Adaptive Duty Cycling for Energy Harvesting Systems

Jason Hsu, Sadaf Zahedi, Aman Kansal, Mani Srivastava
Electrical Engineering Department
University of California Los Angeles
{jasonh,kansal,szahedi,mbs} @ ee.ucla.edu

Vijay Raghunathan
NEC Labs America
Princeton, NJ
vijay@nec-labs.com

ABSTRACT

Harvesting energy from the environment is feasible in many applications to ameliorate the energy limitations in sensor networks. In this paper, we present an adaptive duty cycling algorithm that allows energy harvesting sensor nodes to autonomously adjust their duty cycle according to the energy availability in the environment. The algorithm has three objectives, namely (a) achieving energy neutral operation, *i.e.*, energy consumption should not be more than the energy provided by the environment, (b) maximizing the system performance based on an application utility model subject to the above energy-neutrality constraint, and (c) adapting to the dynamics of the energy source at run-time. We present a model that enables harvesting sensor nodes to predict future energy opportunities based on historical data. We also derive an upper bound on the maximum achievable performance assuming perfect knowledge about the future behavior of the energy source. Our methods are evaluated using data gathered from a prototype solar energy harvesting platform and we show that our algorithm can utilize up to 58% more environmental energy compared to the case when harvesting-aware power management is not used.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer Communication Networks—*Distributed Systems*

General Terms

Algorithms, Design

Keywords

Energy harvesting; low power design; energy neutral operation

1. INTRODUCTION

Energy supply has always been a crucial issue in designing battery-powered wireless sensor networks because the lifetime and utility of the systems are limited by how long the batteries are able to sustain the operation. The fidelity of the data produced by a sensor network begins to degrade once sensor nodes start to run out of battery power. Therefore, harvesting energy from the environment has been proposed to supplement or completely replace battery supplies to enhance system lifetime and reduce the maintenance cost of replacing batteries periodically.

However, metrics for evaluating energy harvesting systems are different from those used for battery powered systems. Environmental energy is distinct from battery energy in two ways. First it is an inexhaustible supply which, if appropriately used, can allow the system to last forever, unlike the battery which is a limited resource. Second, there is an uncertainty associated with its availability and measurement, compared to the energy stored in the battery which can be known deterministically. Thus, power management methods based on battery status are not always applicable to energy harvesting systems. In addition, most power management schemes designed for battery-powered systems only account for the dynamics of the energy consumers (*e.g.*, CPU, radio) but not the dynamics of the energy supply. Consequently, battery powered systems usually operate at the lowest performance level that meets the minimum data fidelity requirement in order to maximize the system life. Energy harvesting systems, on the other hand, can provide enhanced performance depending on the available energy.

In this paper, we will study how to adapt the performance of the available energy profile. There exist many techniques to accomplish performance scaling at the node level, such as radio transmit power adjustment [1], dynamic voltage scaling [2], and the use of low power modes [3]. However, these techniques require hardware support and may not always be available on resource constrained sensor nodes. Alternatively, a common performance scaling technique is *duty cycling*. Low power devices typically provide at least one low power mode in which the node is shut down and the power consumption is negligible. In addition, the rate of duty cycling is directly related to system performance metrics such as network latency and sampling frequency. We will use duty cycle adjustment as the primitive performance scaling technique in our algorithms.

ISLPED'06, October 4th, 2006, Tegernsee, Germany.

2. RELATED WORK

Energy harvesting has been explored for several different types of systems, such as wearable computers [4], [5], [6], sensor networks [7], *etc*. Several technologies to extract energy from the environment have been demonstrated including solar, motion-based, biochemical, vibration-based [8], [9], [10], [11], and others are being developed [12], [13]. While several energy harvesting sensor node platforms have been prototyped [14], [15], [16], there is a need for systematic power management techniques that provide performance guarantees during system operation. The first work to take environmental energy into account for data routing was [17], followed by [18]. While these works did demonstrate that environment aware decisions improve performance compared to battery aware decisions, their objective was not to achieve energy neutral operation. Our proposed techniques attempt to maximize system performance while maintaining energy-neutral operation.

3. SYSTEM MODEL

The energy usage considerations in a harvesting system vary significantly from those in a battery powered system, as mentioned earlier. We propose the model shown in Figure 1 for designing energy management methods in a harvesting system. The functions of the various blocks shown in the figure are discussed below. The precise methods used in our system to achieve these functions will be discussed in subsequent sections.

Harvested Energy Tracking: This block represents the mechanisms used to measure the energy received from the harvesting device, such as the solar panel. Such information is useful for determining the energy availability profile and adapting system performance based on it. Collecting this information requires that the node hardware be equipped with the facility to measure the power

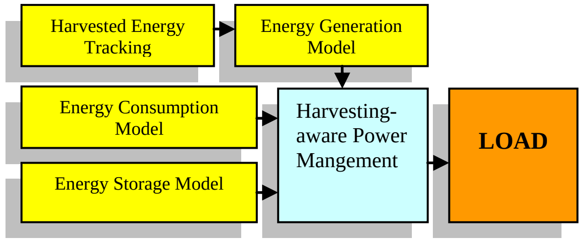

Figure 1. System model for an energy harvesting system.

generated from the environment, and the Heliomote platform [14] we used for evaluating the algorithms has this capability.

Energy Generation Model: For wireless sensor nodes with limited storage and processing capabilities to be able to use the harvested energy data, models that represent the essential components of this information without using extensive storage are required. The purpose of this block is to provide a model for the energy available to the system in a form that may be used for making power management decisions. The data measured by the energy tracking block is used here to predict future energy availability. A good prediction model should have a low prediction error and provide predicted energy values for durations long enough to make meaningful performance scaling decisions. Further, for energy sources that exhibit both long-term and short-term patterns (*e.g.*, diurnal and climate variations vs. weather patterns for solar energy), the model must be able to capture both characteristics. Such a model can also use information from external sources such as local weather forecast service to improve its accuracy.

Energy Consumption Model: It is also important to have detailed information about the energy usage characteristics of the system, at various performance levels. For general applicability of our design, we will assume that only one sleep mode is available. We assume that the power consumption in the sleep and active modes is known. It may be noted that for low power systems with more advanced capabilities such as dynamic voltage scaling (DVS), multiple low power modes, and the capability to shut down system components selectively, the power consumption in each of the states and the resultant effect on application performance should be known to make power management decisions.

Energy Storage Model: This block represents the model for the energy storage technology. Since all the generated energy may not be used instantaneously, the harvesting system will usually have some energy storage technology. Storage technologies (*e.g.*, batteries and ultra-capacitors) are non-ideal, in that there is some energy loss while storing and retrieving energy from them. These characteristics must be known to efficiently manage energy usage and storage. This block also includes the system capability to measure the residual stored energy. Most low power systems use batteries to store energy and provide residual battery status. This is commonly based on measuring the battery voltage which is then mapped to the residual battery energy using the known charge to voltage relationship for the battery technology in use. More sophisticated methods which track the flow of energy into and out of the battery are also available.

Harvesting-aware Power Management: The inputs provided by the previously mentioned blocks are used here to determine the suitable power management strategy for the system. Power management could be carried to meet different objectives in different applications. For instance, in some systems, the harvested energy may marginally supplement the battery supply and the objective may be to maximize the system lifetime. A more interesting case is when the harvested energy is used as the primary source of energy for the system with the objective of achieving indefinitely long system lifetime. In such cases, the power management objective is to achieve energy neutral operation. In other words, the system should only use as much energy as harvested from the environment and attempt to maximize performance within this available energy budget.

4. THEORETICALLY OPTIMAL POWER MANAGEMENT

We develop the following theory to understand the energy neutral mode of operation. Let us define $P_s(t)$ as the energy harvested from the environment at time t, and the energy being consumed by the load at that time is $P_c(t)$. Further, we model the non-ideal storage buffer by its round-trip efficiency η (strictly less than 1) and a constant leakage power P_{leak}. Using this notation, applying the rule of energy conservation leads to the following inequality:

$$B_0 + \eta\int_0^T [P_s(t) - P_c(t)]^+ dt - \int_0^T [P_c(t) - P_s(t)]^+ dt - \int_0^T P_{leak} dt \geq 0 \quad (1)$$

where B_0 is the initial battery level and the function $[X]^+ = X$ if $X > 0$ and zero otherwise.

DEFINITION 1 (ρ,σ_1,σ_2) function: A non-negative, continuous and bounded function $P(t)$ is said to be a (ρ,σ_1,σ_2) function if and only if for any value of finite real number T , the following are satisfied:

$$\rho T - \sigma_2 \leq \int_T P(t)dt \leq \rho T + \sigma_1 \qquad (2)$$

This function can be used to model both energy sources and loads. If the harvested energy profile $P_s(t)$ is a $(\rho_1,\sigma_1,\sigma_2)$ function, then the average rate of available energy over long durations becomes ρ_1, and the burstiness is bounded by σ_1 and σ_2. Similarly, $P_c(t)$ can be modeled as a (ρ_2,σ_3) function, when ρ_2 and σ_3 are used to place an upper bound on power consumption (the inequality on the right side) while there are no minimum power consumption constraints.

The condition for energy neutrality, equation (1), leads to the following theorem, based on the energy production, consumption, and energy buffer models discussed above.

THEOREM 1 (ENERGY NEUTRAL OPERATION): Consider a harvesting system in which the energy production profile is characterized by a $(\rho1, \sigma1, \sigma2)$ function, the load is characterized by a $(\rho2, \sigma3)$ function and the energy buffer is characterized by parameters η for storage efficiency, and P_{leak} for leakage power. The following conditions are sufficient for the system to achieve energy neutrality:

$$\rho_2 \leq \eta\rho_1 - P_{leak} \qquad (3)$$

$$B_0 \geq \eta\sigma_2 + \sigma_3 \qquad (4)$$

$$B \geq B_0 \qquad (5)$$

where B_0 is the initial energy stored in the buffer and provides a lower bound on the capacity of the energy buffer B. The proof is presented in our prior work [19].

To adjust the duty cycle D using our performance scaling algorithm, we assume the following relation between duty cycle and the perceived utility of the system to the user: Suppose the utility of the application to the user is represented by U(D) when the system operates at a duty cycle D. Then,

$$U(D) = 0, \qquad if\ D < D_{\min}$$

$$U(D) = k_1 D + \beta, \quad if\ D_{\min} \leq D \leq D_{\max}$$

$$U(D) = k_2\ , if\ D > D_{\max}$$

This is a fairly general and simple model and the specific values of D_{min} and D_{max} may be determined as per application requirements. As an example, consider a sensor node designed to detect intrusion across a periphery. In this case, a linear increase in duty cycle translates into a linear increase in the detection probability. The fastest and the slowest speeds of the intruders may be known, leading to a minimum and

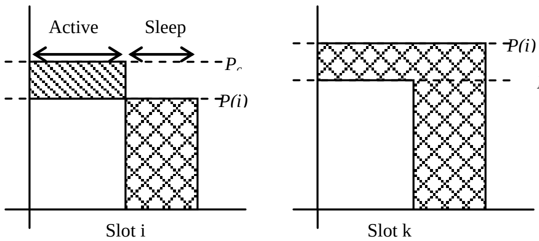

Figure 2. Two possible cases for energy calculations

maximum sensing delay tolerable, which results in the relevant D_{max} and D_{min} for the sensor node. While there may be cases where the relationship between utility and duty cycle may be non-linear, in this paper, we restrict our focus on applications that follow this linear model. In view of the above models for the system components and the required performance, the objective of our power management strategy is adjust the duty cycle *D(i)* dynamically so as to maximize the total utility *U(D)* over a period of time, while ensuring energy neutral operation for the sensor node.

Before discussing the performance scaling methods for harvesting aware duty cycle adaptation, let us first consider the optimal power management strategy that is possible for a given energy generation profile. For the calculation of the optimal strategy, we assume complete knowledge of the energy availability profile at the node, including the availability in the future. The calculation of the optimal is a useful tool for evaluating the performance of our proposed algorithm. This is particularly useful for our algorithm since no prior algorithms are available to serve as a baseline for comparison.

Suppose the time axis is partitioned into discrete slots of duration ΔT, and the duty cycle adaptation calculation is carried out over a window of N_w such time slots. We define the following energy profile variables, with the index *i* ranging over $\{1,\ldots, N_w\}$: $P_s(i)$ is the power output from the harvested source in time slot *i*, averaged over the slot duration, P_c is the power consumption of the load in active mode, and *D(i)* is the duty cycle used in slot *i*, whose value is to be determined. *B(i)* is the residual battery energy at the beginning of slot *i*. Following this convention, the battery energy left after the last slot in the window is represented by $B(N_w+1)$. The values of these variables will depend on the choice of *D(i)*.

The energy used directly from the harvested source and the energy stored and used from the battery must be accounted for differently. Figure 2 shows two possible cases for $P_s(i)$ in a time slot. $P_s(i)$ may either be less than or higher than P_c , as shown on the left and right respectively. When $P_s(i)$ is lower than P_c, some of the energy used by the load comes from the battery, while when $P_s(i)$ is higher than P_c, all the energy used is supplied directly from the harvested source. The crosshatched area shows the energy that is available for storage into the battery while the hashed area shows the energy drawn from the battery. We can write the energy used from the battery in any slot *i* as:

$$B(i)-B(i+1)=\Delta T D(i)\left[P_c-P_s(i)\right]^{+}-\eta\Delta T P_s(i)\left\{1-D(i)\right\}-\eta T D(i)\left[P_s(i)-P_c\right]^{+} \quad (6)$$

In equation (6), the first term on the right hand side measures the energy drawn from the battery when $P_s(i) < P_c$, the next term measures the energy stored into the battery when the node is in sleep mode, and the last term measures the energy stored into the battery in active mode if $P_s(i) > P_c$. For energy neutral operation, we require the battery at the end of the window of N_w slots to be greater than or equal to the starting battery. Clearly, battery level will go down when the harvested energy is not available and the system is operated from stored energy. However, the window N_w is judiciously chosen such that over that duration, we expect the environmental energy availability to complete a periodic cycle. For instance, in the case of solar energy harvesting, N_w could be chosen to be a twenty-four hour duration, corresponding to the diurnal cycle in the harvested energy. This is an approximation since an ideal choice of the window size would be infinite, but a finite size must be used for analytical tractability. Further, the battery level cannot be negative at any time, and this is ensured by having a large enough initial battery level B_0 such that node operation is sustained even in the case of total blackout during a window period. Stating the above constraints quantitatively, we can express the calculation of the optimal duty cycles as an optimization problem below:

$$\max \sum_{i=1}^{N_w} D(i) \quad (7)$$

$$B(i)-B(i+1)=\Delta T D(i)\left[P_c-P_s(i)\right]^{+}-\eta\Delta T P_s(i)\left\{1-D(i)\right\}-\eta T D(i)\left[P_s(i)-P_c\right]^{+} \quad (8)$$

$$B(1)=B_0 \quad (9)$$

$$B(N_w+1)\geq B_0 \quad (10)$$

$$D(i)\geq D_{\min} \quad \forall \mathrm{i}\in\{1,\ldots,\mathrm{N_w}\} \quad (11)$$

$$D(i)\leq D_{\max} \quad \forall \mathrm{i}\in\{1,\ldots,\mathrm{N_w}\} \quad (12)$$

The solution to the optimization problem yields the duty cycles that must be used in every slot and the evolution of residual battery over the course of N_w slots. Note that while the constraints above contain the non-linear function $[x]^+$, the quantities occurring within that function are all known constants. The variable quantities occur only in linear terms and hence the above optimization problem can be solved using standard linear programming techniques, available in popular optimization toolboxes.

5. HARVESTING-AWARE POWER MANAGEMENT

We now present a practical algorithm for power management that may be used for adapting the performance based on harvested energy information. This algorithm attempts to achieve energy neutral operation without using knowledge of the future energy availability and maximizes the achievable performance within that constraint.

The harvesting-aware power management strategy consists of three parts. The first part is an instantiation of the energy generation model which tracks past energy input profiles and uses them to predict future energy availability. The second part computes the optimal duty cycles based on the predicted energy, and this step uses our computationally tractable method to solve the optimization problem. The third part consists of a method to dynamically adapt the duty cycle in response to the observed energy generation profile in real time. This step is required since the observed energy generation may deviate significantly from the predicted energy availability and energy neutral operation must be ensured with the actual energy received rather than the predicted values.

5.1. Energy Prediction Model

We use a prediction model based on Exponentially Weighted Moving-Average (EWMA). The method is designed to exploit the diurnal cycle in solar energy but at the same time adapt to the seasonal variations. A historical summary of the energy generation profile is maintained for this purpose. While the storage data size is limited to a vector length of N_w values in order to minimize the memory overheads of the power management algorithm, the window size is effectively infinite as each value in the history window depends on all the observed data up to that instant. The window size is chosen to be 24 hours and each time slot is taken to be 30 minutes as the variation in generated power by the solar panel using this setting is less than 10% between each adjacent slots. This yields N_w = 48. Smaller slot durations may be used at the expense of a higher N_w.

The historical summary maintained is derived as follows. On a typical day, we expect the energy generation to be similar to the energy generation at the same time on the previous days. The value of energy generated in a particular slot is maintained as a weighted average of the energy received in the same time-slot during all observed days. The weights are exponential, resulting in decaying contribution from older data. More specifically, the historical average maintained for each slot is given by:

$$\overline{x}_k = \alpha \overline{x}_{k-1} + (1-\alpha) x_k$$

where α is the value of the weighting factor, $\overline{x}_k$ is the observed value of energy generated in the slot, and $\overline{x}_{k-1}$ is the previously stored historical average. In this model, the importance of each day relative to the previous one remains constant because the same weighting factor was used for all days.

The average value derived for a slot is treated as an estimate of predicted energy value for the slot corresponding to the subsequent day. This method helps the historical average values adapt to the seasonal variations in energy received on different days. One of the parameters to be chosen in the above prediction method is the parameter α, which is a measure of rate of shift in energy pattern over time. Since this parameter is affected by the characteristics of the energy and sensor node location, the system should have a training period during which this parameter will be determined. To determine a good value of α, we collected energy data over 72 days and compared the average error of the prediction method for various values of α. The error based on the different values of α is shown in Figure 3. This curve suggests an optimum value of α = 0.15 for minimum prediction error and this value will be used in the remainder of this paper.

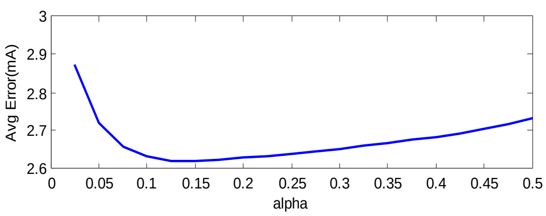

Figure 3. Choice of prediction parameter.

5.2. Low-complexity Solution

The energy values predicted for the next window of N_w slots are used to calculated the desired duty cycles for the next window, assuming the predicted values match the observed values in the future. Since our objective is to develop a practical algorithm for embedded computing systems, we present a simplified method to solve the linear programming problem presented in Section 4. To this end, we define the sets S and D as follows:

$$S = \{ i \mid P_s(i) - P_c \geq 0 \}$$
$$D = \{ i \mid P_c - P(i)_s > 0 \}$$

The two sets differ by the condition that whether the node operation can be sustained entirely from environmental energy. In the case that energy produced from the environment is not sufficient, battery will be discharged to supplement the remaining energy. Next we sum up both sides of (6) over the entire N_w window and rewrite it with the new notation.

$$\sum_{i=1}^{Nw} B_i - B_{i+1} = \sum_{i \in D} \Delta T D(i)[P_c - P_s(i)] - \sum_{i=1}^{Nw} \eta \Delta T P_s(i) + \sum_{i=1}^{Nw} \eta \Delta T P_s(i) D(i) - \sum_{i \in S} \eta \Delta T D(i)[P_s(i) - P_c]$$

The term on the left hand side is actually the battery energy used over the entire window of N_w slots, which can be set to 0 for energy neutral operation. After some algebraic manipulation, this yields:

$$\sum_{i=1}^{Nw} P_s(i) = \sum_{i \in D} D(i) \left(\frac{P_c}{\eta} + P_s(i) \left(1 - \frac{1}{\eta} \right) \right) + \sum_{i \in S} P_c D(i) \qquad (13)$$

The term on the left hand side is the total energy received in N_w slots. The first term on the right hand side can be interpreted as the total energy consumed during the D slots and the second term is the total energy consumed during the S slots. We can now replace three constraints (8), (9), and (10) in the original problem with (13), restating the optimization problem as follows:

$$\max \sum_{i=1}^{N_w} D(i)$$
$$\sum_{i=1}^{Nw} P_s(i) = \sum_{i \in D} D(i) \left(\frac{P_c}{\eta} + P_s(i) \left(1 - \frac{1}{\eta} \right) \right) + \sum_{i \in S} P_c D(i)$$
$$\mathrm{D(i)} \geq \mathrm{D}_{\min} \quad \forall i \in \{1,...,\mathrm{Nw}\}$$
$$\mathrm{D(i)} \leq \mathrm{D}_{\max} \quad \forall i \in \{1,...,\mathrm{Nw}\}$$

This form facilitates a low complexity solution that doesn't require a general linear programming solver. Since our objective is to maximize the total system utility, it is preferable to set the duty cycle to D_{min} for time slots where the utility per unit energy is the least. On the other hand, we would also like the time slots with the highest P_s to operate at D_{max} because of better efficiency of using energy directly from the energy source. Combining these two characteristics, we define the utility co-efficient for each slot *i* as follows:

$$W(i) = \begin{cases} 1/P_c & \text{for} \quad i \in S \\ 1 \Big/ \left(\frac{P_c}{\eta} + P_s(i) \left(1 - \frac{1}{\eta} \right) \right) & \text{for} \quad i \in D \end{cases}$$

where *W(i)* is a representation of how efficient the energy usage in a particular time slot *i* is. A larger *W(i)* indicates more system utility per unit energy in slot *i* and vice versa. The algorithm starts by assuming D(i) $= D_{min}$ for $i = \{1 \ldots N_W\}$ because of the minimum duty cycle requirement, and computes the remaining system energy *R* by:

$$R = \sum_{i=1}^{Nw} P_s(i) - \sum_{i \in D} D(i) \left(\frac{P_c}{\eta} + P_s(i) \left(1 - \frac{1}{\eta} \right) \right) - \sum_{i \in S} P_c D(i) \qquad (14)$$

A negative *R* concludes that the optimization problem is infeasible, meaning the system cannot achieve energy neutrality even at the minimum duty cycle. In this case, the system designer is responsible for increasing the environment energy availability (*e.g.*, by using larger solar panels). If *R* is positive, it means the system has excess energy that is not being used, and this may be allocated to increase the duty cycle beyond D_{min} for some slots. Since our objective is to maximize the total system utility, the most efficient way to allocate the excess energy is to assign duty cycle D_{max} to the slots with the highest *W(i)*. So, the coefficients *W(i)* are arranged in decreasing order and duty cycle D_{max} is assigned to the slots beginning with the largest coefficients until the excess energy available, R (computed by (14) in every iteration), is insufficient to assign D_{max} to another slot. The remaining energy, R_{Last}, is used to increase the duty cycle to some value between D_{min} and D_{max} in the slot with the next lower coefficient. Denoting this slot with index *j*, the duty cycle is given by:

$$\mathrm{D(j)} = \left. \begin{cases} R_{Last} / P_c & if \quad j \in D \\ \dfrac{R_{Last}}{(P_s(j) - P_c)/\eta - P_s(j)} & if \quad j \in S \end{cases} \right\} + D_{\min}$$

The above solution to the optimization problem requires only simple arithmetic calculations and one sorting step which can be easily implemented on an embedded platform, as opposed to implementing a general linear program solver.

5.3. Slot-by-slot continual duty cycle adaptation.

The observed energy values may vary greatly from the predicted ones, such as due to the effect of clouds or other sudden changes. It is thus important to adapt the duty cycles calculated using the predicted values, to the actual energy measurements in real time to ensure energy neutrality. Denote the initial duty cycle assignments for each time slot *i* computed using the predicted energy values as D(i) = $\{1, ..., N_w\}$. First we compute the difference between predicted power level $P_s(i)$ and actual power level observed, $P_s'(i)$ in every slot *i*. Then, the excess energy in slot *i*, denoted by X, can be obtained as follows:

$$X = \begin{cases} P_s(i) - P_s'(i) & if \quad P_s'(i) > P_c \\ P_s(i) - P_s'(i) - D(i)[P_s(i) - P_s'(i)](1 - \frac{1}{\eta}) & if \quad P_s'(i) \leq P_c \end{cases}$$

Input: D: *Initial duty cycle*, X: *Excess energy due to error in the prediction*, P: *Predicted energy profile*, i: *index of current time slot*
Output: D: *Updated duty cycles in one or more subsequent slots*

AdaptDutyCycle()
Iteration: At each time slot do:
if $X > 0$
    $W_{sorted} = W\{1, ..., Nw\}$ sorted in decending order.
    Q := indices of W_{sorted}
    for $k = 1$ to $|Q|$
        if $Q(k) \leq i$ or $D(Q(k)) \geq D_{max}$ //slot is already passed
            continue
        if $R(Q(k), D_{max} - D(Q(k))) < X$
            $D(Q(k)) = D_{max}$
            $X = X - R(j, D_{max} - D(Q(k)))$
        else
        //X insufficient to increase duty cycle to D_{max}
            if $P(Q(k)) > P_l$
                $D(Q(k)) = D(Q(k)) + X/P_l$
            else
                $D(Q(k)) = D(Q(k)) + \dfrac{X}{P_c/\eta + P_s(Q(k))(1-1/\eta))}$
if $X < 0$
    $W_{sorted} = W\{1, ..., Nw\}$ sorted in ascending order.
    Q := indices of W_{sorted}
    for $k = 1$ to $|Q|$
        if $Q(k) \leq I$ or $D(Q(k)) \leq D_{min}$
            continue
        if $R(Q(k), D_{max} - D(Q(k))) > X$
            $D(Q(k)) = D_{min}$
            $X = X - R(j, D_{min} - D(Q(k)))$
        else
            if $P(Q(k)) > P_c$
                $D(Q(k)) = D(Q(k)) + X/P_c$
            else
                $D(Q(k)) = D(Q(k)) + \dfrac{X}{P_c/\eta + P_s(Q(k))(1-1/\eta))}$

ALGORITHM 1 Pseudocode for the duty-cycle adaptation algorithm

The upper term accounts for the energy difference when actual received energy is more than the power drawn by the load. On the other hand, if the energy received is less than P_c, we will need to account for the extra energy used from the battery by the load, which is a function of duty cycle used in time slot i and battery efficiency factor η. When more energy is received than predicted, X is positive and that excess energy is available for use in the subsequent solutes, while if X is negative, that energy must be compensated from subsequent slots.

CASE I: X<0. In this case, we want to reduce the duty cycles used in the future slots in order to make up for this shortfall of energy. Since our objective function is to maximize the total system utility, we have to reduce the duty cycles for time slots with the smallest normalized utility coefficient, W(i). This is accomplished by first sorting the coefficient W(j) ,where j>i. in decreasing order, and then iteratively reducing Dj to Dmin until the total reduction in energy consumption is the same as X.

CASE II: X>0. Here, we want to increase the duty cycles used in the future to utilize this excess energy we received in recent time slot. In contrast to Case I, the duty cycles of future time slots with highest utility coefficient W(i) should be increased first in order to maximize the total system utility.

Suppose the duty cycle is changed by d in slot j. Define a quantity R(j,d) as follows:

$$R(j,d) = \begin{cases} P_l \cdot \mathrm{d} & if \quad P_j > P_l \\ d\left(\dfrac{P_l}{\eta} + P_i\left(1 - \dfrac{1}{\eta}\right)\right) & if \quad P_j <= P_l \end{cases}$$

The precise procedure to adapt the duty cycle to account for the above factors is presented in Algorithm 1. This calculation is performed at the end of every slot to set the duty cycle for the next slot. We claim that our duty cycling algorithm is energy neutral because an surplus of energy at the previous time slot will always translate to additional energy opportunity for future time slots, and vice versa. The claim may be violated in cases of severe energy shortages especially towards the end of window. For example, a large deficit in energy supply can't be restored if there is no future energy input until the end of the window. In such case, this offset will be carried over to the next window so that long term energy neutrality is still maintained.

6. EVALUATION

Our adaptive duty cycling algorithm was evaluated using an actual solar energy profile measured using a sensor node called Heliomote, capable of harvesting solar energy [14]. This platform not only tracks the generated energy but also the energy flow into and out of the battery to provide an accurate estimate of the stored energy.

The energy harvesting platform was deployed in a residential area in Los Angeles from the beginning of June through the middle of August for a total of 72 days. The sensor node used is a Mica2 mote running at a fixed 40% duty cycle with an initially full battery. Battery voltage and net current from the solar panels are sampled at a period of 10 seconds. The energy generation profile for that duration, measured by tracking the output current from the solar cell is shown in Figure 4, both on continuous and diurnal scales. We can observe that although the energy profile varies from day to day, it still exhibits a general pattern over several days.

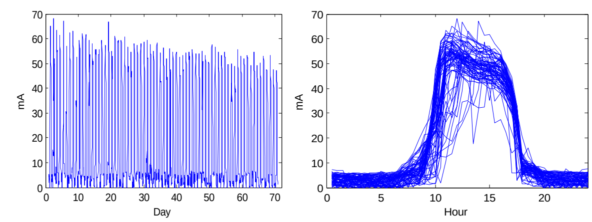

Figure 4 Solar Energy Profile (**Left:** Continuous, **Right:** Diurnal)

6.1. Prediction Model

We first evaluate the performance of the prediction model, which is judged by the amount of absolute error it made between the predicted and actual energy profile. Figure 5 shows the average error of each time slot in mA over the entire 72 days. Generally, the amount of error is larger during the day time because that's when the factor of weather can cause deviations in received energy, while the prediction made for night time is mostly correct.

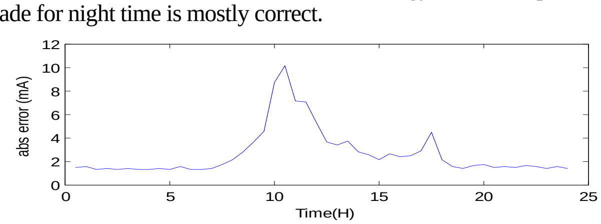

Figure 5. Average Predictor Error in mA

6.2. Adaptive Duty cycling algorithm

Prior methods to optimize performance while achieving energy neutral operation using harvested energy are scarce. Instead, we compare the performance of our algorithm against two extremes: the theoretical optimal calculated assuming complete knowledge about future energy availability and a simple approach which attempts to achieve energy neutrality using a fixed duty cycle without accounting for battery inefficiency.

The optimal duty cycles are calculated for each slot using the future knowledge of actual received energy for that slot. For the simple approach, the duty cycle is kept constant within each day and is computed by taking the ratio of the predicted energy availability and the maximum usage, and this guarantees that the senor node will never deplete its battery running at this duty cycle.

$$D = \eta \cdot \sum_{i \in \{1..Nw\}} P_s(i) \Big/ N_w \cdot P_c$$

We then compare the performance of our algorithm to the two extremes with varying battery efficiency. Figure 6 shows the results, using D_{max} = 0.8 and D_{min} = 0.3. The battery efficiency was varied from 0.5 to 1 on the x-axis and solar energy utilizations achieved by the three algorithms are shown on the y-axis. It shows the fraction of net received energy that is used to perform useful work rather than lost due to storage inefficiency.

As can be seen from the figure, battery efficiency factor has great impact on the performance of the three different approaches. The three approaches all converges to 100% utilization if we have a perfect battery (η=1), that is, energy is not lost by storing it into the batteries. When battery inefficiency is taken into account, both the adaptive and optimal approach have much better solar energy utilization rate than the simple one. Additionally, the result also shows that our adaptive duty cycle algorithm performs extremely close to the optimal.

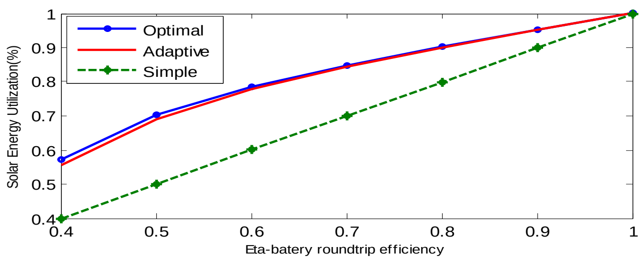

Figure 6. Duty Cycles achieved with respect to η

We also compare the performance of our algorithm with different values of D_{min} and D_{max} for η=0.7, which is typical of NiMH batteries. These results are shown in Table 1 as the percentage of energy saved by the optimal and adaptive approaches, and this is the energy which would normally be wasted in the simple approach. The figures and table indicate that our real time algorithm is able to achieve a performance very close to the optimal feasible. In addition, these results show that environmental energy harvesting with appropriate power management can achieve much better utilization of the environmental energy.

D_{max}	0.8	0.8	0.8	0.5	0.9	1.0
D_{min}	0.05	0.1	0.3	0.2	0.2	0.2
Adaptive	51.0%	48.2%	42.3%	29.4%	54.7%	58.7%
Optimal	52.3%	49.6%	43.7%	36.7%	56.6%	60.8%

TABLE 1. Energy Saved by adaptive and optimal approach.

7. CONCLUSIONS

We discussed various issues in power management for systems powered using environmentally harvested energy. Specifically, we designed a method for optimizing performance subject to the constraint of energy neutral operation. We also derived a theoretically optimal bound on the performance and showed that our proposed algorithm operated very close to the optimal. The proposals were evaluated using real data collected using an energy harvesting sensor node deployed in an outdoor environment.

Our method has significant advantages over currently used methods which are based on a conservative estimate of duty cycle and can only provide sub-optimal performance. However, this work is only the first step towards optimal solutions for energy neutral operation. It is designed for a specific power scaling method based on adapting the duty cycle. Several other power scaling methods, such as DVS, sub-module power switching and the use of multiple low power modes are also available. It is thus of interest to extend our methods to exploit these advanced capabilities.

8. ACKNOWLEDGEMENTS

This research was funded in part through support provided by DARPA under the PAC/C program, the National Science Foundation (NSF) under award #0306408, and the UCLA Center for Embedded Networked Sensing (CENS). Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA, NSF, or CENS.

REFERENCES

[1] R Ramanathan, and R Hain, "Toplogy Control of Multihop Wireless Networks Using Transmit Power Adjustment" in Proc. Infocom. Vol 2. 26-30 pp. 404-413. March 2000

[2] T.A. Pering, T.D. Burd, and R. W. Brodersen, " The simulation and evaluation of dynamic voltage scaling algorithms", in Proc. ACM ISLPED, pp. 76-81, 1998

[3] L. Benini and G. De Micheli, Dynamic Power Management: Design Techniques and CAD Tools. Kluwer Academic Publishers, Norwell, MA, 1997.

[4] John Kymisis, Clyde Kendall, Joseph Paradiso, and Neil Gershenfeld. Parasitic power harvesting in shoes. In *ISWC*, pages 132-139. IEEE Computer Society press, October 1998.

[5] Nathan S. Shenck and Joseph A. Paradiso. Energy scavenging with shoe-mounted piezoelectrics. *IEEE Micro*, 21(3):30ñ42, May-June 2001.

[6] T Starner. Human-powered wearable computing. *IBM Systems Journal*, 35(3-4), 1996.

[7] Mohammed Rahimi, Hardik Shah, Gaurav S. Sukhatme, John Heidemann, and D. Estrin. Studying the feasibility of energy harvesting in a mobile sensor network. In ICRA, 2003.

[8] ChrisMelhuish. The ecobot project. www.ias.uwe.ac.uk/energy autonomy/EcoBot web page.html.

[9] Jan M.Rabaey, M. Josie Ammer, Julio L. da Silva Jr., Danny Patel, and Shad Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *IEEE Computer*, pages 42–48, July 2000.

[10] Joseph A. Paradiso and Mark Feldmeier. A compact, wireless, self-powered pushbutton controller. In *ACM Ubicomp*, pages 299–304, Atlanta, GA, USA, September 2001. Springer-Verlag Berlin Heidelberg.

[11] SE Wright, DS Scott, JB Haddow, andMA Rosen. The upper limit to solar energy conversion. volume 1, pages 384 – 392, July 2000.

[12] Darpa energy harvesting projects. http://www.darpa.mil/dso/trans/energy/projects.html.

[13] Werner Weber. Ambient intelligence: industrial research on a visionary concept. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 247–251. ACM Press, 2003.

[14] V Raghunathan, A Kansal, J Hsu, J Friedman, and MB Srivastava, "Design Considerations for Solar Energy Harvesting Wireless Embedded Systems," (IPSN/SPOTS), April 2005.

[15] Xiaofan Jiang, Joseph Polastre, David Culler, Perpetual Environmentally Powered Sensor Networks, (IPSN/SPOTS), April 25-27, 2005.

[16] Chulsung Park, Pai H. Chou, and Masanobu Shinozuka, "DuraNode: Wireless Networked Sensor for Structural Health Monitoring," to appear in *Proceedings of the 4th IEEE International Conference on Sensors*, Irvine, CA, Oct. 31 - Nov. 1, 2005.

[17] Aman Kansal and Mani B. Srivastava. An environmental energy harvesting framework for sensor networks. In *International symposium on Low power electronicsand design*, pages 481–486. ACM Press, 2003.

[18] Thiemo Voigt, Hartmut Ritter, and Jochen Schiller. Utilizing solar power in wireless sensor networks. In *LCN*, 2003.

[19] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *Technical Report TR-UCLA-NESL-200603-02, Networked and Embedded Systems Laboratory*, UCLA, March 2006.