Title
An Environmental Energy Harvesting Framework for Sensor Networks

Authors
Aman Kansal
Mani Srivastava

Publication Date
2003

An Environmental Energy Harvesting Framework for Sensor Networks

Aman Kansal and Mani B Srivastava
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, USA
(kansal,mbs)@ee.ucla.edu

May 21, 2003

Abstract

Energy constrained systems such as sensor networks can increase their usable lifetimes by extracting energy from their environment. However, environmental energy will typically not be spread homogeneously over the spread of the network. We argue that significant improvements in usable system lifetime can be achieved if the task allocation is aligned with the spatio-temporal characteristics of energy availability. To the best of our knowledge, this problem has not been addressed before. We present a distributed framework for the sensor network to adaptively learn its energy environment and give localized algorithms to use this information for task sharing among nodes. Our framework allows the system to exploit its energy resources more efficiently, thus increasing its lifetime. These gains are in addition to those from utilizing sleep modes and residual energy based scheduling mechanisms. Performance studies for an experimental energy environment show up to 200% improvement in lifetime.

1 Introduction

Distributed autonomous systems such as sensor-actuator networks are being envisioned to carry out complex task sets without human intervention [23, 21, 3]. The true autonomy of such systems depends on their reliable operation for extended times without maintenance. Energy supply is a major design constraint in these systems and the lifetime is limited by battery supplies. The size of the battery exceeds most other system components in many sensor nodes [12, 14]. One way to alleviate this problem is to provide the network with capabilities to automatically feed itself from its environment, which opens up the possibility of achieving significantly longer lifetimes and reducing the battery size.

Mechanisms exist for compact devices to extract energy from the environment, such as solar power, microbial fuel cells [9], vibrations and acoustic noise [13]. The naive approach would be for each node to scavenge its own energy. This is not enough because the energy availability will typically not be homogeneous at all nodes. There is a potential here to extract more work out of the same energy environment if the task distribution among nodes is adapted to the detailed characteristics of environmental energy availability. We explore this challenge.

1.1 Key Contributions

First, we observe the need to design new algorithms for scheduling and task allotment within the sensor network when nodes are provided with recharging capabilities. Current energy aware task allotment strategies base their decisions on residual energy at each node and do not exploit the fact that energy replenishing capability may be present at some nodes. As far as possible, tasks should be carried out using the environmental energy (or energy from nodes which have ample recharging opportunity) rather than consuming energy from a battery which does not have any recharging opportunity. Specifically, the spatio-temporal properties of the energy supply across the deployment terrain of the network must be exploited for task allotment within the network. We refer to the problem of extracting the maximum work out of a given energy environment as the "harvesting problem."

Second, we make the first step towards solving the harvesting problem by providing a distributed framework, referred to as the environmental energy harvesting framework (EEHF), to

- adaptively learn the energy properties of the environment and the renewal opportunity at each node through local measurements
- make the information available in a succinct form for use in energy aware task assignment such as load balancing, leader elections for clustering techniques, and energy aware communication.

We also suggest modifications to existing load balancing methods to exploit the information provided by EEHF. The details of the energy environment can also be used for incremental deployment in regions of scarcity and adjustment of performance constraints expected from the deployed system. We illustrate EEHF applied to a specific task, that of routing, in an example energy environment.

1.2 Related Work

To our knowledge no ongoing or previous work has addressed the problem of learning the energy environment of a sensor network for modifying the task sharing among network nodes. Here, we mention previous work which helped in proposing a solution to this new problem. Low power design techniques have been suggested for all aspects of sensor network design [15, 10]. These techniques attempt to minimize the energy usage at all levels of system operation. The usability of the network does not depend only on the total energy available in all its nodes but also on how this energy is distributed among them. If a particular section of the network dies, while the remaining sections have abundant energy, the network may no longer be able to provide the service it was deployed for. Thus, in addition to minimizing energy usage, the system lifetime can be increased by modifying the task allotment according to the energy available at network nodes. Some examples of such techniques can be found in [19, 20, 25, 7, 24] for routing and data gathering. The first requirement for these techniques is to get the information about energy availability at the nodes.The remaining energy in a battery can be estimated from its discharge function and measured voltage supplied [20]. Network wide energy scans can be collected [26]. These methods report residual battery sizes and do not include the characteristics of environmental energy. Methods to predict energy consumption [11] have been proposed which can be exploited by EEHF. A method to transfer energy from sources to depleted nodes was proposed in [16], using mobile nodes. EEHF does not require mobile nodes and attempts to distribute the tasks as per the energy availability. The algorithms in [16] can help improve EEHF when mobile nodes are available.

EEHF is a significant development over the methods suggested in the above works. It incorporates the harvesting capability, and presents a unified framework for using these methods according to the network environment.

1.3 Outline

The next section motivates why the framework is important and discusses the issues in its design. Section 3 describes our proposed EEHF and presents modifications in known power management strategies to incorporate its use. Section 4 studies the performance of our framework for a particular task allotment application—routing, in a specific energy availability environment, based on data collected from James Reserve [6]. Comparisons are made with a residual energy based scheme. Section 5 concludes the paper.

2 Design Challenges

We first discuss why there is a need to learn the environment. It seems at the first glance that nodes which have higher environmental energy available will have higher residual battery supplies and just looking at the battery should be sufficient for task allocation. There are two reasons why the knowledge about the environmental energy is explicitly needed.

The first is that the workload in the network may not follow the replenishment cycles: over some time intervals it may happen that the energy consumed at a node with recharging opportunity is more that the energy gained from the environment, thus reducing its residual battery to a smaller size than a node with no recharging. This is very likely once we start allotting more work to nodes with higher battery size. At the instant that their battery depletes to the level of the battery in nodes without recharging opportunity, knowing the environment becomes essential.

The second reason is that knowing only residual energy is not sufficient to decide how much extra energy can be consumed at the energy-rich nodes to save energy at the constrained nodes without jeopardizing the richer node's own lifetime. Consider for example the transmit energy used at every node for maintaining a connected topology.

It is clear that nodes with more environmental energy should use higher transmit power to save energy at other nodes. However, we need to know the environmental energy available at a node to decide how much extra transmit energy can be used by this node without unduely depleting its battery.

2.1 Design Issues

EEHF aims to provide a scalable solution to the harvesting problem. The design challenges here involve learning the environment locally, sharing the learnt information and scheduling tasks for optimal lifetime.

2.1.1 Learning the Environment

The environmental energy at any time will typically be unequal at different points in space and will vary with time. In the case of seismic energy for instance, the vibrations may occur at different locations at different points in time, in an unknown pattern. Solar energy will display diurnal and seasonal patterns. The network application is likely to cause non-uniform energy usage, leading to energy discrepancies.

When taking a tasking decision, a node which is more likely to have excess energy in the future has to be used. Hence, based on the availability (and consumption on tasks beyond the control of the scheduler) till the present time, the future availability in terms of amount of energy and the time duration over which it will be available, need to be predicted. Depending on the patterns in available energy, the algorithm may determine a period over which good predictions can be made. When no useful predictions can be made, the scheme should reduce to residual energy based methods.

2.1.2 Sharing Network-wide Information

The system wide space-time characteristics of both the environmental energy and activity cannot be learnt by a single node. The information has to be gathered and used in a distributed fashion.

The complete environment observed by the network can be described by a set of curves giving time variation of the energy availability at all nodes, for all times. However, considering the bandwidth limitations of low power nodes and scalability issues in networks with a large number of nodes this description would be too large to communicate to and store at all nodes. Thus, appropriate metrics, retaining just the amount of information required for the task sharing methods, are required.

These metrics have to be spread and updated in the network to the extent required for useful task allotment.

After learning the environmental characteristics, the framework has to take scheduling decisions such that the system lifetime is maximized. Apart from redistributing tasks, the framework may provide for redistribution of energy itself, such as through methods in [16] when mobility is available; this is not considered in the present work.

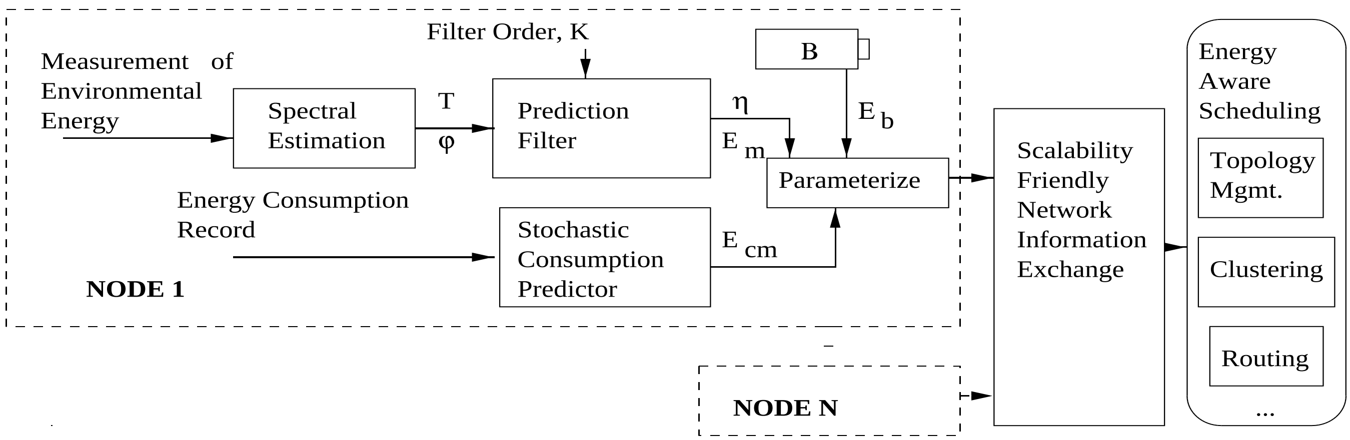

Figure 1: Interactions between various EEHF algorithms

3 Detailed Framework Design

EEHF is a distributed framework for aligning the task distribution with energy availability. Finding the optimal set of parameters to describe the environment from a scheduling perspective and the methods to utilize them is a difficult problem. To obtain initial insights on the feasibility of EEHF, we work with a heuristic set and evaluate its performance.

The following parameters are used to track the characteristics of energy availability:

T: time epoch over which availability prediction is made

E_m: mean energy expected in subsequent T duration

E_{cm}: energy consumed in every T interval on tasks not in control of the scheduler

η: prediction confidence, a number between 0 and 1

ϕ: information regarding when the next recharging opportunity is expected within next T time

E_b: current battery remaining

B: maximum battery size, beyond which environmentally available energy cannot be stored

The challenge is now reduced to finding a single cost C at each node based on the above parameters:

$$C = f(T, E_m, E_{cm}, \eta, \phi, E_b, B) \quad (1)$$

where $f(.)$ is a function to be specified by EEHF. This function $f(.)$ should be such that scheduling decisions based on C automatically select nodes with higher environmental energy to the maximum extent required for optimizing the lifetime. For instance, when

$E_m - E_{cm}$ and η are high at a node the scheduler may use this node even to its last joule as this node will become alive within next T, to save energy at nodes with low E_m.

3.1 EEHF Description

Figure 1 presents the EEHF block diagram. We discuss the purpose of each block for a general environment. As an example, the specific algorithm used to fulfill this purpose in the case of solar power is presented for each block.

3.1.1 Spectral Estimation

This is the first step in attempting to detect a pattern in the availability. The function of this block is to learn the key spectral components in the availability waveform and generate the parameter T, the time duration over which E_m is predicted. If only one key spectral component, f, dominates (is 3dB above the next highest component), we may take $T = 1/f$. The waveform approximately repeats every T duration and the average energy earned in this period can be predicted. Spectral estimation is a developed field and numerous spectral estimation algorithms are available [22] for a wide variety of signals. For the solar energy environment we use the fast fourier transform. This block also keeps track of the phase within the major cycle discovered in the availability waveform. The phase, ϕ, gives information regarding when recharging opportunity is available within the T duration.

3.1.2 Prediction Filter

Given T, we attempt to predict E_m for the duration T in future. For this, the value of E_m in the previous K intervals of duration T may be used. A large variety of adaptive filters such as least mean squares (LMS) filter, normalized LMS [5], and fixed coefficient filters based on autoregressive methods is available for this purpose. For solar energy, we use an autoregressive filter with order $K = 1$, to predict the average E_m, for the T learnt. This block also tracks the error in prediction and assigns a confidence value, η to the prediction. Here, we use $\eta = 1 - |\Delta|/E_m$ if $|\Delta| < E_m$ and $\eta = 0$ otherwise, where Δ is the error in prediction in the previous interval.

3.1.3 Stochastic Consumption Predictor

This block tracks the average consumption, E_{cm}, in every T. Algorithms from [11] may be used here.

3.1.4 Parameterize

This block combines the parameters learnt by the above blocks and the remaining battery E_b into one cost metric. The choice of the function $f(.)$ in equation 1 used for this parametrization is critical.

For solar energy environment, we expect T to be same for all nodes and assume B, the battery size, is also same for all nodes. When all nodes are deriving energy from the same source, ϕ within T would have only small variations based on location which we neglect. Thus, an effective battery, E, and the cost C can be calculated as:

$$E = w_1(E_m - E_{cm})\eta + w_2 E_b \tag{2}$$

$$C = 1/E \tag{3}$$

where w_1 and w_2 are assigned so as to give a much higher weight to replenishable energy. The formula for C is based the one proposed in [20] for residual energy based schemes, $C = 1/(1-G)$, where G is the amount of battery already used. This block may be omitted if the scheduler uses the various parameters learnt by the previous blocks directly for its decisions.

3.1.5 Scalability-friendly Information Exchange

The above blocks learnt the temporal characteristics of the energy availability at each node, locally. However, lifetime optimal scheduling requires global decisions to be made based on this information. This block tries to provide the spatial characteristics of the availability. The exact procedure used will depend on the scheduling algorithm which uses the energy availability information. We propose two approaches for sharing the local information which are both scalable with increasing network size and node density:

1. In-network Averaging: A load balancing scheduler may want to assign loads in proportion to the effective batteries at the nodes. For this, the E calculated at each node involved in the task being scheduled needs to be known to the scheduler. Rather than explicitly transferring the E or C calculated at each node to a central scheduling entity, we propose to circulate the information about the average E and maximum of E among all nodes, E_{av} and E_{max} respectively, throughout the group among which load balancing is to be performed. Scalable methods for calculating the network-wide averages or maxima via in-network processing exist [8]. The nodes can then volunteer to accept a workload proportional to $L = (E - E_{av})/(E_{max} - E_{av})$, when L is positive and go to sleep mode otherwise. Scalability is achieved because instead of distributing the local E learnt at every node, only the network-wide average and maxima is calculated and scalable methods exist for this.

2. Distributed Scheduler: Certain tasking algorithms learn the local costs on their own as required. For a routing algorithm, for instance, we assign the cost C calculated at each node to all one-hop links coming into that node. A distributed route discovery algorithm, such as distributed Bellman-Ford [2], then chooses a minimum cost route without the framework having to explicitly provide for the sharing of information. Such an approach can only be used when the scheduler itself is distributed.

For the case of solar power and the example application studied in simulations – routing, we use the second approach.

These methods put together provide a framework for utilizing the energy harvesting capacity in a distributed fashion.

3.2 Using the Framework

We illustrate the use of EEHF in several classes of schedulers below. The existing versions either do not consider the energy properties of nodes or use only the residual battery information. We modify these to use EEHF for exploiting the harvesting capability of the network.

1. Topology Management: One topology management scheme, STEM [18] duty cycles the node radios between sleep and active states. The connection set-up delay at each hop along a route depends on the duty cycle of the radio, since the sender must wait for the receiver to become active. STEM allots the same duty cycle to each node, dividing the tolerable latency equally among the number of hops. However, it is possible to achieve the same latency if some radios sleep for longer durations and others, with higher E, compensate by sleeping shorter durations. Thus, a topology management scheme like STEM can use the E provided by EEHF to adjust sleep durations, to save batteries at energy constrained nodes.

2. Clustering Algorithms/Load Balancing: Some algorithms form nodes into groups and keep one node active within each group, based on the residual energy. One example is GAF [24]. Extending this to use EEHF is straightforward — instead of using the residual energy, GAF can choose the active node depending on the E calculated by EEHF. Hence when available, only the nodes with environmental availability will be used within each group, increasing the lifetime.

3. Routing: Methods to choose routes based on residual energy can be modified to use the effective battery calculated by EEHF; detailed discussion in section 4.

4. Transmission Power Control: Transmit power of radios can be changed [17] to modify the topology to minimize the maximum power transmitted by any radio. Such algorithms can use EEHF to minimize the maximum power transmitted by nodes having lower E by allowing transmit power to increase at nodes with larger E, thus increasing lifetime.

5. Leader Elections/Hierarchies: Data aggregation algorithms such as [4] form hierarchies within the network where nodes higher up in the hierarchy process more data. EEHF can help decide which nodes should be used for heavier computations.

4 An example Application and Performance Study

4.1 The Energy Environment

We present the use of EEHF for a solar energy based sensor network. We recorded the light intensity variations in a possible deployment region using a CCD camera. These recordings were made in a small region of James Reserve [6] for 39 days (January 6 to February 15, 2003). Intensity variation waveforms, sampled at 10 minute interval, for

2 of these points are plotted in the first two subplots in Figure 2 for 5 days. The time axis is in days. The maximum value is normalized to 1 and this value is assumed to correspond to the charging intensity of the solar cells at noon.

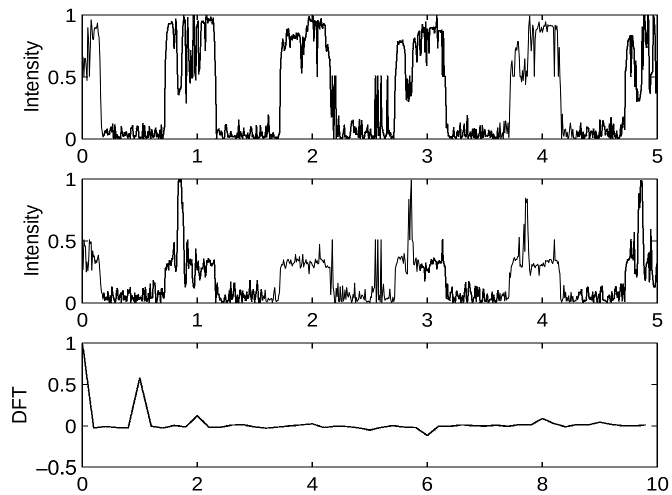

Figure 2: Energy availability waveforms and spectral analysis

The third subplot gives the magnitude spectrum of the first waveform for the first 50 indices. The frequency axis has been converted to day^{-1}. Neglecting the zero frequency, a dominating frequency component is found at $f = 1(day^{-1})$. This gives $T = 1/f = 1(day)$, for use in prediction.

The simulation parameters mirror MICA motes [12] and a mote sized solar-cell, Panasonic BP 376634C [1]. This cell generates about 100mW at noon-time intensity on a clear day. We linearly scale the charging capacity for variations in light intensity, as the charging current varies approximately linearly with intensity for a solar cell, while the voltage stays approximately constant above a certain intensity threshold. Energy is supplied to nodes as per the intensity waveforms recorded above and consumed in data-forwarding. Motes consume 36 mW in active mode and are powered by AA sized batteries having a storage capacity of 1850 mWh, about 50 hours of motes active time.

4.2 Example Application: Routing

We compare the performance of EEHF and a residual energy based scheme, for routing. The simulated sensor network has $N = 100$ nodes in a $100m \times 100m$ region, with nodes having a radio range of $20m$. The nodes are assumed randomly placed in the region for which light intensity variations have been collected, distributing a fraction p in well-lit regions and $(1 - p)$ in dark regions. The light intensity waveforms at these randomly chosen pixels are picked up from the CCD data mentioned in section 4.1.

Energy received in every T (value of T as found in section 4.1) is recorded. The autoregressive prediction filter used to estimate E_m is $E_m(n+1) = \alpha E_{mm}(n) + (1 -$

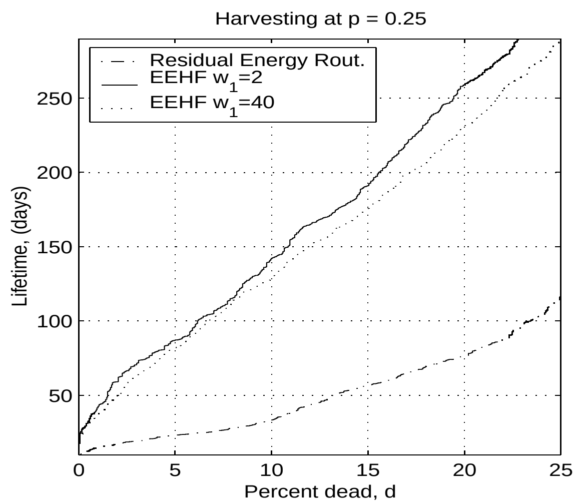

Figure 3: Lifetime improvement with EEHF, over residual energy based routing, $p = 0.25$.

$\alpha)E_m(n)$ with filter coefficient $\alpha = 0.9$, where $E_{mm}(n)$ is the measured value of energy received in the current interval, and n is the discrete time index (initial condition: $E_m(0) = 0$). In equation 2, $w_2 = 0.1$ and $w_1 = 2, 40$. These weights cause nodes with more environmental energy to be used more frequently in EEHF. E_{cm} is assumed same for all nodes across the network and hence not used in calculating the cost for routing.

Traffic sources and destinations are chosen uniformly randomly among the nodes. A traffic flow lasts between a chosen pair for a fixed duration. This duration is much smaller than the maximum flow duration feasible with the battery size as we expect battery sizes to be chosen to support more than one flow in most applications. The route is chosen at the start of the flow and is not changed during the flow.

The residual energy based scheme [20] uses Bellman-Ford routing with costs assigned inversely proportional to residual batteries. Thus, nodes with high residual energy have lower cost and are used more often. EEHF uses the same routing scheme but chooses nodes based on the E from equation 2 instead of residual energy.

Figure 3, 4 and 5 present the simulation results for $p = 0.25, 0.5$ and 0.75 respectively. The results are averaged over 20 random topologies, to suppress any artifacts due to a particular topology. We assume that the network loses utility after the first $d\%$ nodes are dead, where d depends on the application. We plot the lifetime in days for $d \in (0, 25)$. It can be observed that lifetime is more than doubled at low p, as EEHF is able to select nodes with higher environmental energy. The increase in lifetime here is due to the first reason for the need of harvesting presented in the beginning of section 2. At high p (Figure 5), after $(1 - p)$ nodes die, EEHF and residual energy based scheme perform similar as all then nodes have similar environmental energy and residual batteries dominates. These simulations are a sanity check for EEHF and illustrate that

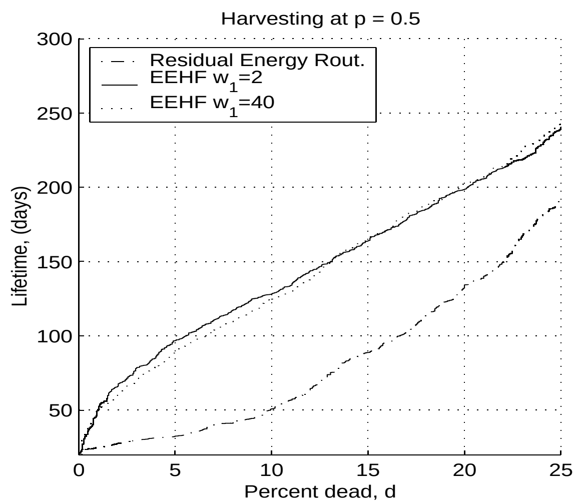

Figure 4: Lifetime improvement with EEHF, over residual energy based routing, $p = 0.5$.

using environmental energy aware algorithms has the potential to increase lifetimes significantly.

5 Conclusions

This paper introduced the problem of energy harvesting in sensor networks. A framework was presented for handling the major challenges involved and an example study showed that the proposed framework is able to utilize the extra knowledge about the environment to increase system lifetime.

EEHF makes the first step towards the development of an optimal set of methods to completely align the task distribution in the network with the energy availability. Future work includes determining the maximum work that can be extracted from a given energy environment and estimating achievable bounds on quality of service in terms of communication delay, sensing fidelity, and other performance measures, if the network has to survive eternally on the available energy resources. Then, to solve the harvesting problem, distributed methods to achieve the maximum lifetime need to be developed and tested on a real system implementation. Facilities to redistribute energy apart from tasks may also be incorporated into EEHF.

6 Acknowledgments

This paper is based on research funded in part by DARPA PAC/C program through AFRL, and by the ONR AINS program under the Minuteman Project. The views

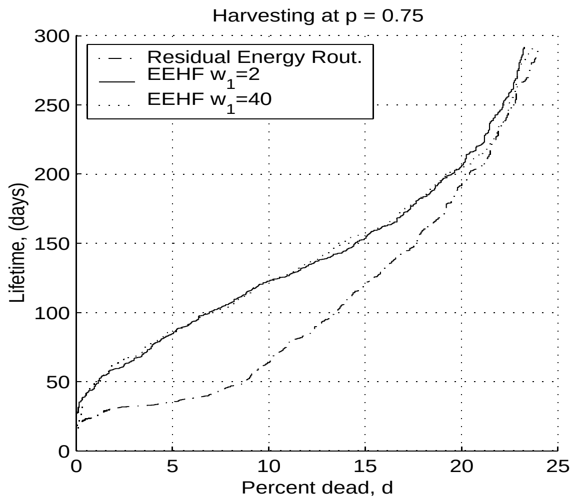

Figure 5: Lifetime improvement with EEHF, over residual energy based routing, $p = 0.75$.

expressed in the paper are authors' only, and do not necessarily represent those of the funding agencies.

References

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

[2] D. Bertsekas and R. Gallagher. *Data Networks*. Prentice Hall, 2nd edition, 1992.

[3] D. Estrin, R. Govindan, and J. Heidemann. Embedding the internet: introduction. *Communications of the ACM*, 43(5):38–41, 2000.

[4] I. Gupta, R. van Renesse, and K. P. Birman. Scalable fault-tolerant aggregation in large process groups. In *Int'l Conference on Dependable Systems and Networks (DSN 01)*, Goteborg, Sweden, 2001.

[5] S. Haykin. *Adaptive filter theory*. Prentice Hall, 1986.

[6] James San Jacinto Mountains Reserve. http://www.jamesreserve.edu.

[7] K. Kalpakis, K. Dasgupta, and P. 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.

[8] S. R. Madden, R. Szewczyk, M. J. Franklin, and D. Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *Proc. Workshop on Mobile Computing and Systems Applications*, 2002.

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

[10] 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.

[11] R. A. F. Mini, B. Nath, and A. 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.

[12] Mica wireless measurement system. Datasheet, Crossbow Technology Inc. http://www.xbow.com/Products/ Product_pdf_files/Wireless_pdf/MICA.pdf.

[13] J. M.Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *IEEE Computer*, pages 42–48, July 2000.

[14] K. Pister. Smart dust: Autonomous sensing and communication in a cubic millimeter. http://robotics.eecs.berkeley.edu/˜
pister/SmartDust/.

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

[16] M. Rahimi, H. Shah, G. S. Sukhatme, J. 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.

[17] R. Ramanathan and R. Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM (2)*, pages 404–413, 2000.

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

[19] R. C. Shah and J. 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.

[20] S. Singh, M. 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.

[21] M. Srivastava, R. Muntz, and M. Potkonjak. Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving enviroments. In *Proc. ACM MobiCom*, July 2001.

[22] P. Stoica and R. Moses. *Introduction to Spectral Analysis*. Prentice Hall, 1997.

[23] D. Tennenhouse. Proactive computing. *Communications of the ACM*, 43(5):43–50, 2000.

[24] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proc. ACM Mobicom*, pages 70–84. ACM Press, 2001.

[25] M. Younis, M. Youssef, and K. 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.

[26] J. Zhao, R. Govindan, and D. 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.