintenance interval is evaluated in Section 8.4.

7. Implementation and Evaluation

In this section, we evaluated our system performance under different real-world settings.

7.1. Evaluation Baselines and Metrics. In our experiment, we set the total capacitance to be 400 F. To reduce the boot-up

time and leakage, we use a combination of a 100 F and a 300 F capacitor, which are controlled by our efficient charging and discharging control algorithms. We call this approach the *efficient control (EC)*, which is compared with the following baseline.

- (i) No Efficient Control (NEC). This approach uses the same array of capacitors but provides no efficient control by simply connecting the capacitors in parallel to power the system.

The Key Advantage of Our System Design is That it Efficiently Minimizes the Energy Leaked Away Inside the Energy Storage System. Since it is Hard to Measure How Much Energy is Leaked Away, The Metric Used to Evaluate the System is the *Remaining Energy*, The Energy Inside the Capacitors That Can be Calculated Based on the Measured Voltage Values of the Capacitors. with the Same Amount of Energy Charged Both Into the System and Consumed, The Larger the Value of the Remaining Energy, The Less the Amount of Energy Leaked Away.

7.2. Implementation. Our hardware contains a custom circuit board to harvest (or share) energy to power the embedded device. We used a typical sustainable embedded device (i.e., MICAz node) as a controller which is powered by our energy storage system. We designed and implemented the control layer using TinyOS and NesC. The MICAz node normally works in low-duty cycle mode. It wakes up to conduct the charging, discharging, and energy sharing control. Besides the control operations, it also measures the remaining energy inside the capacitor-array, turns on the radio during the active time interval, and logs the data into the flash for the purpose of off-line analysis of the system performance.

7.3. Outdoor Energy Harvesting Experiment. The outdoor energy harvesting experiment supports such potential applications as data collection [2, 24] and habitat monitoring [25]. In this experiment, we deployed our systems inside a weatherproof enclosure that was set facing south and at a 40° angle to the ground (shown in Figure 11(a)). The total time duration of the experiment was 96 hours. Figure 12 shows the energy harvested by these systems. We placed the solar panels of these systems close to each other to harvest similar amounts of energy from the environment. The MICAz nodes of these systems ran at a 10% duty cycle to consume the same amount of energy all the time. As shown in Figure 11(b), the EC system can efficiently maintain the remaining energy at a higher level than the NEC systems. At the end of the experiment, the remaining energy inside the EC system is 872.8 J, which is 14.4% more than the remaining energy inside the NEC system. Figure 11(c) shows that our capacitor array circuit can smoothly conduct the charging and discharging controls.

7.4. Energy Sharing Experiments. The energy sharing experiment supports such potential applications as wearable computing [18, 19] and robot-sensor networks [1, 20, 21]. In

this section, we evaluate the energy sharing from one device to another device.

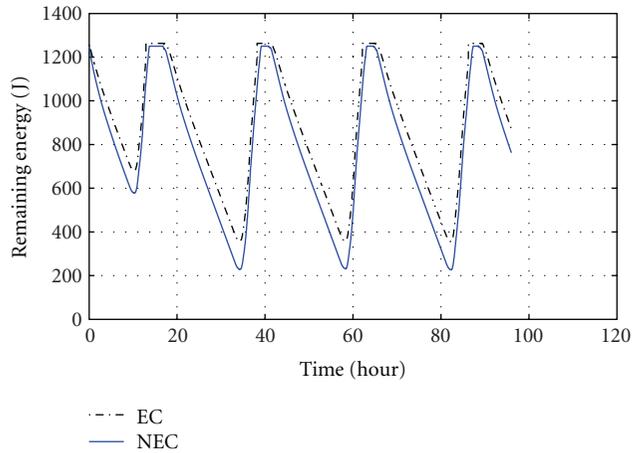
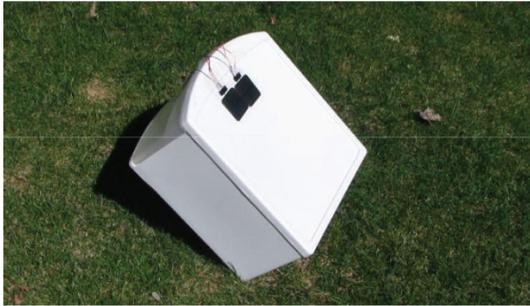
As shown in Figure 13(a), the output of the supplier S_1 is connected to the input of the receptor R_1 . The capacitors in S_1 are connected in parallel and their initial voltages are 2.5 V each. While the capacitors in R_1 are connected in series and their total voltage value is 1.6 V (i.e., 0.4 V for 300 F and 1.2 V for 100 F). The equivalent circuit is shown in Figure 13(b). Figure 13(c) shows the voltage values of the capacitors during the energy sharing process. The energy sharing process starts at time 1 second and ends at time 3.1 second with the total time duration of 2.1 seconds. At the end of the sharing process, the voltage value of the capacitors in S_1 is 2.37 V each, and the voltage values of the 300 F and 100 F capacitors in R_1 are 0.64 V and 1.71 V, respectively. From the voltage values, we can calculate the total energy lost in S_1 is 126.62 J and the total energy gained in R_1 is 113 J which can be used to power the Motorola V557 cell phone to work in standby mode for around 2 hours.

To investigate the impact of the length of interconnect wires, we connected two devices using an 8-inch and a 31-foot power cord. Then we measured these two devices' energy sharing efficiency which is 92.07% for an 8-inch power cord and 83.76% for a 31-foot power cord.

8. Simulation

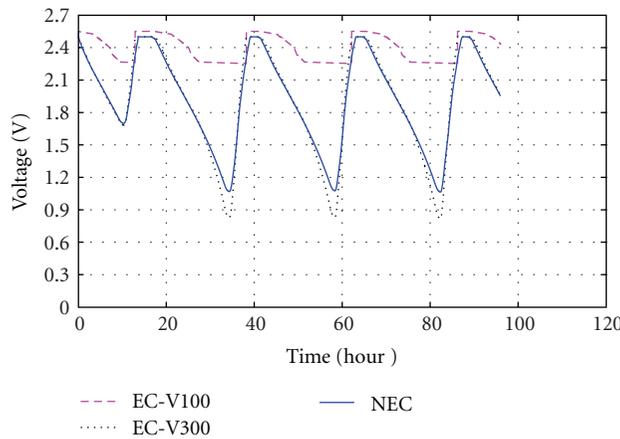
System evaluation has demonstrated the effectiveness of our design in Section 7. However, due to physical constraints, we could not run the experiments for the applications such as wearable networks. Thus to understand the behavior of our design, extensive simulation is necessary. In this section, we choose wearable sensor networks as an example and evaluate its performance under different settings. Since this work is the first one investigating energy sharing applications, the state-of-art approaches (e.g., energy conservation) are complementary, which provide no appropriate baseline for comparison. Therefore, we compare the system that enables energy sharing (ES) with the same system that has no energy sharing (NES). In other words, the sensor devices can not share energy with each other in the no energy sharing wearable networks. The metrics used to evaluate the performance of our system is *Network Lifetime*: the time duration before the first device inside the network runs out of power.

8.1. Simulation Setup. In the simulation, the number of the sensor devices is 25. All of these devices are identical in terms of the energy storage capacity. We use the 3-day real energy traces (shown in Figure 14) collected from a solar panel which is mounted on a graduate student, as a baseline, and generate an energy pattern in which the harvested power varied by a random amount that is -10% to 10% of the baseline for each device. Each device works under a randomly generated working pattern with the mean value of the duty cycle varying from 0.5% to 5%. The same working pattern is applied to the systems with energy sharing and with no energy sharing. Each data point on a



(a) Outdoor harvesting experiment

(b) Remaining energy versus time



(c) Capacitor voltage versus time

FIGURE 11: Outdoor energy harvesting.

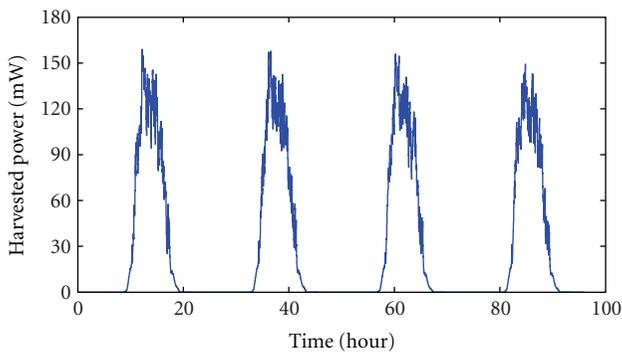


FIGURE 12: Harvested power over 4 days.

graph represents the averaged value of 10 runs, and the 95% confidence intervals of the data are within 3 ~ 8% of the mean shown.

8.2. *Impact of Energy Sharing Efficiency.* In this section, we analyzed the effect of energy sharing efficiency by varying

the value of energy sharing efficiency from 0% to 100%. Figure 15 shows the performance of the systems with no energy sharing (NES) and with energy sharing (ES). When the energy sharing efficiency equals 0, the system with energy sharing has the same lifetime as the system with no energy sharing. As the efficiency increases, the network lifetime of the system with energy sharing also increases. When the efficiency equals 100%, all of the devices in the system with energy sharing is alive all the time (i.e., 72 hours). However, the system with no energy sharing only has a network lifetime equals to 11.3 hours. Since the charge efficiency of rechargeable batteries is as low as 6% [9], from Figure 15, one can tell that the battery-based energy storage system can not take the advantage of the energy sharing feature.

8.3. *Impact of Energy Sharing Error.* The granularity of the A/D converter may introduce the error during the energy sharing. In this experiment, we evaluate the effect of energy sharing error which varies from -25% to 25%. Here we use negative value to indicate that the actual energy gained is less than the energy expected. For example, if a device A wants to receive 100J energy from a device B, -25% means that

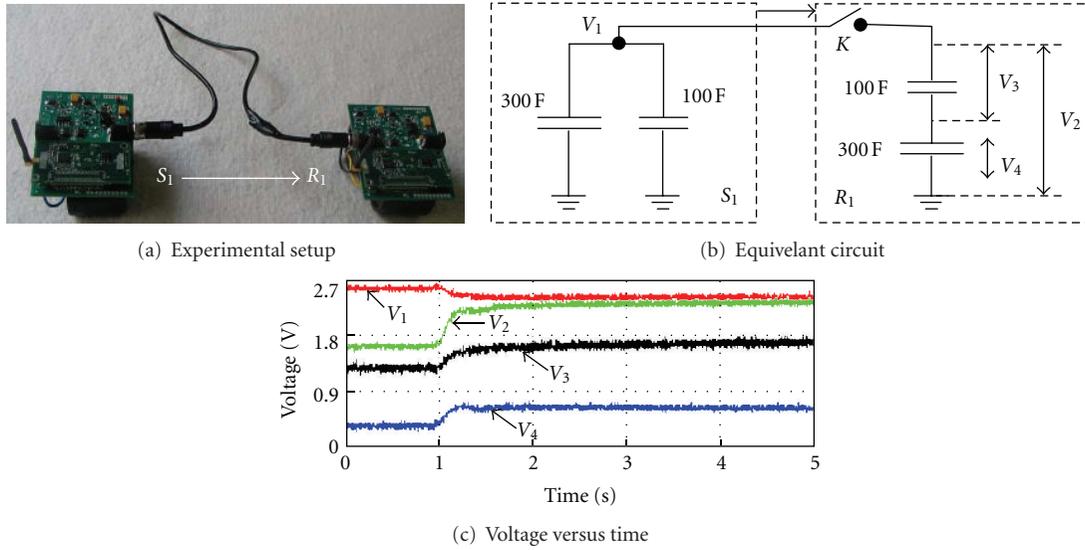


FIGURE 13: Energy sharing between two devices.

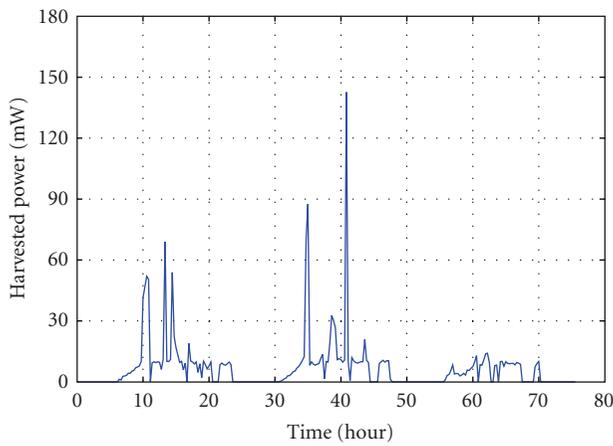


FIGURE 14: Harvested power versus time.

device A actually only receives 75 J. As shown in Figure 16, the energy sharing error could reduce the network lifetime. Moreover, the negative error has more impact on the system performance than the positive error. This is because less energy received by an energy deficient device could reduce the lifetime of the device.

8.4. Impact of Maintenance Time Interval. Maintenance time interval also affects the system performance. A short time interval may introduce unnecessary maintenance overhead. However, a long time interval may affect the system's lifetime. In this section, we evaluate the impact of maintenance time interval by varying the time interval value from 2 minutes to 16 minutes. As shown in Figure 17, the system's lifetime decreases when the maintenance time interval value increases. This is because some energy deficient nodes cannot timely receive energy from their neighboring nodes if the maintenance time interval is too long. Therefore, these nodes

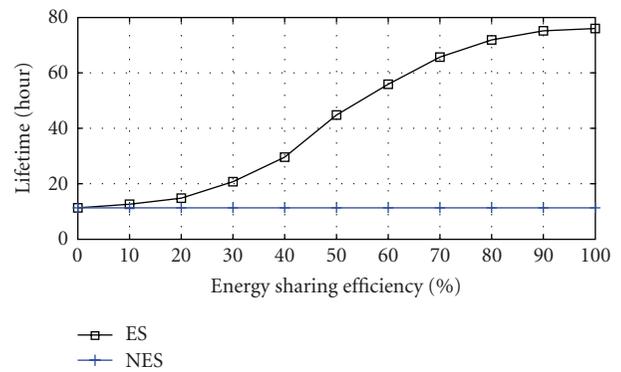


FIGURE 15: Impact of energy sharing efficiency.

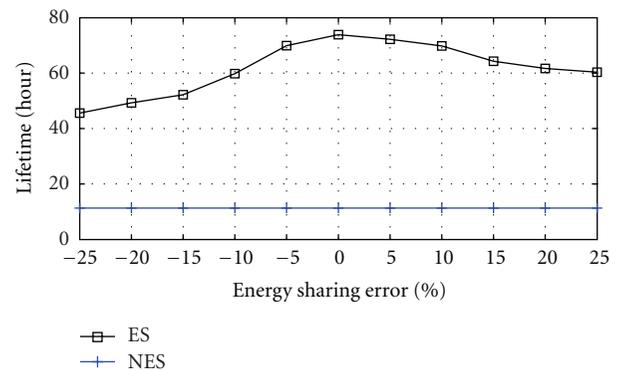


FIGURE 16: Impact of energy sharing error.

will run out of energy which causes the decrease of the system's lifetime. From the simulation result, setting the maintenance time interval value to be 6 minutes can not only reduce the overhead but also achieve a long lifetime.

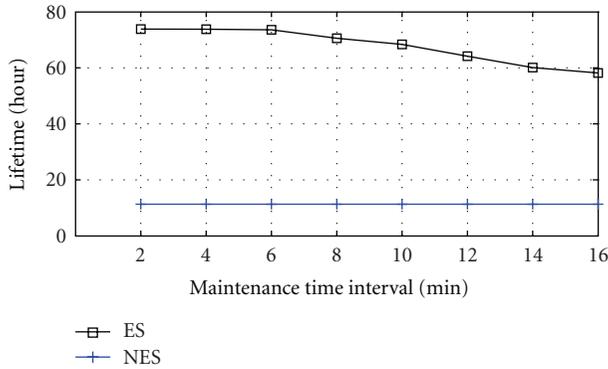


FIGURE 17: Impact of maintenance time interval.

9. Related Work

Energy harvesting is the conversion of ambient energy into usable electrical energy. Several technologies have been developed for extracting energy from the environment, including solar [26], wind [27], kinetic, and vibrational [28] energy. With these energy-harvesting technologies, researchers have designed various types of platforms to collect ambient energy from human activity or environments [23, 26, 27, 29]. Notable ones designed specially for sensor networks are Heliomote [23, 30], Prometheus [26], Trio [29], AmbiMax [27], and PUMA [31]. According to the type of energy storage used, these platforms can be separated into three categories: (i) rechargeable battery-based platforms [30], (ii) designs combining ultracapacitors and rechargeable batteries [26, 29], and (iii) capacitor-based designs.

- (i) In rechargeable battery designs, such as Heliomote [23, 30], the energy harvesting panel is directly connected to its battery. As the primary energy storage device, the rechargeable battery is charged and discharged frequently, leading to low system lifetimes due to the physical limitation on the number of recharge cycles.
- (ii) In designs that combine ultracapacitors and rechargeable batteries such as Prometheus [26], the solar energy is first stored in the primary energy buffer, which is one or multiple ultracapacitors. The rechargeable batteries are then used as the secondary energy buffer. This design inherits both the advantages and limitations of batteries and capacitors. For examples, it is difficult to predict remaining energy because of the inclusion of batteries, and the lifetime of energy storage subsystem is decided by the shelf time of batteries (in the order of a few years). Within this category, several similar systems have been built with a few enhanced features. For examples, AmbiMax [27] harvests energy from multiple ambient power sources (e.g., solar and wind generators), and PUMA [31] uses a power routing switch to route multiple power sources to multiple subsystems. The higher utilization of ambient power

is achieved through a combination of MPPT and power defragmentation.

- (iii) To our best knowledge, previous works have intentionally avoided capacitor-only design, citing the leakage issue [26]. To alleviate leakage, small capacitors are normally used, which makes secondary energy storage (batteries) necessary. However, the development of battery capacities is very slow and still leakage-prone. In addition, charging efficiency for batteries is comparatively low [9]. For example, according to the Natural Resources Defense Council [9], common battery chargers provide an efficiency between 6% and 40%. Different from previous works, this work investigates the frontiers of capacitor-only design and studies not only hardware designs but also related software control techniques to reduce the impact of energy leakage.

Energy conservation is an intensively studied area. Many solutions have been proposed at different layers, including high-efficiency hardware design [32], link layer design [33], topology management [34], node placement [35], network routing [36], sensing coverage [37], data aggregation [38], data placement [39, 40], operating system [41], and application-level energy-aware designs [42]. However, only a few works have focused on energy measurement [43] and energy adaptation [44–46]. Odyssey [44] and ECOsystem [45] demonstrate that application-level adaptations can successfully meet user-specified lifetimes for PDAs and laptops running Linux or other embedded OSes. The differences between these works and ours are that (i) they focus on battery modeling or application adaptations, while our work focuses on battery-free design and control and (ii) they currently do not consider energy harvesting issues.

However, only a few works have focused on energy adaptation [47, 48] and energy storage system [49]. The differences between these works and ours are that (i) they focus on battery modeling or application adaptations, while ours focuses on capacitor driven storage system design and control and (ii) they currently do not take the benefit of energy sharing among embedded devices.

The most closely related work is [50] which focuses on synchronizing the energy supply with demand in a small system powered by a single ultracapacitor. This paper focuses on the energy storage system design and proposes a generic solution for a large system which supports n capacitors and enables energy sharing.

10. Conclusion

The rapid advances in ultracapacitor design and the fast and highly efficient charging properties of ultracapacitor have motivated us to investigate the possibility of using capacitors as the sole energy storage and sharing system for embedded devices. In this paper, we build Energy Mule, an add-on power board that harvests energy from environment and employs an capacitor array as the only energy storage unit to power the embedded device and efficiently share energy among the neighboring embedded devices.

To minimize the energy leaked away inside a single embedded device, we propose adaptive charging and discharging control algorithms. Efficient, fast, and accurate energy sharing is conducted to prolong the service time of the system with multiple devices.

We invested significant amount of effort to evaluate our design in different real-world settings. The results indicate the effectiveness of our design. In future work, we shall focus more on the design of applications so that the system can get higher performance through energy sharing.

Acknowledgment

This paper was supported by NSF under Grant no. 1019343 to CRA for the CI Fellows Project.

References

- [1] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 111–124, 2004.
- [2] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.
- [3] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "FireWxNet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys '06)*, pp. 28–41, June 2006.
- [4] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, pp. 29–39, June 2008.
- [5] B. Anand, A. L. Ananda, M. C. Chan et al., "Game action based power management for multiplayer online game," in *Proceedings of the ACM SIGCOMM Workshop on Networking, Systems, and Applications for Mobile Handhelds (MobiHeld '09)*, 2009.
- [6] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless wakeups revisited: energy management for voip over wi-fi smartphones," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 179–191, June 2007.
- [7] R. K. Balan, N. Ramasubbu, K. Prakobphol, N. Christin, and J. Hong, "mFerio: the design and evaluation of a peer-to-peer mobile payment system," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 291–304, June 2009.
- [8] A. G. Miklas, K. K. Gollu, K. K. W. Chan, S. Saroiu, P. K. Gummadi, and E. Lara, "Exploiting social interactions in mobile systems," in *Proceedings of the International Conference on Ubiquitous Computing (UbiComp '07)*, 2007.
- [9] S. Foster, "The energy efficiency of common household battery charging systems: results and implications," *Natural Resources Defense Council*, http://www.efficientproducts.org/reports/bchargers/NRDC-Ecos.Battery_Charger_Efficiency.pdf.
- [10] W. Rhines, "Mentor Graphics," <http://chipdesignmag.com/lpd/blog/2009/04/15/chip-designers-scramble-for-low-power-solutions/>.
- [11] R. Signorelli, J. Schindall, and J. Kassakian, "Carbon nanotube enhanced double layer capacitor," in *Proceedings of the 15th International Seminar on Double Layer Capacitors and Hybrid Energy Storage Devices*, 2005.
- [12] P. Banerjee, I. Perez, L. Henn-Lecordier, S. B. Lee, and G. W. Rubloff, "Nanotubular metal-insulator-metal capacitor arrays for energy storage," *Nature Nanotechnology*, vol. 4, no. 5, pp. 292–296, 2009.
- [13] Weir et al., "Electrical-energy-storage unit (EESU) utilizing ceramic and integrated-circuit technologies for replacement of electrochemical batteries," US Patent 7033406, 2006.
- [14] "MAXWELL Technologies," <http://www.maxwell.com/products/ultracapacitors/products/k2-series>.
- [15] "NESSCAP Ultracapacitor Products," <http://www.nesscap.com/product/edlc.large1.jsp>.
- [16] "Universal charger for mobile phone handsets," http://www.itu.int/newsroom/press_releases/2009/49.html.
- [17] Sea-Bird Electronics, Inc, "Nickel-Metal Hydride (NiMH) Battery Charger and Battery Pack. User's Manual," http://www.seabird.com/pdf_documents/manuals/NiMH_004.pdf.
- [18] J. Yun, S. Patel, M. Reynolds, and G. Abowd, "A quantitative investigation of inertial power harvesting for human-powered devices," in *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*, pp. 74–83, September 2008.
- [19] U. Maurer, A. Rowe, A. Smailagic, and D. P. Siewiorek, "eWatch: a wearable sensor and notification platform," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '06)*, pp. 142–145, April 2006.
- [20] A. LaMarca, W. Brunette, D. Koizumi et al., "Making sensor networks practical with robots," in *Proceedings of the International Conference on Pervasive Computing (Pervasive '02)*, 2002.
- [21] M. Matusiak, J. Paanajrvi, P. Appelqvist, M. Elomaa, M. Vainio, and A. Halme, "A novel marsupial robot society: towards long-term autonomy," in *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS '08)*, 2008.
- [22] A. LaMarca, D. Koizumi, M. Lease et al., "PlantCare: an investigation in practical ubiquitous systems," in *Proceedings of the 14th International Conference on Ubiquitous Computing (UbiComp '02)*, 2002.
- [23] K. Lin, J. Yu, J. Hsu et al., "Helimote: enabling long-lived sensor networks through solar energy harvesting," in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys '05)*, 2005.
- [24] S. Guo and S. Keshav, "Fair and efficient scheduling in data ferrying networks," in *Proceedings of the 3rd International Conference on Emerging Networking Experiments and Technologies (CoNEXT '07)*, December 2007.
- [25] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware design experiences in ZebraNet," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 227–238, November 2004.
- [26] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 463–468, April 2005.
- [27] C. Park and P. H. Chou, "AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor nodes," in *Proceedings of the 3rd Annual IEEE Communications Society*

- on Sensor and Ad hoc Communications and Networks (Seco '06), pp. 168–177, September 2006.
- [28] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, A. Chandrakasan, and J. Lang, “Vibration-to-electric energy conversion,” in *Proceedings of the International Conference on Low Power Electronics and Design (ISLPED '99)*, pp. 48–53, August 1999.
- [29] P. Dutta, J. Hui, J. Jeong et al., “Trio: enabling sustainable and scalable outdoor wireless sensor network deployments,” in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 407–415, April 2006.
- [30] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, 2007.
- [31] C. Park and P. H. Chou, “Power utility maximization for multiple-supply systems by a load-matching switch,” in *Proceedings of the International Symposium on Lower Power Electronics and Design (ISLPED '04)*, pp. 168–173, August 2004.
- [32] D. Lymberopoulos and A. Savvides, “XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 449–454, April 2005.
- [33] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Impact of radio irregularity on wireless sensor networks,” *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 125–138, 2004.
- [34] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, “ATPC: adaptive transmission power control for wireless sensor networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 223–236, November 2006.
- [35] D. Ganesan, R. Cristescu, and B. Beferull-Lozano, “Power-efficient sensor placement and transmission structure for data gathering under distortion constraints,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 142–150, April 2004.
- [36] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari, “Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks,” in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 108–121, November 2004.
- [37] S. Kang, J. Lee, H. Jang et al., “SeeMon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments,” in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 267–280, June 2008.
- [38] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: a tiny aggregation service for Ad-Hoc sensor networks,” in *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI '02)*, 2002.
- [39] S. Bhattacharya, H. Kim, S. Prabh, and T. Abdelzaher, “Energy-conserving data placement and asynchronous multicast in wireless sensor networks,” in *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys '03)*, 2003.
- [40] C. M. Sadler and M. Martonosi, “DALi: a communication-centric data abstraction layer for energy-constrained devices in mobile sensor networks,” in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 99–112, June 2007.
- [41] C. C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, “A dynamic operating system for sensor nodes,” in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 163–176, June 2005.
- [42] T. He, S. Krishnamurthy, J. A. Stankovic et al., “Energy-efficient surveillance system using wireless sensor networks,” *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 270–283, 2004.
- [43] X. Jiang, P. Dutta, D. Culler, and I. Stoica, “Micro power meter for energy monitoring of wireless sensor networks at scale,” in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 186–195, April 2007.
- [44] J. Flinn and M. Satyanarayanan, “Managing battery lifetime with energy-aware adaptation,” *ACM Transactions on Computer Systems*, vol. 22, no. 2, pp. 137–179, 2004.
- [45] H. Zeng, C. S. Ellis, and A. R. Lebeck, “Experiences in managing energy with ECOSystem,” *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 62–68, 2005.
- [46] A. Lachenmann, P. J. Marrón, D. Minder, and K. Rothermel, “Meeting lifetime goals with energy levels,” in *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys '07)*, 2007.
- [47] X. Liu, P. Shenoy, and M. D. Corner, “Chameleon: application-level power management,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 995–1010, 2008.
- [48] J. Sorber, A. Kostadinov, M. Garber, M. Brennan, M. Corner, and E. Berger, “Eon: a language and runtime system for perpetual systems,” in *Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys '07)*, 2007.
- [49] N. Banerjee, A. Rahmati, M. D. Corner, S. Rollins, and L. Zhong, “Users and batteries: interactions and adaptive energy management in mobile systems,” in *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp '07)*, 2007.
- [50] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z. L. Zhang, “Leakage-aware energy synchronization for wireless sensor networks,” in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 319–332, June 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

