Performance Aware Tasking for Environmentally Powered Sensor Networks

Aman Kansal, Dunny Potter and Mani B Srivastava
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA, USA
{kansal,dpotter,mbs}@ee.ucla.edu

ABSTRACT

The use of environmental energy is now emerging as a feasible energy source for embedded and wireless computing systems such as sensor networks where manual recharging or replacement of batteries is not practical. However, energy supply from environmental sources is highly variable with time. Further, for a distributed system, the energy available at its various locations will be different. These variations strongly influence the way in which environmental energy is used. We present a harvesting theory for determining performance in such systems. First we present a model for characterizing environmental sources. Second, we state and prove two harvesting theorems that help determine the sustainable performance level from a particular source. This theory leads to practical techniques for scheduling processes in energy harvesting systems. Third, we present our implementation of a real embedded system that runs on solar energy and uses our harvesting techniques. The system adjusts its performance level in response to available resources. Fourth, we propose a localized algorithm for increasing the performance of a distributed system by adapting the process scheduling to the spatio-temporal characteristics of the environmental energy in the distributed system. While our theoretical intuition is based on certain abstractions, all the scheduling methods we present are motivated solely from the experimental behavior and resource constraints of practical sensor networking systems.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer Communication Networks—*Distributed Systems*; C.4 [**Computer Systems Organization**]: Performance of Systems; G.m [**Mathematics of Computing**]: Miscellaneous

General Terms

Performance, Theory, Algorithms, Experimentation

Keywords

energy harvesting, process scheduling, performance guarantees

1. INTRODUCTION

Several prototypes and research efforts have demonstrated the usefulness of sensor networks [1, 2, 3] for a wide variety of applications spanning defense [4], education [5, 6], science [7, 8], to arts and entertainment [9]. However, energy supply still remains one of the open challenges in such systems because unfettered deployment rules out traditional wall socket supplies and batteries with acceptable form factor and cost constraints do not yield the lifetimes desired by most applications.

One method to improve the battery lifetime of such systems is to supplement the battery supply with environmental energy. Several technologies exist to extract energy from the environment such as solar, thermal, optical and kinetic energy, vibrational [10, 11, 12, 13, 14, 15]. However, system level methods to efficiently exploit these resources for optimal performance are lacking. Sensor networks are expected to be deployed for several mission critical tasks and operate unattended for extended durations. This makes performance awareness crucial. Environmental sources are highly variable. A key concern then is ensuring a desired level of performance even as the source varies. In distributed systems, not only does the energy source vary in time, but also the energy available at different locations, and thus at different nodes of the sensor network differs. Energy consumption at different nodes may not be uniform either. In this situation, the performance can be improved by scheduling tasks according to the spatio-temporal characteristics of energy availability. The problem then, is to find scheduling mechanisms which can adapt the performance to the available energy profile. We address the problems mentioned above, both analytically and in experiments on our custom designed harvesting hardware.

1.1 Contributions of this paper

This paper makes several contributions towards achieving a sustainable performance in systems using energy harvesting facilities. First, we develop an analytically tractable characterization for energy sources that can be used for deriving performance bounds. This is a very flexible model which can handle a wide variety of energy sources ranging from natural ones like solar energy to robotic energy delivery.

Next, we propose a harvesting theory that helps to determine performance levels given the energy source classification. This theory aims to answer questions such as the following. What is the minimum latency for a particular application in a given energy environment? What performance level can a system achieve if it must survive eternally (until its hardware gets outdated or damaged) from environmental sources? What additional resources may be needed if a particular quality of service must be achieved? The effect of simplifying assumptions in the theory made for abstracting the complex system characteristics is also discussed.

Third, we present our implementation of a solar energy harvesting circuit for sensor nodes. Our circuit not only powers the node from solar energy and charges the battery but also allows fine-grained tracking of the environmental energy available. It tracks the residual battery status and supplies a stable voltage level to the sensor node even as the battery voltage changes with usage. The circuit has custom interfaces for Berkeley motes [16] and the UCLA MK-II [17] nodes, apart from a generic port for other nodes.

Fourth, we present environment aware tasking methods that help to improve the energy efficiency in distributed systems with communication constraints. Harvesting theoretic bounds are not always achievable due to system constraints on communication between distributed components, model non-idealities and certain noise factors. Idealistic circular disc radio models and perfect time synchronization do not hold in practice. Our methods are designed in consideration of these practical concerns and the specific resource constraints of sensor networks.

2. RELATED WORK

There is significant interest in energy harvesting for many different types of systems for improving their sustainable lifetimes, such as for wearable computers [18, 19, 20], sensor networks [21] and other autonomous systems [22]. Several technologies to extract energy from the environment have been demonstrated including solar, motion-based, biochemical, and vibrational [23, 12, 13, 14, 15], and many more are being developed such as [10, 11]. A method to replenish the energy resources from non-environmental sources was given in [21]. However, there is a need to exploit the available energy in such a way that performance guarantees can be provided on operation, which is not addressed by the above projects. We are providing methods to systematically utilize replenishable energy resources in a performance aware manner. The problem we solve is of immediate benefit to all the above research efforts. The mismatch between environmental energy availability and system requirements for a sustainable network infrastructure in developing regions was noted in [24]. Our solutions combined with Delay Tolerant Networking [25] are also useful in such domains.

Energy efficiency is a major concern in wireless sensor networks [26, 27]. Energy aware methods to take tasking decisions have been considered before for routing [28, 29, 30, 31, 32, 33, 34, 35], data gathering [36], topology management [37] and process sharing [38, 39]. Methods have been proposed to collect the residual battery status of a distributed system [40] and also to estimate the future energy consumption at various nodes [41]. However, all these methods are based on the residual battery status and do not take into account the environmental energy availability at the nodes. Environmental energy is distinct from battery status in two ways. First it is a continued 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. Thus, methods based on the battery status are not always applicable to environmental energy aware decisions.

The first work to take environmental energy into account for routing was [42], followed by [43]. While these works did demonstrate that environment aware decisions improve the performance compared to battery aware decisions, for their specific application scenarios, they did not provide any methods to determine sustainable performance.

Also, some of the above mentioned methods such as [37] work only when the wireless channel can be modelled using a circular disc radio range, which is not typically applicable for wireless networks as shown by [44]. Most of the the proposed methods have been evaluated only in simulations. We evaluate our proposal on a prototype implementation.

Methods have also been suggested to evaluate the maximum data throughput from fixed energy resources [45, 46]. These methods are again ignoring replenishable energy. Further these methods require a-priori knowledge of the exact event distributions and traffic characteristics to be supported by the network which is not a reasonable requirement for practical systems. Even if such information was available, the methods suggested can only be used by a central node which has all the information about all other nodes. This is not applicable to a distributed system as it has a high communication overhead and is not scalable.

We also mention some of the previously used theoretical models which are related to our models. One approach to modelling bursty sources is given by the (r, b) token bucket traffic regulator [47, 48, 49, 50] used to model bursty traffic for QoS in Internet. However, that model is not sufficient to model energy sources for harvesting purposes and we introduce a modified model appropriate for this purpose. Methods to chose appropriate model parameters for the existing models have been explored [51, 52] but those methods are aimed at very different objectives and used to characterize packet data traffic sources. We present related models which are geared for modelling energy sources and incorporate the additional constraints required.

3. HARVESTING THEORY

In this section we describe our proposed analytical models for abstracting real environmental sources and prove two harvesting theorems that help to derive bounds on system performance. We also discuss the realistic factors which are not included in the abstract model and affect its accuracy.

There are several options for controlling the power consumption of a device which directly affect its performance, depending on the specific device being powered. If the processor allows dynamic voltage scaling [53], its power consumption may be reduced by reducing its operating speed. Radio is a major energy consumer in embedded sensors [26], and reducing its transmission range may be helpful in reducing power required. This may of course not be possible depending on network connectivity constraints, and may not affect the receive mode power. A third option is to switch the device between active and sleep modes. We will generically use the term 'sustainable performance' to refer to the average power at which the device operates regardless of the actual technique used to vary the power consumption. Depending on the user's utility considerations for latency, processing speed and other parameters, the appropriate power control technique may be used. We will use specific examples of these techniques in our scheduling algorithms.

3.1 Analytical Model for Harvesting

Environmental energy sources vary a great deal. To derive any analytical results on system performance operating from a given energy source, it is necessary to precisely characterize the energy availability. We observe that this does not require building an exhaustive set of abstractions for all possible energy sources, which vary a great deal from periodically repeating to random ones. Instead, we need to model the energy provided by the harvesting mechanism, which typically consists of a module to convert environmental energy to electrical and storing it in a battery. The model should be simple enough for analytical tractability but capture the significant features of the energy availability such that theoretical results based on it do match reality closely.

Definition: $(\rho, \sigma_1, \sigma_2) - source$: *Suppose* $E(t)$ *is a continuous and bounded function of a continuously varying parameter* t. $E(t)$ *is said to be a* $(\rho, \sigma_1, \sigma_2) - source$ *if and only if for any finite real number T, it satisfies:*

$$\int_T E(t)dt >= \rho T - \sigma 1 \tag{1}$$

$$\int_T E(t)dt <= \rho T + \sigma 2 \tag{2}$$

$E(t)$ can be used to model the power output of any energy source at time t. Since we are modelling physical energy sources, the restrictions placed on the function $E(t)$ are justified. It may be noted that the unit of ρ is power, e.g. Watts and the unit of σ_1 and σ_2 is energy, e.g. Joules. We will later give a method to determine the parameter values for a measured source. The source could be a solar cell [54], thermoelectric generator [55], vibrational energy scavenger [15] or even robot operated energy delivery as in [21].

Our definition uses only three parameters keeping it analytically tractable. This availability model is general enough to model a large variety of energy sources, since it captures the asymptotic rate of availability, which is the maximum energy that can ever be used regardless of changes in the model. However, the real power of this definition will become apparent from the following theorems that lead to useful results of immediate practical concern.

Theorem 1. Sustainable Performance at Eternity (Constant Power Operation):

If

-a device is supplied energy by a $(\rho, \sigma_1, \sigma_2)$-source

-operates at constant power ρ, and

-has an energy storage capacity (such as a rechargeable battery) of $\sigma_1 + \sigma_2$

then,

the device utilizes the energy source fully and can operate forever.

Proof: The proof is divided into two parts.

First, we prove that all energy provided by the source is fully utilized. Proof is by contradiction. Consider the first instant at which an infinitesimal amount of energy ΔE has to be dropped instead of being used or stored in the battery. Suppose this energy is received in a small time duration Δt. This means that at the instant before Δt started, the battery was already full. The following then holds:

$$\int_{\Delta t} E(t)dt = \rho \Delta t + \Delta E \tag{3}$$

There must exist some time at which the battery was filled to capacity σ_1 and a duration T, after that time but preceding the duration Δt, during which the battery received enough energy from the source to get filled to full capacity. Thus,

$$\int_T E(t)dt = \rho T + \sigma_1 \tag{4}$$

Now taking the integral over the entire duration $T + \Delta t$.

$$\int_{T+\Delta t} E(t)dt = \rho(T + \Delta t) + \sigma_1 + \Delta E \tag{5}$$

This clearly violates the constraint that the source is a $(\rho, \sigma_1, \sigma_2)$-source. Thus we have a contradiction. This proves the first part of the theorem.

Second we prove that the device can operate indefinitely at the consumption rate ρ. Suppose initially that the battery was loaded with energy σ_2 or more, such as when starting from a fully loaded battery. If not, the theorem becomes applicable only after the instant when the battery is able to collect the σ_2 charge.

Consider the first infinitesimal time duration Δt when the device is unable to operate. Suppose the energy received from the source in this duration is $e_{\Delta t}$. By our assumption

$$e_{\Delta t} < \rho \Delta t \tag{6}$$

since the device is unable to operate at its constant rate ρ.

Then, the battery must have become empty by this instant. Of the total duration over which the battery lost charge, consider the most recent contiguous interval T over which the last σ_2 of the energy was consumed. Then,

$$\int_T E(t)dt = \rho T - \sigma_2 \tag{7}$$

Now consider the total time duration $T + \Delta t$:

$$\int_{T+\Delta t} E(t)dt = \rho T - \sigma_2 + e_{\Delta t} \tag{8}$$

$$< \rho(T + \Delta t) - \sigma_2 \tag{9}$$

using (6). This violates the constraint that the source is a $(\rho, \sigma_1, \sigma_2)$-source. This is a contradiction and hence there can be no time instant when the device cannot operate, completing the proof.

□

The above theorem assumes that the device operates at a constant power. We now generalize it to the case when the device can switch between various power consumption modes and its consumption varies.

Definition. $(\rho', \sigma) - consumer$: *A device is said to be a* (ρ', σ) *consumer if its power consumption,* $E_c(t)$*, satisfies the constraint*

$$\int_T E_c(t)dt \leq \rho' T + \sigma \tag{10}$$

for any value of T.

For generality, we assume that guarantees needed on performance are independent of the specific energy availability pattern. Special cases, such as when the system is required to operate only when environmental energy is available may lead to different constraints on model parameters than given in Theorem 2.

Theorem 2. Sustainable Performance at Eternity (Variable Consumption Profile): If a (ρ', σ)-consumer device is powered by a $(\rho, \sigma_1, \sigma_2)$-source, has an energy storage capacity of $\sigma + \sigma_1 + \sigma_2$, and $\rho' < \rho$, then the device can operate forever.

Proof: We only outline the proof for brevity. Assume the energy storage to consist of two separate batteries, only for the purpose of argument and not for actual implementation: first battery with capacity $\sigma_1 + \sigma_2$ and a second battery with capacity σ. Consider the energy flow as shown in Fig. 1.

The first battery $\sigma_1 + \sigma_2$ can supply a constant supply at power ρ as proved in Theorem 1. The energy flow at rate ρ is sufficient to support the rate ρ' as per the conditions of the theorem and using the second battery, the variations in the consumption can be supported; detailed analysis is similar to the token bucket regulator in [47].

Note: This theorem is only stated for the case that that the device can always operate without considering the full utilization of the source energy. The reason for excluding the complete utilization of energy is that we have imposed only a weak constraint on the energy consuming device. More specifically, equation (10) only imposes an upper bound on the energy consumption of the source

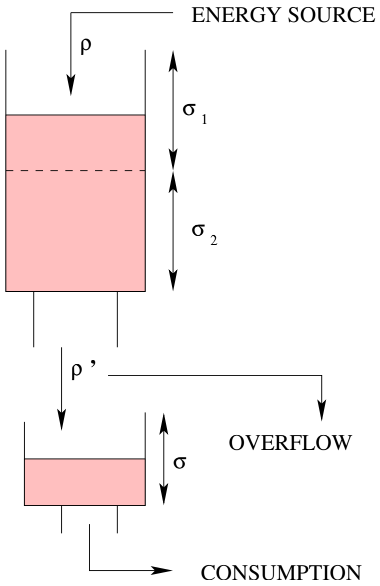

Figure 1: Energy Flow from source to consumer.

and not a lower bound. So, if the consumer just shuts off and stops consuming any energy, it still meets the constraints of the theorem. But as soon as the energy storage capacity of $\sigma + \sigma_1 + \sigma_2$ is exceeded, the system will stop storing the available energy and it will not be fully utilized. This reason for using the weak constraint on the consumer is to simplify the scheduling algorithms which will govern the energy consumption in the system. The consumer node may sometimes want to shut down its radio, regardless of excess energy availability, to reduce interference to other communication. Also, scheduling is much harder in distributed systems. The utility of a distributed system depends on its performance as a whole and not on the performance of individual components. When each component of the distributed system is getting different amounts of energy, distributed scheduling may end up leaving some energy unutilized at nodes which have excessive energy not commensurate with the remaining system. Imposing the weak constraint essentially means that we need not burn energy at a device unless required.

Not having a constraint on the minimum energy consumption does not in any way force us to use suboptimal solutions, it just enlarges the space of acceptable operating points, some of which do not fully utilize the available energy due to practical considerations.

3.2 Implications

The above theory has direct implications for performance guarantees in harvesting supported systems. Let us inspect the questions that we wished to answer using this theory as stated in section 1.1. The first two questions are closely related: What is the minimum latency for a particular application in a given energy environment? What performance level can a system achieve if it must survive eternally? If the parameter ρ for source characterization is determined, we immediately know the average power available to the system. This value of ρ may be smaller or greater than the energy consumption of the device when operating at full power, say P. If $P < \rho$ then the theorem says that the device can always operate provided the system has a rechargeable battery of capacity $\sigma_1 + \sigma_2$. The device may of course shut down or go to a lower power mode when desired and still be sustainable. In this case full performance is ensured as long as our characterization of the energy source is valid. If, on the other hand, $P > \rho$, then the device must lower its energy consumption. Suppose we decide to utilize the sleep mode for power control. Then the device must go into sleep mode or low power mode for certain intervals such that its energy usage can be characterized as a (ρ', σ) consumer. By choosing a low enough duty cycle, we can reduce the average energy consumption from P to ρ'. In this case the duty cycle which is achievable will govern the performance in terms of latency of response from this device.

Another question was: What additional resources may be needed if a particular quality of service must be achieved? When the source has been characterized and it turns out that $\rho < P$, it may happen that the duty cycle which can be supported is lower than required. Here the designer may wish to add an additional source such as an additional solar cell in the case of a single device, or additional components to share the task load in the case of a distributed system. From the values of the model parameters reported by the system, the designer will know what resources are required to reach the desired performance. Depending on this, the designer can select the appropriate performance-resource trade-off.

Another implication of the theory is that it gives explicit battery sizes required for sustainability. Thus, adding larger battery sizes will not improve the sustainable performance, though it may improve the performance for a finite duration.

3.3 Effect of real-world non-idealities

We now mention some detailed features of real systems that cause a departure from the ideal theoretical performance.

The battery is modelled as an ideal energy storage device. This is not exactly true as the amount of energy stored in the battery depends on the charging current profile and the total energy supplied after a recharge depends on the load characteristics [56]. In addition, certain batteries, such as the NiCd rechargeable battery, have memory effects depending on battery chemistry. Accounting for battery peculiarities in load scheduling [57] is possible, but modelling these effects leads to severe complications in the abstract model. The run time scheduling algorithms can compensate these effects for operating close to the theoretically predicted bounds.

Another non-ideality comes in due to the aging effect of components. Battery properties change with usage. The transparency of the protective plastic covering over the solar cells diminishes because polymers absorb solar ultraviolet radiation and undergo photolytic, photo-oxidative, and thermo-oxidative reactions with exposure to sunlight [58]. Such factors are extremely hardware specific and difficult to account for at design time.

Errors will also arise due to inaccuracies in estimating the parameters used for source characterization, and energy usage characterization. While long streams of data may be collected for certain environments before system is deployed, the deployed system will be forced to learn the parameters at run time in most situations. One approach to tackle this is to obtain an estimate of the error in the parameters used and then provide a bound on the error in performance level.

Such approximations exist in any engineering system as most theoretical models do make idealized assumptions, and hence we do not consider these effects a serious drawback of our analysis.

4. SYSTEM IMPLEMENTATION

We now describe our experimental hardware, henceforth referred to as Heliomote. The Heliomote consists of two parts:

1. **Harvestor:** This is our custom hardware containing the solar cells, rechargeable batteries, battery charging circuitry, and the solar energy tracking components.

2. **Sensor Node:** This is an off the shelf mica2 sensor node consisting of a processing and wireless communication platform, with an interface to connect sensors [16].

The harvestor hardware is designed as a small PCB which plugs in directly into the mica2 connector, sandwiched between the sensor board and the processor board of the mica2, so as to leave the sensor board at the top. The battery contacts must now be routed through the harvestor hardware. Figure 2 shows the harvestor board with two solar cells plugged into a mica2 mote.

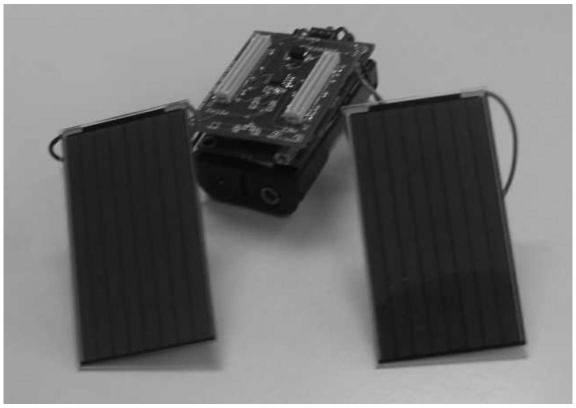

Figure 2: Heliomote: a solar energy harvesting sensor node.

A block diagram of the Heliomote is shown in Figure 3. The purpose and design of each block is briefly described below.

Solar Cell. This transducer generates electric current from incident solar radiation. The output voltage is fairly constant and the output current varies with intensity of radiation [54]. The specific solar cell chosen was essentially guided by the size – this cell is about the same size at the mica2 mote. Two were added as the output of one is not sufficient to power the mote.

Overcharge Protection. The rechargeable batteries have a maximum limit up to which they can be safely charged and charging beyond the limit may damage the battery permanently. The battery voltage increases with the charge level of the battery and this circuit shuts down the charging current current from the solar cell when the maximum voltage is reached. The circuit consists of a comparator with hysteresis that controls an analog switch to route excess solar current to ground when the batteries have reached the upper overcharge voltage threshold of 2.8 V, and to resume charging when the voltage goes down to 2.6 V. The upper threshold is chosen from the battery specifications.

Batteries. NiMH batteries are used. These rechargeable batteries are easily available and have good charge capacity [59]. Furthermore, these batteries do not suffer from memory effect which affects the charging circuit design for NiCd batteries. The charging circuitry depends on the battery chemistry because of the specific current and voltage characteristics of the battery.

Undercharge Protection. Rechargeable batteries can get permanently damaged if drained to the last milliampere. The load on battery must be turned off when the battery voltage falls to the undercharge specification of the battery. The circuit consists of a comparator with hysteresis that controls an analog switch which turns off the battery output when the voltage reaches a low undercharge threshold of 2.3 V, and in turns it back on only when the batteries accumulate some charge reaching 2.55 V.

Table 1: Hardware specifications for the harvestor circuit

Parameter	Value
Noon time solar cell output current	60mA (sunny day)
Solar cell output voltage	3.3V
DC/DC converter input range (for 3V o/p)	2V-3V
AA-sized NiMH battery capacity	1800mAh

Energy Monitor. This smart monitoring chip (Max-DS2438) measures the current flowing into the battery and the battery voltage. It provides a one-wire [60] data interface for the microcontroller of the sensor node. This information is used by the sensor node to learn its energy environment. We have data connectors customized for the mica2 motes [16] and another sensor node, the MK-II [17]. We also provide solder locations for attaching jumper wires if our harvestor is to be used with another sensor node.

DC/DC Converter. Rechargeable batteries such as the nickel metal hydride (NiMH) batteries we use have a lower output voltage (1.2V nominal) than the non-rechargeable batteries (1.5V nominal) for which the sensor node mica2 is designed. To provide the energy at the specified voltage to the sensor node, we incorporate a DC to DC converter which outputs a constant 3V regardless of the battery output voltage. So, as the battery output voltage changes with reducing charge, the sensor node still receives its 3V. We also provide solder contacts for attaching jumper wires if a node is to be powered directly from the battery output, such as may be required if the node itself monitors the battery voltage rather than using our data interface.

Some of the important specifications of the hardware are presented in Table 1.

A popular operating system for the mica2 nodes is TinyOS [61]. We added a firmware driver to TinyOS to add support for our harvestor board. The driver is implemented as a TinyOS component, and can be used to communicate using the one-wire protocol, for getting energy tracking information from the harvestor board. The component provides the following interfaces to TinyOS applications:

1. `async command result_t getData()`: This command will collect the data from the Heliomote, namely battery current, battery voltage and total accumulated current.

2. `async event result_t dataReady`: This event is signaled when the data has been collected from the one-wire interface. The results are placed in the appropriate variables.

Apart from these interfaces we have also implemented a library of one-wire data interface function calls which may be used for communicating with any other hardware that uses the one-wire protocol.

When the Heliomote is in operation, the energy consumed is supplied by the solar cells to the extent available and the remaining requirement is fulfilled by the battery. When excess energy is available from the solar cell, it is used to charge the battery.

Thus, the Heliomote is capable of storing environmental energy and using it for its operation. Also, it can track the energy supplied

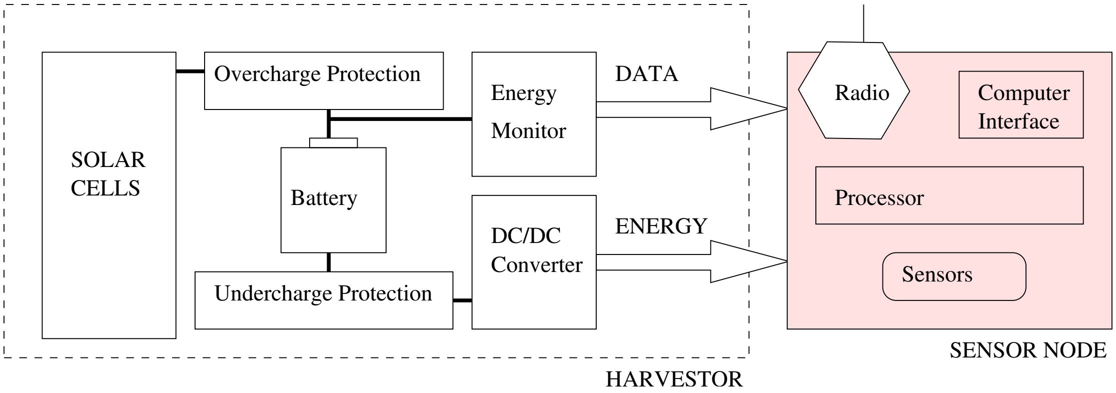

Figure 3: Heliomote Block Diagram.

by the solar cells and provide it over a data interface to the sensor node microcontroller for scheduling decisions.

5. HARVESTING AWARE PERFORMANCE CONTROL: SINGLE SERVER CASE

In this section we show how the theory described in section 3 leads to useful results for real systems and how it can be directly applied for practical performance aware harvesting algorithm design. The goal is to learn the energy source and adapt performance accordingly. We first consider harvesting at a single node and in the next section, discuss it for a distributed system with multiple nodes.

5.1 Direct Application of Theoretical Results

This simple example shows a direct application of the theory. Suppose a heliomote is to be deployed at a location where sunlight is available for part of the day. Will the heliomote last forever if it was supplied with a rechargeable button cell battery worth few 10's of mAh, or does it need a several hundred Ahr battery to sustain forever? The harvesting theory described in section 3 can be used directly to resolve this issue.

Let $V(t)$ and $I(t)$ denote the battery voltage and current flowing through the battery (negative values denote discharge) respectively. Then,

$$P(t) = \int_T V(t).I(t)dt \tag{11}$$

gives the power being used to charge the battery at time t. Negative $P(t)$ means that the battery is actually supplying the power to drive the heliomote while a positive value means that the heliomote is running on solar power and the battery is being charged from solar energy.

Figure 4 shows the power flowing into the battery observed by a test heliomote, when placed in sunlight. Direct sunlight was available from a window. Measuring the battery current instead of the current out of the solar cell ensures that only the actual solar power available to the batteries and not any power lost due to circuit inefficiencies is considered. The current and battery voltage values were recorded at 12 second intervals for 9 days. The days vary among cloudy, hazy and sunny. The node was active at full power for the entire duration. Since the negative portions of the waveform dominate, the batteries suffer a net discharge and were replaced every other day.

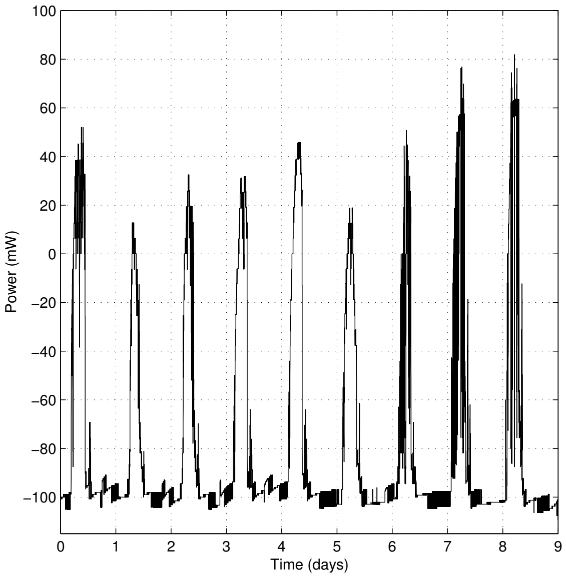

Figure 4: Solar energy based charging power recorded for 9 days

Table 2: Solar cell parameters in experimental environment

Parameter	Value	Units
ρ	23.6	mW
σ_1	1.4639×10^6	J
σ_2	1.8566×10^6	J

For the waveform plotted in Figure 4, we state the source characterization parameter values in Table 2. With these values the solar cell is a $(\rho, \sigma_1, \sigma_2)$-source for the duration for which the waveform is plotted. The characterization is not valid if the solar cell is moved to a different environment or if the environment itself changes.

We now need a (ρ', σ) classification of the consumer node. The sensor node used in the heliomote has a sleep mode power drawn $P_{sleep} = 3$mW and the maximum current drawn $P_{max} = 100$mW. The power drawn actually varies between transmit and receive modes but we chose the maximum value for a worst case achievable performance analysis. Since $P_{max} > \rho$, the node must switch between sleep and fully active modes. To achieve a $\rho' = \rho$, we can set the node to sleep for 78.7% time and active for the remaining time. Suppose that the node must wake up for $T_{min} = 2$ seconds every time it enters active mode, resulting in a sleep duration of 7.416

seconds, leading to a $\sigma = 153J$. The choice of T_{min} depends on application and is discussed in section 5.2.

Battery Size Calculation. If we assume that the collected data is representative of that environment over the duration for which the system is expected to sustain itself, then it immediately follows from Theorem 2 that a rechargeable battery with capacity $= \sigma + \sigma_1 + \sigma_2 = 3.32 \times 10^6$J, or 922.43mAh, is required for sustaining the node indefinitely from the solar energy source. Since a battery of exactly this size may not be manufactured commercially, we need to select a standard battery. The NiMH AA sized battery used in the Heliomote has 1800mAh capacity. Thus, the heliomote will survive forever in the tested environment at the suggested duty cycle. The higher battery capacity than required for sustainable operation is useful for the initial phase of learning the environment. The larger battery capacity does not yield a higher sustainable duty cycle, since the sustainable performance depends on ρ and not on σ_1, σ_2 or σ. A smaller, lighter battery can be used to replace this one without loss in performance. This immediately tells the designer what performance trade-offs are possible by using extra batteries or extra solar cells.

The battery size calculation is very useful because the battery size is always a large portion of sensor nodes and in fact dominates the node size in the case of motes [16].

The environment will in fact not be same all year round as it was on the 9 days when the data was collected. The location shown in Figure 2 is located in the northern hemisphere and the data was collected in late October, with the sun blocked by haze on some of the days. Better energy performance is expected during summer months. However, to account for deep winter months or overly cloudy days, we may wish to reduce our estimate of ρ. The estimates of σ, σ_1 and σ_2 can be increased further to account for the fact that batteries are not 100% efficient. An efficiency of 70% is typical which means that the calculated capacity should be multiplied by $100/70$. With these adjustments, the sustainable performance calculated above becomes achievable for a long duration.

The above analysis shows the immediate applicability of our theory to system hardware design. We now discuss how the theory is useful for performance evaluation and task sharing.

5.2 Algorithms for Performance Adaptation

This section discusses harvesting algorithms which a system powered by the above source could use. For a single device, these consist of two steps: characterizing the source and then adapting performance according to available resources.

Determining the true value of ρ, σ_1 and σ_2 for a source requires the complete knowledge of $E(t)$ for $t \in [0, \infty)$. This knowledge is clearly unavailable. However, note that $E(t)$ is not any arbitrary function but the power supplied by a physical source. Clearly, the parameters can be estimated only if we assume that the $E(t)$ observed over a finite duration is representative of the complete $E(t)$. This is not unreasonable for most sources. Solar energy follows a diurnal and annual cycle. Winds have known repetitive patterns. If the source is highly opportunistic and erratic, we cannot expect to derive any performance constraints for it. Also, a finite time duration will be required for estimating the source parameters. If the system is geared for long term sustainability, the performance in the transient phase of learning the parameters should not be a concern.

We may start the system at an arbitrary performance level and let it gradually attain the sustainable performance. Assume that the initial battery charge is sufficient for the system to last for the transient phase of learning the source parameters. Suppose the acceptable error margin in the estimate of sustainable performance is $\pm\Delta$. The sustainable performance depends directly on ρ and hence we wish to estimate ρ to within Δ. The power parameter ρ can be estimated by averaging the energy obtained from the energy source over time. Let the average at time t, $\rho(t)$ be calculated as:

$$\rho(t) = \frac{\int_0^t E(t)dt}{t} \tag{12}$$

Since the device will sample $E(t)$ at discrete times, the above integral will be evaluated as a discrete summation. Apart from the running estimate of average, also store the most recent local minima and maxima observed in $\rho(t)$. A small number of samples of the $\rho(t)$ waveform need to be stored for this. When the difference between the maxima and the minima reaches Δ we assume that estimation of ρ is complete. Figure 5 shows the waveforms $E(t)$ and $\rho(t)$ for the Heliomote data in our experiment. After 9 days, $\Delta = 2.7$mW.

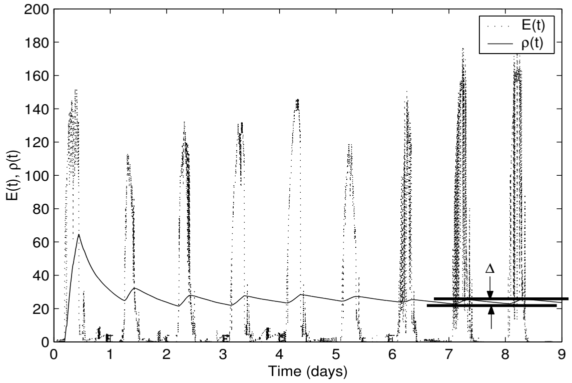

Figure 5: Estimating ρ from $E(t)$.

The waveform $E(t)$ is stored at the device until the time when the estimation of ρ is complete. The period at which the $E(t)$ samples are stored will depend on the memory capacity available at the device. For example for solar energy, if a sample is stored every 30 minutes, 1KB of memory is sufficient to store 20 days of the $E(t)$ waveform. The procedure to estimate σ_1 and σ_2 is as follows. From the stored waveform, subtract ρ. Suppose the i-th contiguous time durations for which the new $E(t) < 0$, is denoted T_{low-i}.

$$\sigma_1 = max_i \left\{ \int_{T_{low-i}} E(t)dt \right\} \tag{13}$$

The maximization is carried out in the implementation by traversing the waveform from the beginning and keeping the largest observed integral over T_{low-i}. The value of σ_2 is similarly calculated by considering all contiguous intervals over which $E(t) > 0$.

Once the source has been characterized, the device can adjust its performance to operate at the available rate ρ. We consider the method of using sleep mode and active mode for controlling the average power consumption. Suppose active mode power is P_{max} and sleep mode power is P_{sleep}. Then the duty cycle x satisfies:

$$\rho = xP_{max} + (1-x)P_{sleep} \tag{14}$$

neglecting the energy consumption of switching between modes. The value of x determined from this equation can be used to decide the sleep duration if the minimum time spent in active mode, T_{min} is known. This depends on the application using the system. For instance, in a sensor network, the node may periodically enter active mode to listen for any possible data waiting to be sent to it [62]. Then the minimum time spent in active mode is the time required for the node to recognize a beacon packet from a transmitter which is waiting to send data to it. Thus, the node would have a maximum latency of data reception, T_{Rx}, given by:

$$T_{Rx} = T_{min}/x \tag{15}$$

Using the above method in the Heliomote, performance was adjusted to suit the energy source parameters learnt from the first nine days of solar data (shown in Fig. 4) and the Heliomote entered sleep mode for the duty cycle calculated using (14) and a $T_{min} = 2$s. The residual battery voltage is plotted in Fig. 6 for 60 hours following the performance adjustment. The figure shows that the battery voltage is not deteriorating and the Heliomote can be sustained indefinitely. The battery voltage does increase during the day as charge is stored and decreases at night as charge is withdrawn.

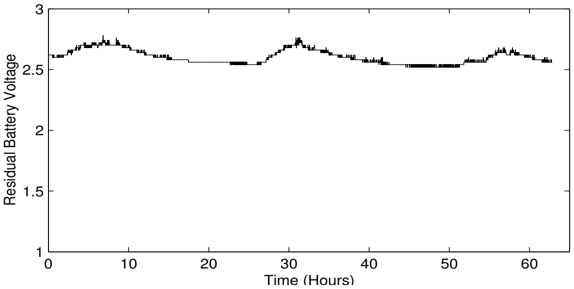

Figure 6: Residual battery plot with time, after the Heliomote has adjusted performance to match the environmental energy source characteristics.

The above experiment establishes the feasibility of operating solely on the energy scavenged from the environment. The battery is used to store the energy when excess energy is available and supply it when the energy source is not providing any energy. The average residual battery is thus kept constant.

6. HARVESTING AWARE PERFORMANCE CONTROL: DISTRIBUTED MULTI -SERVER CASE

In a multi-server system, i.e., a distributed system with several nodes having harvesting opportunity, the scheduling problem would be to distribute the workload among the nodes in such a way that the overall performance of the system is maximized. Finding the optimal solution requires complete knowledge of what energy resources are available at every node what all tasks would be performed by the system. Many of these tasks, such as routing a data packet from one node to another involve energy consumption at several intermediate nodes, making the scheduling decisions coupled among nodes. Also, in a distributed system, scalability concerns dictate that all the information at every node cannot be communicated to a central node for scheduling decisions. Moreover, in many distributed systems such as sensor networks, communication itself is the major energy consumer and distributed decisions are the assumed norm. We thus describe a distributed solution, which yields an achievable sustainable-performance level, though not necessarily optimal.

Our distributed scheduling method is designed for a monitoring application in a sensor network. The method informs the user what the sustainable performance is and operates the network at this performance level.

The monitoring application considered is one where each sensor node senses data and reports it to a base node when an event of interest is sensed. The sensing transducers alone consume minimal power and are kept on when the radio and processor on the sensor node is in sleep mode. Assume that when the sensor reading crosses a threshold the mote radio and processor components can be woken up. This is viewed as an event detection and the mote wishes to send the measured event data to the base. The topology management and routing schemes must ensure that whenever a node wishes to send data, the data can be delivered to the base node with minimal delay. We will use this delay to measure the performance level of the network.

While the use of sleep mode has been considered for topology management in [62], no methods were given to determine the route specific latencies and the energy status of the nodes, either environmental or residual battery, was not considered.

6.1 Eternally Sustainable Operation

The scheme for achieving eternity for the above monitoring application is described below. The network first carries out initialization and route discovery, and later enters the sustainable operation mode. The transition from the arbitrary, potentially non-sustainable, performance level at the beginning to the sustainable performance level does not require any centralized coordination. This ensures scalability of the network with increasing number of nodes. The transition happens gradually in a distributed manner and when it is complete, the base node automatically comes to know about the sustainable performance level. Our power control method is based on duty cycling the nodes between sleep and active modes. Extensions to cases where power control is achieved by dynamic voltage scaling will be similar.

6.1.1 Communication Protocol

We now show a method to successfully transmit packets in the presence of nodes which are not always awake. The proposed communication method does not assume any time synchronization unlike the sleep modes in Bluetooth [63], 802.11 [64] or SMAC [65] and does not require the use of two radios as in [62]. The communication protocol is designed to work smoothly with the performance adjustment scheme described in section 6.1.2.

When a node has data to transmit to its next hop neighbor along the data path, the node transmits a BEACON packet and listens for response for a period T_{ack}. It repeats this process until an ACK is received. T_{ack} is the time required by an active node to send an ACK after receiving a BEACON packet. Suppose the time required to transmit a BEACON packet is T_{beacon}. Every node in the network wakes up for a duration $T_{min} = 2T_{beacon} + T_{ack}$ to listen for any BEACON messages from nodes attempting to send data to it. Note that listening for T_{beacon} is not sufficient as we do not have accurate time synchronization among nodes. After every awake period, it sleeps for a duration

$$T_{sleep} = \frac{1-x}{x} T_{min} \tag{16}$$

where x is the sustainable duty cycle at this node. Whenever a node receives a BEACON packet, it transmits an ACK and stays awake until it has received the relevant data and forwarded it to the next hop. With this arrangement, the maximum delay in receiving an ACK when repeatedly sending a BEACON is $T_{sleep} + Tmin$. We consider a monitoring application where the events themselves are rather infrequent and the energy spent in transmitting the event data is negligible compared to the energy spent in periodically entering active mode to listen for potential data, in order to maintain a communication topology in the network.

The route discovery from all nodes to the base is based on the formation of a data gathering tree, adopted from [66]. The base transmits an INIT packet. All nodes which receive this packet, treat the sender of the INIT packet as their parent and send an ACK with random back-off delay. Denote the node from which an ACK is received to be the child. They now know the route to the base, which is one hop away. These nodes assign themselves depth = 1. They retransmit the INIT message with their own ID. All nodes which have not already assigned themselves a depth acknowledge this packet. These nodes now know their next hop on the route to the base and assign themselves depth = 2. The process continues. When a node transmits an INIT message but does not receive any responses for a timeout duration, it assumes that there are no nodes deeper than itself along the path that passes through it. Such nodes are denoted leaf nodes. Thus all nodes now know their routes to the base. These routes may be refreshed periodically. Note that this route discovery did not have to wait for the nodes to learn their energy source parameters.

6.1.2 Distributed Performance Adjustment

The network starts at an arbitrary performance level. All nodes start estimating their ρ, σ_1 and σ_2 parameters. This takes place until time T_{learn}, which may be different at each node. In this phase the network is possibly depleting its batteries. Once the estimation of the source parameters is completed, using the same methods as given in section 5.2, the nodes will begin transition to the sustainable performance level. The nodes already know their active mode power consumption and sleep mode power consumption. Based on ρ, P_{max} and P_{sleep}, the node determines its duty cycle x using (14). Using the knowledge about its initial battery capacity, the learnt parameters and using $\rho' = \rho$, each node uses Theorem 2 to decide if it can sustain forever. If the battery size available turns out to not satisfy the conditions of Theorem 2, then the parameter ρ must be adjusted. We do not specify the details of doing it as we assume that sufficiently large batteries will be available. The node sets its response latency, L, equal to $T_{min} + T_{sleep}$. If however, the node is not receiving any energy from the environment, it may have to select a large value of L for itself. This value will have to be chosen depending on the application level utility trade-off between response latency and sustainable lifetime, and is not discussed here.

When any node has completed its estimate of L it adjusts its duty cycle to the corresponding value. When a leaf node has completed its initial estimation of L, it sends a LATENCY message to its parent containing its response latency. Thus the nodes gradually transition into their individual sustainable performance levels, with no centralized coordination. A node which is not a leaf, waits for the LATENCY packet from all its children. Let L_i denote the latency value heard from i-th child and let it have N_c children. When it has collected all LATENCY packets from its children and has estimated its own L, it sets

$$L = max_{i=1...N_c}\{L_i\} + L \tag{17}$$

It sends this cumulative value of L to its parent. This L thus represents the total worst case data transfer latency along the path from this node to the leaf with worst delay. All nodes which have children (except the base node, which is assumed to be always on) follow the same pattern. Eventually, when all nodes have estimated their L and sent this information to their parents, the base node would have received the L values from all depth 1 nodes. The base then chooses L_{max} equal to the largest L among these. This L_{max} is declared to be the maximum latency of data transfer in the network along the worst path. The worst path may not be the longest path as the maximum latency may come from a path which has nodes with very low duty cycles. Thus, the base node can inform the user or application of the sustainable performance. The latency represents the maximum time that it may take some event generated at an arbitrary node to be reported in the absence of channel errors. As a wireless channel has a finite error probability, the performance level provided is not a hard guarantee. The latency will be more when retransmissions are required due to channel errors.

The in-network processing of the transmitted L values, shown in (17), at each parent reduces the data sent to the parent nodes and the amount of data sent to the base node is thus proportional to the number of depth 1 nodes and not to the total number of nodes in the network. This is very significant for ensuring scalability.

Based on the above discussion, the precise algorithm used by the networked nodes is summarized below:

Algorithm followed at each sensor node:

```
1. Set $T_{sleep} = 0$, childset = $\phi$, childlatencylist =$\phi$, parent = UNKNOWN, $IS\_LEAF$=FALSE.

2. Spawn thread to estimate L. Generates event $L\_estimated$ when done.

3. If received INIT message and parent == UNKNOWN
   (a) Set parent = sender-ID from INIT message
   (b) Send ACK with random backoff
   (c) Rebroadcast INIT message with own ID
   (d) Start ACK wait timer

4. If receive ACK
   (a) childset = childset ∪ sender-ID from ACK
   (b) Delete ACK wait timer

5. If ACK wait timer expires, set $IS\_LEAF$ = TRUE.

6. If $L\_estimated$ event received:
   (a) If $IS\_LEAF$ == TRUE,
       If parent ≠ UNKNOWN,
       send LATENCY message to parent.
       adjust $T_{sleep}$ and duty cycle for this $L$
       else
       wait for INIT message.
   (b) If $IS\_LEAF$ == FALSE, check if childlatencylist is complete. If yes,
       i. Adjust duty cycle and $T_{sleep}$ for this value
       ii. calculate L using equation 17 and send LATENCY message to parent.
       If not, wait for childlatencylist to complete.

7. If receive LATENCY message:
   store latency value in childlatencylist at appropriate child-ID.
```

Apart from the above algorithm, the node will execute the procedure of repeating a BEACON and waiting for ACK when it wishes to send data.

6.2 Latency Performance

In our distributed scheduling method, we are finding the latency of the first path that is found by the node to the base. Clearly, latency could be reduced if we find the lowest latency path from each node to the base. The lowest latency paths can be found using a link cost metric equal to the response latency of the receiver node in the link. This has two disadvantages. First, link costs are established only after nodes have estimated the energy source parameters. This means that the routes established initially will have to be changed. The change can take place only after all the nodes have learnt their energy source parameters. The asynchronous and gradual transition to the sustainable performance will not be possible. Second, a large number of packets need to be exchanged to find the lowest cost route compared to the level discovery based method we use.

We now compare the latency achieved in our simplified distributed method and an optimal lowest latency route discovery method. The multi-server harvesting schemes are not yet implemented on a real system as we are still in the process of building a network with multiple Heliomotes installed in an environment with solar energy. To evaluate expected performance, we perform a simulation attempting to mirror the actual Heliomote parameters as closely as possible. Our experiments with a single Heliomote in a sample environment yielded an average energy availability, $\rho = 23.6$mW as mentioned earlier. However, in a network, not all nodes will have the same energy availability. We consider the energy availability to be uniformly random with mean $\rho = 23.6$mW, varying between 13.6mW and 33.6mW. We consider a network with 100 nodes. The communication range is 20m. $T_{min} = 250$ms. Node density listed in the following figures is number of nodes per radio coverage circle. In the simulation we use the Bellman-Ford routing algorithm to find the minimum latency route. The routes for our simplified distributed method are assumed to be the shortest hop count routes, since it is likely that the INIT packet reaches a node via the shortest hop route first.

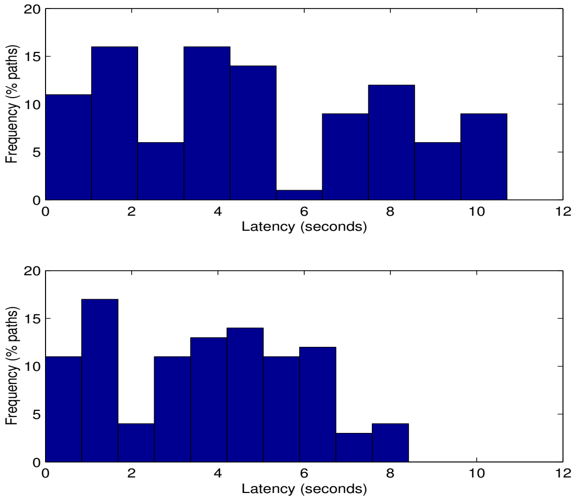

Figure 7: Histogram of path delays in a simulated test case for the distributed deployment of Heliomotes.

Figure 7 shows the histograms of path latencies observed for the minimum latency routing and our simplified distributed routing, for one random network topology. As expected, the histogram shifts towards lower delay values when the optimal lowest latency routing is used. Figure 8 shows the maximum latency for the two types of routing for different network deployment densities. All values are obtained by averaging over 10 random topologies. The optimal method does not yield drastic improvements over the simplified distributed method. If the small latency gain is significant from the point of view of the application, maintaining these optimal routes should be considered. However, the routing control overhead then needs to be accounted for in the energy calculations.

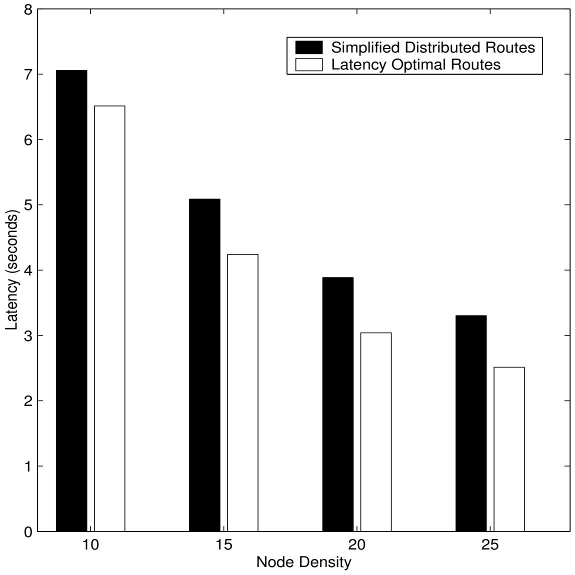

Figure 8: Maximum path latency observed for simplified distributed routing and for optimal minimum latency routing.

The above study shows that performance aware scheduling in distributed systems which survive from environmental energy is feasible to operate at sustainable power levels. We presented our scheduling scheme for a monitoring application, other schemes can be designed for different classes of applications.

7. CONCLUSIONS

We presented a theoretical model which can be used to characterize an energy source with a small number of parameters and these parameters can then be used to determine the performance. The theorems we proved show in which cases this performance is sustainable. We also presented a hardware implementation of a system which can harvest energy and survive on it. Our tests verify the usefulness of our theoretical model and yield significant insights into how environmental energy can be efficiently managed. The battery size estimates obtained from our theory are another additional benefit for system designers.

We also discussed some scheduling algorithms for distributed systems, along with their simulated performance. We are now building a distributed system with our harvesting nodes to design and verify harvesting algorithms for multi-server systems. Future work also includes determining methods for estimating the source characterization parameters for various energy scavenging technologies. Our theory could also be extended towards efficient utilization of sporadic and opportunistic energy availability in a performance-aware manner.

8. ACKNOWLEDGEMENTS

This paper is based on research supported in part by the NSF Center on Embedded Networked Sensing and by the Office of Naval Research.

9. REFERENCES

[1] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[2] A Savvides and Mani B Srivastava. A distributed computation platform for wireless embedded sensing. In *IEEE nternational Conference on Computer Design: VLSI in Computers and Processors*, pages 220 –225, September 2002.

[3] J Rabaey, J Ammer, J.L da Silva Jr., and D Patel. Picoradio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes. pages 9–12, Orlando, Fl, USA, April 2000.

[4] Andrew Blanchard, Andrew Cilia, and James Cai. High bandwidth mega-mesh sensor networks. In *High-Speed Networking Workshop IEEE HSN (in conjunction with INFOCOM 2003)*, San Francisco, California, USA, March 2003.

[5] M Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving enviroments. In *Proceedings of the Seventh ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 132 – 138, July 2001.

[6] P Steurer and MB Srivastava. System design of smart table. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 473 –480, March 2003.

[7] Alberto Cerpa, Jeremy Elson, Deborah Estrin, Lewis Girod, Michael Hamilton, and Jerry Zhao. Habitat monitoring: Application driver for wireless communications technology. In *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.

[8] A Mainwaring, J Polastre, R Szewczyk, D Culler, and J Anderson. Wireless sensor networks for habitat monitoring. In *First ACM Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, USA, September 2002.

[9] Jeff Burke, Eitan Mendelowitz, Joseph Kim, and Rex Lorenzo. Networking with knobs and knats: Towards ubiquitous computing for artists. In *Ubiquitous Computing, Concepts and Models Workshop*, Gothenburg, Sweden, 2002.

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

[11] 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.

[12] Chris Melhuish. The ecobot project. www.ias.uwe.ac.uk/energy_autonomy/EcoBot_web_page.html.

[13] 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.

[14] 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.

[15] Scott Meninger, Jose Oscar Mur-Miranda, Rajeevan Amirtharajah, Anantha Chandrakasan, and Jeffrey Lang. Vibration-to-electric energy conversion. In *Proceedings of the 1999 international symposium on Low power electronics and design*, pages 48–53. ACM Press, 1999.

[16] Mica, mica2 motes and sensors. http://www.xbow.com/.

[17] Andreas Savvides. Medusa mk-2 node. http://nesl.ee.ucla.edu/projects/ahlos/hardware.htm.

[18] John Kymisis, Clyde Kendall, Joseph Paradiso, and Neil Gershenfeld. Parasitic power harvesting in shoes. In *Second IEEE International Conference on Wearable Computing, (ISWC)*, pages 132–139. IEEE Computer Society Press, October 1998.

[19] Nathan S. Shenck and Joseph A. Paradiso. Energy scavenging with shoe-mounted piezoelectrics. *IEEE Micro*, 21(3):30–42, May-June 2001.

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

[21] Mohammed Rahimi, Hardik Shah, Gaurav S. Sukhatme, John Heidemann, and D. Estrin. Studying the feasibility of energy harvesting in a mobile sensor network. In *IEEE Int'l Conference on Robotics and Automation*, 2003.

[22] William J. Kaiser, Gregory J. Pottie, Mani Srivastava, Gaurav S. Sukhatme, John Villasenor, and Deborah Estrin. Networked infomechanical systems (nims) for ambient intelligence. Technical Report CENS Technical Report 31, Center of Embedded and Networked Sensing, UCLA, 2003.

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

[24] Pravin Bhagwat, Bhaskaran Raman, and Dheeraj Sanghi. Turning 802.11 inside-out. In *Workshop on Hot Topics in Networks (HotNets-II)*, November 2003.

[25] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM Press, 2003.

[26] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.

[27] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. An architecture for a power-aware distributed microsensor node. In *IEEE Workshop on Signal Processing Systems (SiPS '00)*, October 2000.

[28] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, pages 181–190, 1998.

[29] Mohamed Younis, Moustafa Youssef, and Khaled Arisha. Energy-aware routing in cluster-based sensor networks. In *Proc. 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'02)*, Fort Worth, Texas, October 2002.

[30] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, volume 1, pages 350–355, Orlando, FL, March, 17-21 2002.

[31] Qun Li, Javed Aslam, and Daniela Rus. Online power aware routing in wireless ad-hoc networks. In *ACM SIGMOBILE*, Rome, Itay, July 2001.

[32] Jae-Hwan Chang and Leandros Tasiulas. Maximum lifetime routing in wireless sensor networks. In *Advanced Telecommunications and Information Distribution Research Program (ATIRP)*, College Park, MD, USA, March 2000.

[33] Morteza Maleki, Karthik Dantu, and Massoud Pedram. Lifetime prediction routing in mobile ad hoc networks. In *Wireless Communication and Networking Conference*, New Orleans, FL, USA, March 2003.

[34] V Rodoplu and Teresa H Meng. Minimum energy mobile wireless networks. In *IEEE International Conference on Communications*, volume 3, pages 1633 – 1639, Atlanta, GA, USA, June 1998.

[35] RG Gallager, PA Humblet, and PM Spira. A distributed algorithm for minimum weight spanning trees. Technical Report LID-P-906-A, Lab. Information Decision Systems, Massachusetts Institute of Technology, October 1979.

[36] Konstantinos Kalpakis, Koustuv Dasgupta, and Parag Namjoshi. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. Technical Report UMBC-TR-02-13, Department of Computer Science and Electrical Engineering,University of Maryland, Baltimore County, August 2002. Accepted for publication in Computer Networks.

[37] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. ACM International Conference on Mobile Computing and Networking (Mobicom)*, pages 70–84. ACM Press, 2001.

[38] L. Shang, R. P. Dick, and N. K. Jha. An economics-based power-aware protocol for computation distribution in mobile ad-hoc networks. In *14th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, November 2002.

[39] P Rong and M Pedram. Extending the lifetime of a netork of battery powered mobile devices by remote processing: A markovian decision based approach. In *ACM Design Automation Conference (DAC)*, Anaheim, CA, USA, June 2003.

[40] Jerry Zhao, Ramesh Govindan, and Deborah Estrin. Residual energy scans for monitoring wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC'02)*, volume 1, pages 356–362, Orange County Convention Center, Orlando, FL, USA, 17-21 March 2002.

[41] Raquel A. F. Mini, Badri Nath, and Antonio A. F. Loureiro. A probabilistic approach to predict the energy consumption in wireless sensor networks. In *IV Workshop de Comunicao sem Fio e Computao Mvel*, So Paulo, Brazil, October 23-25 2002.

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

[43] Thiemo Voigt, Hartmut Ritter, and Jochen Schiller. Utilizing solar power in wireless sensor networks. In *The 28th Annual IEEE Conference on Local Computer Networks (LCN)*, Bonn/Konigswinter, Germany, October 2003.

[44] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *ACM SenSys*, Los Angeles, CA, USA, November 2003.

[45] Manish Bhardwaj and A P Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignments. In *INFOCOM 2002*, pages 1587–1596, New York, USA, June 2002.

[46] M Bhardwaj, T Garnett, and AP Chandrakasan. Upper bounds on the lifetime of sensor networks. In *IEEE International Conference on Communications*, June 2001.

[47] Abhay K. Parekh and Robert G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking (TON)*, 1(3):344–357, 1993.

[48] Abhay K Parekh. *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*. PhD thesis, Massachusetts Institute of Technology, February 1992.

[49] R L Cruz. A calculus for network delay, part I: Network elements in isolation. *IEEE Transactions on Information Theory*, 37:114 – 131, 1991.

[50] R L Cruz. A calculus for network delay, part II: Network analysis. *IEEE Transactions on Information Theory*, 37:132 – 141, 1991.

[51] C Dovrolis, Maruthy Vedam, and Parmesh Ramanathan. The selection of the token bucket parameters in the ietf guaranteed service class. Technical report, Georgia Institute of Technology, 1998.

[52] S Low and P Varaiya. A simple theory of traffic and resource allocation in atm. In *IEEE Infocom*, pages 1633–1637, 1994.

[53] Rex Min, Travis Furrer, and Anantha Chandrakasan. Dynamic voltage scaling techniques for distributed microsensor networks. In *IEEE Computer Society Workshop on VLSI*, pages 43 – 46, April 2000.

[54] Panasonic solar cells technical handbook. http://www.solarbotics.net/library/datasheets/sunceram.pdf.

[55] Low power thermoelectric generators. http://www.dts-generator.com/main-e.htm.

[56] D Rakhmatov, S Vrudhula, and D A Wallach. A model for battery lifetime analysis for organizing applications on a pocket computer. *IEEE Transactions on VLSI systems*, to appear.

[57] D Rakhmatov and S Vrudhula. Energy management for battery powered embedded systems. *ACM Transactions on Embedded Computing Systems*, August 2003.

[58] A.L. Andrady, S.H. Hamid, X. Hu, and A. Torikai. Effects of increased solar ultraviolet radiation on materials. http://www.gcrio.org/UNEP1998/UNEP98p60.html.

[59] Energgizer no. NH15. http://data.energizer.com/. AA Rechargeable NiMH battery Datasheet.

[60] Overview of 1-wire technology and its use. http://www.maxim-ic.com/appnotes.cfm/appnote_number/1796/ln/en.

[61] Tinyos: a component based os for the networked sensor regime. http://webs.cs.berkeley.edu/tos/.

[62] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M.B. Srivastava. Optimizing sensor networks in the energy-density-latency design space. *IEEE Transactions on Mobile Computing (TMC)*, pages 70– 80, January-March 2002.

[63] Bluetooth core specification ver 1v102. http://www.bluetooth.org.

[64] Ieee p802.11 wireless LAN standard. http://grouper.ieee.org/groups/802/11/main.html.

[65] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York, NY, USA, June 2002.

[66] C Intanagonwiwat, R Govindan, and D Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *ACM Mobicom*, August 2000.