Performance Evaluation of Resource Allocation Policies for Energy Harvesting Devices

Maria Gorlatova*, Andrey Bernstein*†, Gil Zussman*
*Electrical Engineering, Columbia University, New York, NY, 10027
†Electrical Engineering, Technion, Haifa, Israel, 32000
Email: mag2206@columbia.edu, andreyb@techunix.technion.ac.il, gil@ee.columbia.edu

***Abstract*—We focus on resource allocation for energy harvesting devices. We analytically and numerically evaluate the performance of algorithms that determine time fair energy allocation in systems with *predictable* and *stochastic* energy inputs. To gain insight into the performance of networks of devices, we obtain results for the simple cases of a single node and a link. Due to the need for *low complexity* algorithms, we focus on *simple policies* (some of which proposed in the past as heuristics) and analytically derive performance guarantees. We also evaluate the performance via simulation, using real-world energy traces that we collected for over a year, and in a testbed of energy harvesting devices developed within the EnHANTs project.**

***Index Terms*—Energy harvesting, ultra-low-power networking, indoor radiant energy, measurements, energy-aware algorithms**

I. Introduction

Recent advances in the areas of solar, piezoelectric, and thermal energy harvesting, and in ultra-low-power wireless communications will soon enable the realization of energy harvesting wireless devices. When networked together, they can compose rechargeable sensor networks [5], [15], networks of computational RFIDs [11], and Energy Harvesting Active Networked Tags (EnHANTs) [10]. Such networks will find applications in various areas, and thus *networking energy harvesting devices* has lately been gaining attention. Work in this area includes design of energy-harvesting aware algorithms [5], [6], [9], [12]–[17], [19], development of energy harvesting devices, and characterizations of different energy sources [9], [11] (for reviews of related work see [5], [9], [10]).

Energy sources may have different characteristics. We consider the *predictable profile energy model* [5], [9], [12], [14] in which ideal energy profiles that accurately represent the future are available, and the *stochastic energy model* [6], [9], [13] in which the energy availability can be modeled by a stochastic process. Examples of the latter include a mobile device harvesting light energy, a floorboard that gathers energy when stepped on, and a solar cell in a room where lights go on and off as people enter and leave. We mostly focus on the case of *stationary* (i.i.d.) process to describe the energy availability, for which we provide optimal spending policies. In addition, we consider *non-stationary* stochastic models, where the energy availability characteristics may change arbitrarily with time. In this case, we propose to apply online learning algorithms, such as [20]. In our model, we also consider linear energy storage device (i.e., a battery) and a non-linear device (i.e., a capacitor).

Energy availability may have high time-variability [11], [12], [19], and therefore, we aim to, as much as possible, allocate the varying energy *in a uniform way with respect to time*. For that, we use the *lexicographic maximization* and the *network utility maximization* frameworks, which are typically applied to achieving fair resource allocation among different nodes rather than among different time slots. Once the energy spending rates are determined by these frameworks, they can be converted to duty cycle, sensing rate, or communication rate.

Energy harvesting shifts the nature of energy-aware protocols from *minimizing* energy expenditure to *optimizing* it over time. Therefore, the resource allocation problems are highly complex [9]. On the other hand, since the devices are resource constrained, there is a need for *very* low (computation and communication) complexity algorithms. While some attempts have been made to develop algorithms for specific types of networks (e.g., directed graphs [15] and trees [5]), most previous work on implementable algorithms focused on a single node or a link [12], [13], [16], [19]. In order to provide insight into the development of low complexity algorithms for a network, we focus in this work on a single node and a link. We analytically and numerically evaluate the performance of *approximate and heuristic* policies, some of which are proposed in [5], [13], [15], [16]. In particular, for a *single node*, we study the following policies:

- **Optimal** ***(OPT)*** policies for both the *predictable profile* and the *stationary stochastic* models serve as a benchmark for other policies. For the stationary stochastic case, we use a *Markov Decision Process (MDP)*, prove that *energy state discretization* can be applied, and provide bounds on the performance degradation due to discretization.
- **Spend-What-You-Get** ***(SG)*** policy – within a time slot a node spends the expected energy input for that slot, and therefore, the complexity is very low (similar policies are proposed in [15], [16]). For both the deterministic and stationary stochastic models, we provide performance guarantees.
- **Constant Rate** ***(CR)*** policy – a node spends energy at a constant rate in all time slots, resulting in very low complexity (it is proposed in [5]). For the *predictable profile* model, we provide a performance guarantee.

978-1-61284-824-2/11/$26.00 ©2011 IEEE

- **Energy Storage Threshold-based *(THR)*** policy – a set of energy storage thresholds and corresponding rates are chosen, and the node determines the spending rates based on the current storage level (similar policies are proposed in [13], [16]). We study the parameter settings for the *stationary stochastic* model.
- **Energy Storage-Linear *(SL)*** policy – the spending rate is a *linear function* of the energy storage level. We study the parameter settings for the *stationary stochastic* model.

For *links* (node-pairs) we study the following policies:

- **Optimal *(OPT)*** policies (under which nodes need to exchange their parameters) for both energy models.
- **Decoupled Rate Control *(DRC)*** policies – the nodes first determine *independently* their spending rates, and then jointly calculate the data rates (similar approaches are used in [5], [15]). We examine a few versions:
  - **Node-optimal DRC *(DRC-NOPT)*** – the nodes' spending rates are determined according to the *optimal single-node policy*. We provide a performance guarantee for the *predictable profile* model.
  - ***DRC-SG*, *DRC-CR*, etc.** – one of the above-described policies is used to solve the two single-node problems. These policies are evaluated numerically for the *predictable profile* model.

Within the *Energy Harvesting Active Networked Tags (EnHANTs)* project [10] we have been developing energy harvesting devices and characterizing the availability of indoor ambient light energy. To evaluate the performance of the algorithms, we use simulations based on traces that we collected for over a year [9] as well as experiments with the EnHANTs prototypes [8]. In many of the considered cases, the simple policies perform very well.

This paper is organized as follows. Section II describes the model. Sections III and IV present the analytical results for the *predictable profile* and *stationary stochastic* energy models, respectively. In Section V we discuss the *non-stationary* stochastic model and the corresponding online learning algorithm. Section VI presents *energy trace-based* and *testbed* evaluation of the presented policies. We summarize and discuss future work in Section VII. Due to space constraints, the proofs are omitted and can be found in [7].

II. Model and Preliminaries

We focus on *discrete-time* models, where the time axis is separated into K slots, and a decision is made at the beginning of a slot i ($i = \{0, 1, ..., K-1\}$). We denote the energy storage capacity by C and the amount of energy stored by $B(i)$ ($0 \leq B(i) \leq C$). We denote the initial and the final energy levels by B_0 and B_K. The energy spending rate is denoted by $s(i)$. The amount of energy a device has access to is denoted by $D(i)$, which can be a given value or a random variable. The *effective* amount of energy a device can harvest from the environment is denoted by $Q(i)$. In general, $Q(i)$ may depend both on the available energy $D(i)$ and on the current energy level: $Q(i) = q(D(i), B(i))$ and hence can be non-linear in $D(i)$ (e.g., the *non-linear energy storage model* applies to a capacitor). For a *linear energy storage model* device (such as a *battery*), $q(D(i), B(i)) = D(i)$ and in general $Q(i) \leq D(i)$ [9]. The 'storage evolution' for the models we consider can be expressed as:

TABLE I
Nomenclature.

$D(i)$	Environmental energy (J/slot)
K	Number of slots
C	Energy storage capacity (J)
$B(i)$,B_0,B_K	Energy storage state, initial, and final levels (J)
$s(i)$	Energy spending rate (J/slot)
$Q(i)$	Effective energy harvested (J/slot)
h	Quantization resolution (J)
$r(i)$	Data rate (bits/slot)
$c_{\text{tx}}, c_{\text{rx}}$	Energetic costs to transmit and to receive (J/bit)
$U(\cdot)$	Utility function
Z	Objective function value
T	Node downtime

$$B(i) = \min\{B(i-1) + Q(i-1) - s(i-1), C\} \quad (1)$$

Note that for the stochastic energy model, we consider *quantizing* the above energy-related parameters, and denote the quantization resolution by h.

We consider a *single node* and a *node pair (link)*. We denote the endpoints of a link by u and v, the effective amount of energy each node can harvest by $Q_u(i)$ and $Q_v(i)$, and their data rates by $r_u(i)$ and $r_v(i)$. For a *single node* we optimize the energy spending rate vector $s(i)$, which provide inputs for determining *duty cycle*, *sensing rate*, or *communication rate*. For a *link*, we optimize either the spending rates $s_u(i)$ and $s_v(i)$ or the communication rates $r_u(i)$ and $r_v(i)$. We denote the costs to transmit and receive bits by c_{tx} and c_{rx}. The constraints relating node energy spending rates and data rates on a link for slot i are:

$$c_{\text{tx}} r_u(i) + c_{\text{rx}} r_v(i) \leq s_u(i), c_{\text{tx}} r_v(i) + c_{\text{rx}} r_u(i) \leq s_v(i). \quad (2)$$

We focus on time-uniform (time-fair) allocation of resources, and use the *lexicographic maximization* and *network utility maximization* frameworks. In the former, we *lexicographically maximize* an energy spending rate vector (for a stand-alone node), or a data rates vector (for a link). In the latter, we maximize the overall utility, where the utility function $U(\cdot)$ for each individual assignment is *concave* and *non-decreasing*. For deriving numerical results, we use $U(\cdot) = \log(1+(\cdot))$. We denote the total objective function value by Z (i.e., $Z = \sum_i U(\cdot)$), and use subscripts to indicate the policy under which Z was obtained (e.g., Z_{OPT} for the *OPT* policy and Z_{CR} for the *CR* policy). As another performance measure, we consider the *downtime* of a node and a link, namely, the fraction of slots the node or the link do not spend energy. We denote the downtime of a node by $T = |\{i | s(i) = 0\}|/K$ and the downtime of a link by $T^L = |\{i | r_u(i) = 0, r_v(i) = 0\}|/K$.

III. Predictable profile Energy Model

In this section, we analyze various policies for a *single node* model and discuss a *link model*. Section VI provides numerical results demonstrating the performance of the policies based on real-world energy traces.

A. Single Node

The optimal solution for a single node can be obtained by solving the following problem [9].

Time Fair Utility Maximization (TFU) Problem:

$$\max_{s(i)} \left\{ Z \triangleq \sum_{i=0}^{K-1} U(s(i)) \right\} \tag{3}$$

subject to:

$$s(i) \leq B(i) \; \forall \; i \tag{4}$$

$$B(i) \leq B(i-1) + Q(i-1) - s(i-1) \; \forall \; i \geq 1 \tag{5}$$

$$B(0) = B_0; \; B(K) \geq B_K; B(i) \leq C \; \forall \; i \tag{6}$$

$$B(i), s(i) \geq 0 \; \forall \; i \tag{7}$$

We now provide bounds on the optimal solution as well as an approximation ratio for the *CR* policy. Observation 1 applies to both *linear* and *non-linear* energy storage models, while Observations 2 and 3 apply to the *linear* energy storage model (the proofs can be found in [7]).

Observation 1:

$$Z_{\text{OPT}} \leq K \cdot U\left(\left(B_0 - B_K + \sum_{i=0}^{K-1} D(i)\right)/K\right).$$

Observation 2: The total energy allocated by the optimal solution is $\sum_i \min(Q(i), C) + B_0 - B_K$. The optimal solution will allocate all available energy if $C > \max(Q(i))$.

Observation 3: Under the *CR* policy, for[1] $B_K = B_0 \leq \sum_i Q(i)$ and $U(s) = \log(1+s)$, $Z_{CR} \geq Z_{OPT} \cdot \left(\frac{B_0}{\sum_{i=0}^{K-1} Q(i)}\right)$.

The following proposition provides an approximation ratio for the *SG* policy for both *linear* and *non-linear* energy storage models.

Proposition 1: Under the *SG* policy and for $U(s) = \log(s+M)$, $Z_{SG} \geq Z_{OPT} \cdot \log(G(Q'))/\log(\overline{Q'})$, where M is a constant, $\overline{(\cdot)}$ and $G(\cdot)$ denote the *arithmetic* mean and the *geometric* mean of a sequence, and $Q'(i) \triangleq Q(i) + M \; \forall i$.

For example, consider a case of $Q(i)$ such that L samples of $Q(i)$ are equal to some non-zero constant, and the rest are equal to zero. Such $Q(i)$ may correspond to the case where the indoor lights are on for a portion of the day. Using Proposition 1, we demonstrate that for $B_K = B_0$ (*energy neutrality* [12]) and for $U(s) = \log(1+s)$, the *SG* policy is a K/L-*approximation algorithm* (for instance, if the indoor lights are on for 8 hours per day, the *SG* policy is a 3-approximation algorithm). Denote $\widehat{Q} = \sum_i Q(i)$. For $U(s) = \log(1+s)$, $U(Q(i) = 0) = 0$, and thus:

$$\frac{Z_{OPT}}{Z_{SG}} \leq \frac{\sum(U(\hat{Q}/K))}{\sum(U(\hat{Q}/L))} = \frac{K \cdot U(\widehat{Q}/K)}{L \cdot U(\widehat{Q}/L)}$$

$$= \frac{K}{L} \frac{\log(\hat{Q}/K+1)}{\log(\hat{Q}/L+1)} \leq \frac{K}{L}.$$

The last inequality stems from the fact that $K > L$, thus $\hat{Q}/K < \hat{Q}/L$, and hence $\log(\hat{Q}/K+1)/\log(\hat{Q}/L+1) < 1$. Therefore, K/L is the upper bound. This bound is tight, as for $\hat{Q} \to \infty$ we can demonstrate that

$$\lim_{\hat{Q}\to\infty} \frac{K}{L} \frac{\log(\hat{Q}+K) - \log(K)}{\log(\hat{Q}+L) - \log(L)} = \lim_{\hat{Q}\to\infty} \frac{K}{L} \frac{\hat{Q}+K}{\hat{Q}+L} = \frac{K}{L}.$$

B. Node Pair (Link)

The optimal solutions for a link can be obtained by solving the following problems [9].

Link Time Fair Utility Maximization (LTFU) Problem:

$$\max_{r_u(i), r_v(i)} \sum_{i=0}^{K-1} [U(r_u(i)) + U(r_v(i))] \tag{8}$$

s.t. : constraints (2) $\forall$ i; u, v : constraints (4) – (7)

Link Time Fair Lexicographic Assignment (LTFL) Problem:

Lexicographically maximize:

$$\{r_u(0), ..., r_u(K-1), r_v(0), ..., r_v(K-1)\} \tag{9}$$

s.t. : constraints (2) $\forall$ i; u, v : constraints (4) – (7)

The results in this section apply to the *linear* energy storage model. First, we show below that under specific conditions the solutions to both problems are equal.

Proposition 2: When $c_{\text{tx}} = c_{\text{rx}}$, the *LTFL* problem and the *LTFU* problem have the same solution.

We now examine the performance of the following set of algorithms.

Decoupled Rate Control (DRC) Algorithms: For a given link (u, v), the algorithms first determine $s_u(i)$ and $s_v(i)$ for every slot i according to some single-node policy, optimal (*DRC-NOPT*), or approximate (i.e., *DRC-SG*, *DRC-CR*). Then, for each slot i, under constraints (2), the algorithms obtain a solution to

$$\max_{r_u(i), r_v(i)} \{U(r_u(i)) + U(r_v(i))\}. \tag{10}$$

Small per-slot problem (10) can be easily solved. For example if $c_{\text{tx}} = c_{\text{rx}}$, the solution to (10) is

$$r_u(i) = r_v(i) = \min(s_u(i), s_v(i))/(c_{\text{tx}} + c_{\text{rx}}). \tag{11}$$

Fig. 1 shows schematically the difference between solving link problems optimally and applying the *DRC* algorithms. Now let $\tilde{Z}$ denote the solution of (10) for $s_u(i) \leftarrow [B_{0,u} - B_{K,u} + \sum_i Q_u(i)]/K$ and $s_v(i) \leftarrow [B_{0,v} - B_{K,v} + \sum_i Q_v(i)]/K$. For example, for the case of $c_{\text{tx}} = c_{\text{rx}}$,

$$\tilde{Z} = 2 \cdot U\Big(\frac{1}{c_{\text{tx}} + c_{\text{rx}}} \min\big\{[B_{0,u} - B_{K,u} + \sum_i Q_u(i)]/K,$$

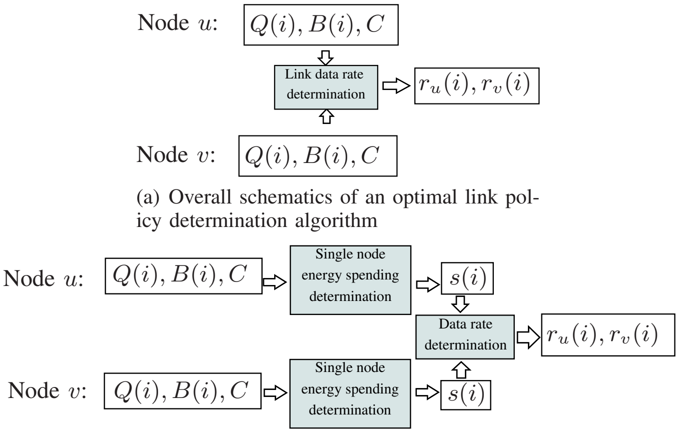
$$[B_{0,v} - B_{K,v} + \sum_i Q_v(i)]/K\big\}\Big).$$

We then have the following.

Proposition 3: $Z_{OPT} \leq K \cdot \tilde{Z}$.

The next Proposition implies that the *DRC-NOPT* policy

[1] Namely, under *energy neutrality* [12], with a relatively small energy storage.



(a) Overall schematics of an optimal link policy determination algorithm

(b) Overall schematics of the *DRC* algorithm

Fig. 1. Comparison of an optimal link policy determination and the *DRC* algorithms.

obtains the optimal solution to the *LTFL* problem for a link (u, v) in which u *and* v *have the same energy parameters* $Q(i), C, B_0$ and B_K.

Proposition 4: DRC-NOPT solves the *LTFL* problem optimally, if for all slots i, node-optimal $s_u(i) \le s_v(i)$.

The following observation discusses the downtime under the *DRC-SG* policy.

Observation 4: Under the *DRC-SG* policy, $\max[T_u, T_v] \le T^L_{u,v} \le T_u + T_v$.

For example, consider a case where $Q_u(i)$ and $Q_v(i)$ are vectors with L non-zero entries. For a (u, v) where $Q_v(i) = Q_u(i)\ \forall\ i$ ($Q_u(i)$ and $Q_v(i)$ are *synchronized*), $T^L_{u,v} = (K - L)/K$. On the other hand, for a (u, v) where $Q_v(i)$ is *shifted* with respect to $Q_u(i)$, $T^L_{u,v}$ can be as high as $2 \cdot (K - L)/K$.

IV. Stochastic Energy Models

We now study models in which the energy harvested in slot i is a *random process* $\{D(i)\}$. We examine the model of a single node with $\{D(i)\}$ *i.i.d.* random variables. We let D denote the "representative" variable for $D(i)$ and p_D denote its *probability density function* (pdf). In addition, we also briefly discuss the extension of the model to a *link*. In this Section we focus on the the *linear storage model* (i.e., $q(d, b) = d$).

A. Single Node – Optimal Policies and Dicretization Bounds

We formulate the problem as an average cost Markov Decision Process (MDP). Let $\mathcal{B} = [0, C]$ and $\mathcal{S} = [0, C]$ denote the state and action spaces of the MDP, respectively. For any $b \in \mathcal{B}$ and $s \in \mathcal{S}$, the transition density is denoted by $p(\cdot|b, s)$. It determines the next energy storage level $B(i+1)$ given that the current energy storage level is $B(i) = b$ and the spending rate is $s(i) = s$. This transition density is determined by p_D and (1). A policy π is a collection of decision rules $\pi_i : \mathcal{B}^i \times \mathcal{S}^{i-1} \to \Delta(\mathcal{S})$ which at each time i prescribe a probability distribution over the actions ($\Delta(\mathcal{S})$ denotes the probability simplex over the set $\mathcal{S}$). The goal is to find an optimal policy, which maximizes the average utility. In particular, let[2] $\lambda_\pi(b) \triangleq \lim_{K\to\infty} \mathbb{E}_\pi\left(\sum_{i=0}^{K-1} U(s(i))\right)/K$ denote the asymptotic expected average utility obtained by starting from state $B_0 = b$ and using a given policy π. The optimal average utility is then $\lambda^*(b) \triangleq \sup_\pi \lambda_\pi(b)$. It is well known (i.e., [18]) that under certain *ergodicity* (or *mixing*) conditions, the optimal average utility does not depend on b. In our case, we use the following mixing condition.

Assumption 4.1 (Mixing): There exists a scalar $\rho \in (0, 1]$ and a measure ν with $\nu(\mathcal{B}) \ge \rho$ such that $p(A|b, s) \ge \nu(A),\ \forall A \subseteq \mathcal{B}, (b, s) \in \Gamma$.

For our problem, we prove the following.

Lemma 4.1: If when $B(i) = C$, $s(i) \ge \alpha > 0$ holds for some α, Assumption 4.1 is satisfied with $\nu(y) \triangleq \min_{(b,s)\in\Gamma} p_D(y - b + s),\ y \in \mathcal{B},\ \rho \triangleq \int_{\mathcal{B}} \min_{(b,s)\in\Gamma} p_D(y - b + s)dy > 0$.

In the view of Lemma 4.1, we let

$$\Gamma \triangleq \{(b, s) \in \mathcal{B} \times \mathcal{S} :\ \max(b - C + \alpha, 0) \le s \le b\} \quad (12)$$

denote the set of *admissible* state-action pairs.

Under the mixing condition, an optimal policy is *deterministic Markov stationary* policy $\pi^* : \mathcal{B} \to \mathcal{S}$ and can be found by solving the *optimality equation* $\lambda + J(b) = \mathcal{T}J(b),\ b \in \mathcal{B}$, where $\mathcal{T}$ is *Bellman's operator*, defined for any bounded function J as $\mathcal{T}J(b) = \max_{s\in\mathcal{S}} \left\{U(s) + \int_{\mathcal{B}} p(b'|b, s)J(b')db'\right\}$. Specifically, a solution (λ^*, J^*) of the optimality equation is such that $\lambda^*(b) \equiv \lambda^*$ and an optimal policy is given by $\pi^*(b) = \text{argmax}_{s\in\mathcal{S}} \left\{U(s) + \int_{\mathcal{B}} p(b'|b, s)J^*(b')db'\right\}$. However, since our state and action spaces are infinite, there is no practical algorithm to solve the optimality equation. To address this, we *discretize* the state and action spaces *uniformly*, using a fixed discretization parameter h. We denote the obtained finite spaces by $\mathcal{B}_h$ and $\mathcal{S}_h$. In particular, if $b \in \mathcal{B}_h$, it is a multiple of h, and similarly for $\mathcal{S}_h$. For any $b \in \mathcal{B}$, we let $x_b \in \mathcal{B}_h$ denote the *representative* point of b in $\mathcal{B}_h$ (which is the closest point to b in $\mathcal{B}_h$).

The discretized set of admissible state-action pairs is then

$$\begin{aligned}\Gamma_h \quad \triangleq \quad & \Big\{(b, s) \in \mathcal{B} \times \mathcal{S}_h :\ |s - s_b| \le h/2 \\ & \text{for some } \max(x_b - C + \alpha, 0) \le s_b \le x_b\Big\}.\end{aligned}$$

Finally, the transition function in the discretized model is: $p_h(b'|b, s) \triangleq p(x_{b'}|b, s)/\int_{\mathcal{B}} p(x_y|b, s)dy$. The corresponding Bellman's operator is then $\mathcal{T}_hJ(b) = \max_{s\in\mathcal{S}_h} \left\{U(s) + \int_{b'\in\mathcal{B}} p_h(b'|b, s)J(b')\right\}$. It is easy to see that this operator returns a *simple* function for any given function J. Moreover, the solution J^*_h of the optimality equation

$$\lambda_h + J_h(b) = T_hJ_h(b),\ b \in \mathcal{B} \quad (13)$$

is also a simple function. The solution (λ^*_h, J^*_h) can be found using value/policy iteration algorithms or linear programming (see [18] for details).

[2] $\mathbb{E}_\pi$ denotes the expectation with respect to the probability law induced by the MDP while using policy π, and $\{s(i)\}$ are the spending rates under this policy.

We use the results in [2] to provide the performance bounds, due to the introduced discretization process. To use these results, in addition to the mixing condition (Lemma 4.1), our MDP model should satisfy the following continuity condition.

Assumption 4.2 (Lipschitz Continuity): There exists a constant $\beta > 0$ such that $|U(s) - U(s')| \leq \beta\,|s - s'|$, $\|p(\cdot|b,s) - p(\cdot|b',s')\|_v \leq \beta\,\|(b,s) - (b',s')\|_\infty$, for all $(b,s),(b',s') \in \Gamma$, where $\|\cdot\|_v$ is the total variation norm.

The first part of Assumption 4.2 can be satisfied by choosing an appropriate utility function. Let β_U denote its continuity constant. For the second part, we impose the following on the probability distribution of the representative random variable D.

Assumption 4.3: Suppose that there exists a finite constant $D_{\max}$ such that the variable D takes values in the interval $[0, D_{\max}]$. Let $P_{\max} \triangleq \max_{d \in [0,D_{\max}]} p_D(d)$. Moreover, assume that there exists a finite constant $\beta_D > 0$ such that $|p_D(d) - p_D(d')| \leq \beta_D\,|d - d'|\,,\ \forall 0 \leq d, d' \leq D_{\max}$.

Lemma 1: Under Assumption 4.3, there exists $\beta = \beta(\beta_U, \beta_D) > 0$ such that Assumption 4.2 is satisfied. In particular, $\beta = \max\{\beta_U, 2(C\beta_D + P_{\max})\}$.

Hence, the following theorem bounds the distance of the optimal average reward λ_h^* in the discretized model from the optimal average reward λ^*. This theorem is in fact an application of Theorem 4.3.5 in [2] to our case.

Theorem 1: Under Assumption 4.3, there exists $\bar{h} > 0$ and β_λ (depending only on β of Assumption 4.2 and ρ of Assumption 4.1) such that for all $h \in (0, \bar{h}]$, we have that $|\lambda^* - \lambda_h^*| \leq \beta_\lambda h$.

Proof: By Lemma 1, Assumption 4.2 is satisfied. By Lemma 4.1, Assumption 4.1 is satisfied as well. Thus, the proof follows from Theorem 4.3.5 in [2]. The exact expressions for the parameters β_λ and $\bar{h}$ are obtained from the proofs of Theorems 2.4.1 and 2.4.2 therein. ■

An optimal policy π_h^* (in the discretized model) may be computed *offline*. Therefore, the actual choice of the spending rate by a device can be done by using the precomputed function $\pi_h^* : \mathcal{B}_h \rightarrow \mathcal{S}_h$. The quantized policies are used to derive numerical results that appear in Section VI.

B. Single Node – Bounds and Heuristic Policies

We now provide some analytical insights into the behavior of the optimal and the *SG* policies for the stochastic energy model. The following observations apply to both energy storage models.

Observation 5: $\mathbb{E}(Z_{OPT}) \leq U(\mathbb{E}(D))$.

Observation 6: $\mathbb{E}(Z_{SG}) = \mathbb{E}(U(Q))$.

We also consider *energy storage state-based* policies, namely the *THR* and the *SL* policies.

- *THR* policy: for a set of storage state thresholds $[B_1, B_2, ...B_T]$ and a set of constants spending rates $[s_1, s_2, ...s_T]$, $s_{THR}(i) \leftarrow 0\ \forall\ B(i) \leq B_1; s_{THR}(i) \leftarrow s_1\ \forall\ B_1 < B(i) \leq B_2; ...; s_{THR}(i) \leftarrow s_T\ \forall\ B(i) > B_T$. That is, for example, for $T = 1$, the *THR* is an ON-OFF policy, and for $T = 2$ is is a bi-level policy.
- *SL* policy: $s_{SL}(i) \leftarrow \alpha_{SL} \cdot [B(i)/C]$ for some parameter α_{SL}.

These policies require choosing parameters, and the policies' performance heavily depends on the choice of the parameters. For policies relying on a small parameter set, simple brute-force algorithms can be used to select the best ones. Consider, for example, a *THR* policy with $T = 1$. A simple algorithm to find the best values for s_1 and B_1 is as follows. For each possible B_1, the algorithm considers all feasible values of s_1, and for each $\{B_1, s_1\}$ combination the algorithm calculates the transition probabilities, determines the stationary probabilities of the states, and calculates Z, choosing the $\{B_1, s_1\}$ combination that maximizes Z. For every state in the state space, the algorithm needs to compute the transition probabilities and the resulting stationary storage state probabilities. However, the state space the algorithm considers is relatively small, $O(|C/h|^2)$. In a similar manner, the *SL* policy parameter α_{SL} can be computed by going through at most $O(|C/h|)$ possible α_{SL} values.

Section VI demonstrates the performance of different policies using real-world traces.

C. Link Model

The MDP formulation can be extended to a *link* (u, v) as follows. We let $D(i) \triangleq (D_u(i), D_v(i))$ denote the energy harvested in slot i by both devices. We let $D \triangleq (D_u, D_v)$ denote the "representative" variable for $D(i)$ and p_D denote its pdf. In this case, p_D is a *joint* pdf of D_u and D_v. The state space of the MDP is $\mathcal{B} = [0, C]^2$, and the action space at state $b = (b_u, b_v) \in \mathcal{B}$ is given by $\mathcal{S}(b) \triangleq \{(r_u, r_v) : c_{\text{tx}} r_u + c_{\text{rx}} r_v = s_u \leq b_u,\ c_{\text{tx}} r_v + c_{\text{rx}} r_u = s_v \leq b_v\}$. The goal is to find an optimal policy that maximizes the average utility $\lim_{K \to \infty} \mathbb{E}_\pi \left(\sum_{i=0}^{K-1} U(r_u(i)) + U(r_v(i)) \right) / K$, which is done using methods similar to those of Section IV-A. Also, corresponding discretization bounds can be obtained.

Similarly to the predictable energy model, the *DRC* algorithms can be used with this model. In this case, the *DRC* policies are calculated using the marginal pdfs of D_u and D_v (rather than the joint pdf), and thus do not account for the dependency between D_u and D_v.

V. Non-Stationary Energy Models

In Section IV, we assumed that the harvested energy is a stationary (i.i.d.) process. However, in many environments, the harvested energy characteristics change with time, making *non-stationary models* a better fit. For instance, if slots represent days, the distribution of the harvested energy in a day is different in different seasons of the year. Moreover, sometimes the harvesting conditions around the device change arbitrarily with time (e.g., due to changes in the location of the device). In such cases, the appropriate model is an MDP with *non-homogeneous* transition function. Since the changes in the distribution cannot be known in advance, we use an *online learning algorithm* and measure its performance *in hindsight* against stationary policies. In particular, we adopt the algorithm and the results of [20] to our problem.

$$J_i(b) = \max_{s\in\mathcal{S}} \left\{ \hat{U}_i(s) + \inf_{\delta\in\Delta} \sum_{b'\in\mathcal{B}} J_i(b')p^{\delta}(b'|b,s) - J_i(b_0) \right\},\ b \in \mathcal{B}, \tag{14}$$

$$s_i(b_i) \in \operatorname*{argmax}_{s\in\mathcal{S}} \left(\hat{U}_i(s) + \mathcal{N}_i(s) + \inf_{\delta\in\Delta} \sum_{b'\in\mathcal{B}} J_i(b')p^{\delta}(b'|b_i,s) \right). \tag{15}$$

A. Setting

For simplicity, we assume that the MDP was already discretized, as described in Section IV. We omit the subscript h throughout, since it is fixed. For any $b \in \mathcal{B}$ and $s \in \mathcal{S}$, the transition density in slot i is denoted by $p_i(\cdot|b,s)$. It determines the next energy storage level $B(i+1)$ given that the current energy storage level is $B(i) = b$ and the spending rate is $s(i) = s$. As before, this transition density is determined by (1), where we assume the *linear storage* model. However, in contrast to the case studied in Section IV, the transition probabilities change with time, since the distribution of D changes with time. In addition, in this section we assume a more general case, where the utility function may also change with time, that is $U_i(s_i)$.

Let $\mathcal{L}$ denote a finite set of indices. Let each $p^{\ell} : \mathcal{B} \times \mathcal{B} \times \mathcal{S} \to [0,1]$, for $\ell \in \mathcal{L}$, denote a fixed transition function. The set $\mathcal{L}$ can be interpreted as the set of *slot types*. If the slot is of type $\ell \in \mathcal{L}$, the energy storage level in the next slot is determined according to p^{ℓ}. We assume that both the types set $\mathcal{L}$ and transition probabilities $\left\{p^{\ell}\right\}_{\ell\in\mathcal{L}}$ are known. The *actual* type of slot i, however, is *unknown in advance*. For example, consider a room with several possible device positions. Every position has a different lighting condition. The set of the positions corresponds to $\mathcal{L}$ and is known in advance (together with the harvested energy characteristics at these positions). However, the actual position of the device at each time slot is unknown.

Moreover, we assume that the type of slot i can be some *convex combination* of the basic types, which we denote by $\delta_i \in \Delta \subseteq \Delta(\mathcal{L})$. Thus, the actual transition probability at slot i is $p_i(b'|b,s) = \sum_{\ell\in\mathcal{L}} \delta_i(\ell)p^{\ell}(b'|b,s)$. We call any $\delta \in \Delta(\mathcal{L})$ a *mixed* type, and write $p^{\delta}(b'|b,s) \triangleq \sum_{\ell\in\mathcal{L}} \delta(\ell)p^{\ell}(b'|b,s)$. As proposed in [20], we measure the performance of an online algorithm by comparing its average utility to that of an *optimal stationary policy in hindsight* $J_K^* \triangleq \max_{\pi:\mathcal{B}\to\mathcal{S}} \sum_{i=0}^{K-1} \mathbb{E}_{\pi}\left(U_i(\pi(b_i^*))\right)/K$, where the initial state b_0^* is fixed, each next state b_{i+1}^* is distributed according to $p_i(\cdot|b_i^*, \pi(b_i^*))$, and the expectation is taken with respect to the sequence $b_0^*, ..., b_{K-1}^*$. The *regret* of the algorithm after K time slots is then $R_K \triangleq J_K^* - \sum_{i=0}^{K-1} \mathbb{E}\left(U_i(s_i)\right)/K$, where the initial state b_0 is fixed, the spending rate s_i is determined according the algorithm's rule, the next state is distributed according to $p_i(\cdot|b_i,s_i)$, and the expectation is taken with respect to the sequence $b_0, ..., b_{K-1}$. The goal is to use an online algorithm which asymptotically minimizes the regret with respect to every possible sequence of the harvested energy level $D(0), D(1), ..., D(K-1)$.

B. Online Algorithm

In order to use the algorithm of [20], we need some ergodicity or mixing assumption to be satisfied, similarly to the requirements in Section IV-A.

Assumption 5.1 (Bounded Mixing and Cover Times): The sequence of transition functions $p_0, p_1, ..., p_{K-1}$ is non-periodic. There exists constants τ and τ_{cov} such that for every (randomized) policy $\pi : \mathcal{B} \to \Delta(\mathcal{S})$ and every mixed type δ, the Markov chain induced by the transition function $p^{\delta}(\cdot|s,\pi(s))$ is ergodic with expected mixing time at most τ and expected cover time[3] at most τ_{cov}.

In addition, in order to obtain small regret, we need a limit on the extent to which the transition functions may vary with time.

Assumption 5.2 (ϵ-arbitrary Transition Functions): Let P^{δ} denote the transition matrix of a given mixed type $\delta \in \Delta$ for a given policy π. Let $F^{\delta} \triangleq \left[I - P^{\delta} + P^{\delta}_{\infty}\right]^{-1}$, where $P^{\delta}_{\infty} \triangleq \lim_{K\to\infty} \frac{1}{K}\sum_{i=0}^{K-1}\left(P^{\delta}\right)^i$. F^{δ} is called the *fundamental matrix* associated with P^{δ}. We assume that there exists a finite constant F such that[4] $\left\|F^{\delta}\right\|_{\infty} \le F$ for all $\delta \in \Delta$. Moreover, we assume that there exists $\epsilon > 0$ such that $\|P^{\delta} - P^{\delta'}\|_{\infty} < \epsilon$ for every $\delta, \delta' \in \Delta$.
Roughly speaking, this assumption implies that we have a bound on how much the transition law can change when the type of the time slot changes. In [7], we present an example for the case where these assumptions hold in our setting.

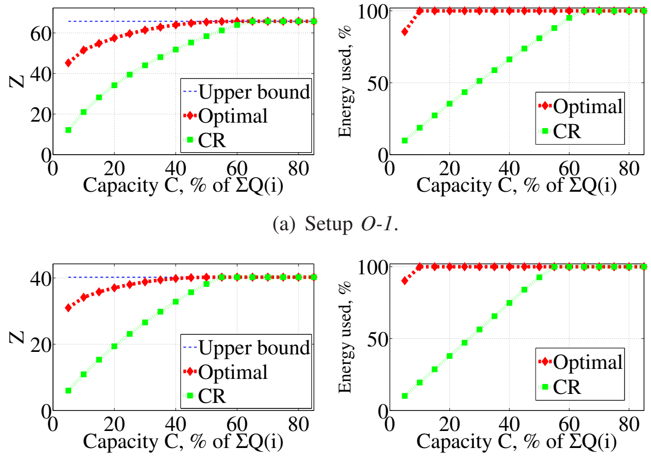
Let $\hat{U}_{K-1}(s) \triangleq \sum_{i=0}^{K-1} U_i(s)/K$ denote the *empirical average utility* until time slot $K-1$. Below we present the Online Robust Dynamic Programming (ORDP) Algorithm and the corresponding low-regret result.

Algorithm 1 ORDP Algorithm:

```
At time slot 0, use an arbitrary spending rate $s_0$.
for $i = 1, 2, ...$ do
  1. Solve Bellman equations (14) for MDPs with infinite-horizon average-reward objective (via linear programming or otherwise; e.g., see [18]), where $b_0 \in \mathcal{B}$ is a fixed state and $J_i(b_0)$ is a normalization term.
  2. Sample a random variable $\mathcal{N}_i$ uniformly over the support $[-i^{-0.5}, i^{0.5}]^{|\mathcal{S}|}$.
  3. Output the action according to (15).
```

[3] See [20] for the standard definitions of the mixing and cover times of a Markov chain.

[4] We let $\|M\|_{\infty}$ denote the maximum absolute row-sum of a matrix M.



(a) Setup *O-1*.

(b) Setup *L-3*.

Fig. 2. Z (left) and % of energy used (right), for a *single node* with a *predictable profile energy*, under the optimal solution and the *CR* policy. The x-axis expresses the storage capacity as a *percentage* of the total incoming energy.

Theorem 2: Under Assumptions 5.1 and 5.2, for all $K > 4e\tau/\epsilon$ we have that

$$R_K \leq (Z+1)\epsilon + \sqrt{|\mathcal{B}|\,|\mathcal{S}|\,/K} + 4\tau^2\sqrt{\log(|\mathcal{S}|)/K} + 4\tau/K,$$

for any sequence of the harvested energy $D(0), ..., D(K-1)$.

We note that the ORDP algorithm solves a linear program at each time slot, which is computationally expensive. An alternative is to periodically compute a new policy and follow this policy for a while. Choosing the length of the intervening intervals provides a mean of trade-off between the regret bound and the computational complexity of the solution.

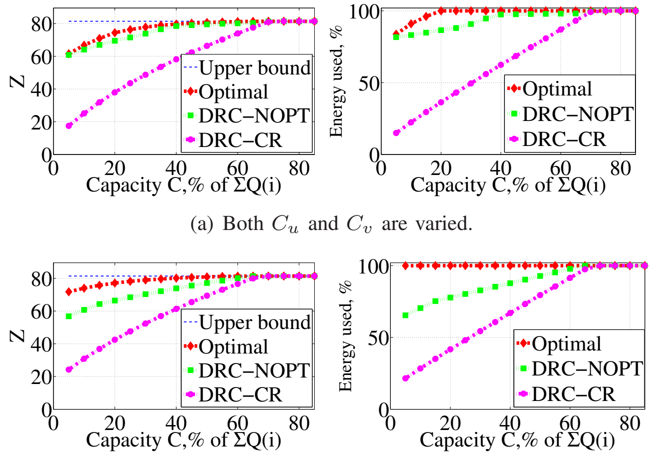
VI. Numerical and Experimental Results

A. Trace-based Simulation

To evaluate the performance of the various policies, we performed an extensive simulation study using traces from outdoor locations [1] and from our measurement campaign, in which we recorded indoor light energy traces at a set of locations at Columbia University for more than a year [9]. The traces are available online at enhants.ee.columbia.edu. We use the notation *L-1, L-2*, ... for the locations of the light energy traces, corresponding to the measurement locations in [9]. For simplicity, we use $c_{\text{tx}} = c_{\text{rx}}$, and set them to 0.5 nJ/bit [10]. As a utility function, we use $U(\cdot) = \log(1 + (\cdot))$. Further technical details about the traces are provided in [7].

For a *single node* with a *predictable profile energy model*, Fig. 2 illustrates the optimal solution and the performance of the *CR* policy, for energy profiles of two different setups, and shows the upper bound derived in Observation 1. It can be seen that this bound is tight for large C. In our numerical results, the actual ratio between the *CR* solution and the optimal solution is substantially lower than the approximation ratio given in Observation 3.

For a *link* with a *predictable profile energy model*, we use light energy traces *concurrently recorded in nearby locations*.



(a) Both C_u and C_v are varied.

(b) C_u is kept the same, and C_v is varied.

Fig. 3. Z (left) and energy used (right), for *a link* (u, v) = *(L-1,L-2)* with a *predictable profile energy*. The results include the optimal solution, and the *DRC-NOPT* and *DRC-CR* policies.

Fig. 3 illustrates the optimal solution and the performance of the *DRC-CR* policy for a link (u, v) = *(L-1,L-2)*. Fig. 3(a) shows the case in which both $C_u(i)$ and $C_v(i)$ are varied, while Fig. 3(b) shows the case in which $C_v(i)$ is varied and $C_u(i)$ is kept constant. We note that the *DRC-NOPT* obtains results that are close to the optimal solution in the first case but not in the second case. Separately, we studied the *DRC-SG* policy and have noticed that, for the traces examined, T^L is mostly relatively close to the lower bound derived in Observation 4. For example, for a link *(L-1,L-2)*, $\max(T_u, T_v) = 0.52$, and $T^L_{u,v} = 0.57$, and for a link *(L-2,L-3)*, $\max(T_u, T_v) = 0.52$, and $T^L = 0.64$.

For a *single node* and the *stationary stochastic model*, Fig. 4 shows the optimal solution and the solutions obtained by the *SL* and *THR1* (*THR* with one threshold) policies. The policies were evaluated using an empirical pdf of the diurnal energy of setup *L-1*. The calculations of the optimal solutions rely on discretization procedure described in Section IV-A. It can be seen that for this setup, the performance of the *SL* policy is very close to optimal.

Finally, we note that preliminary simulation results based on our traces show that the *non-stationary learning framework* of Section V usually provide better performance than the schemes that were designed for the stationary model. The extensive performance evaluation for this case remains subject for future work.

B. Testbed Experimental Results

To evaluate the performance of the policies in realistic environments, we also used the testbed of energy harvesting devices that we have recently developed [8]. In this testbed, the devices harvest the energy from *indoor light*, and adjust their communication parameters accordingly. We implemented the *CR* and *SL* single-node policies. We also implemented the *DRC* algorithms that can be used with any single-node policy. Testbed implementation allows us to examine the behavior of

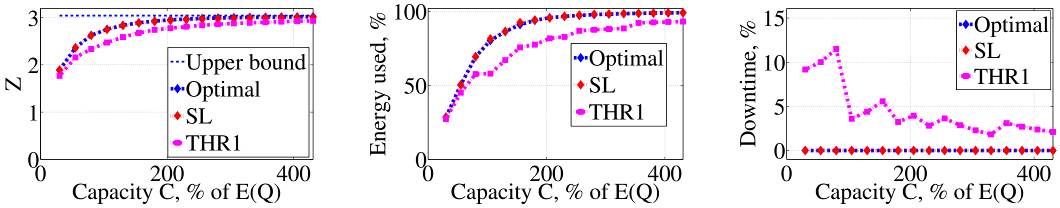


Fig. 4. Z, % of energy spent, and the % of downtime under the optimal solution and the *SL* and *THR1* (ON-OFF) policies, for setup *L-1*. The x-axis expresses the storage capacity as a *percentage* of the *expected* energy.

various policies with *widely varying* and *controlled* energy sources.

For example, under the *DRC-SG* policy, we examined the effect on the performance of the dependence on $Q_u(i)$ and $Q_v(i)$. With strongly correlated $Q_u(i), Q_v(i)$ (i.e., harvesting the energy of the same source), similarly to the light energy traces examined above, $T^L_{u,v}$ is close to the lower bound derived in Observation 4. However, when $Q_u(i)$ and $Q_v(i)$ were independent (i.e., u and v positioned next to two lamps controlled by different people), $T^L_{u,v}$ was closer to the *upper bound* derived in Observation 4. For example, for $T_u = 0.65$ and $T_v = 0.55$, the link downtime $T^L_{u,v}$ was 0.98. Namely, while both u and v had substantial amounts of energy, the data rate on (u, v) was extremely low.

With the *SL* policy implementation we have observed, for example, that low α_{SL} values lead to smooth spending rates, but cause substantial energy storage level variations, while high α_{SL} values lead to highly non-uniform energy spending rates.

VII. Conclusions

In this work we analyzed and evaluated numerically and experimentally a number of simple energy allocation policies for the predictable profile model and the stochastic model. Our analysis applies to linear and non-linear storage models. Due to the problems' complexity, the analysis presented in this paper applies to a node and to a node pair (link). Most algorithms that were developed for a network are too complex for resource-constrained nodes. Therefore, we plan to develop simple algorithms for a network. However, the curse of dimensionality makes it challenging to directly extend the examined stochastic models to larger scenarios, and therefore, approximate solution techniques should be applied (such as Approximate Linear Programming as in [3], [4]).

VIII. Acknowledgements

This work was supported in part by the Vodafone Americas Foundation Wireless Innovation Project, NSF grants CNS-0916263, CCF-0964497, and CNS-10-54856, and DHS Task Order #HSHQDC-10-J-00204. We thank Zainab Noorbhaiwala, Gerald Stanje, John Sarik, and Hao Wang for their assistance with the experimental work.

References

[1] "Measurement and Instrumentation Data Center, National Renewable Energy Laboratory (NREL), US DOE," www.nrel.gov/midc/.

[2] C.-S. Chow, "Multigrid algorithms and complexity results for discrete-time stochastic control and related fixed-point problems," Ph.D. dissertation, MIT, Cambridge, MA, 1990.

[3] D. De Farias and B. Van Roy, "Approximate linear programming for average-cost dynamic programming," in *Advances in NIPS'03*, Dec. 2003.

[4] ——, "A cost-shaping linear program for average-cost approximate dynamic programming with performance guarantees," *Math. Oper. Res.*, vol. 31, no. 3, pp. 597–620, 2006.

[5] K.-W. Fan, Z. Zheng, and P. Sinha, "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks," in *Proc. ACM SenSys'08*, Nov. 2008.

[6] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Control of wireless networks with rechargeable batteries," *IEEE Trans. Wireless. Comm.*, vol. 9, no. 2, pp. 581–593, 2010.

[7] M. Gorlatova, A. Bernstein, and G. Zussman, "Performance evaluation of resource allocation policies for energy harvesting devices," Columbia University, Tech. Rep. 2011-01-05, Apr. 2011, available at: http://enhants.ee.columbia.edu/images/papers/CU_EE_2011-01-05.pdf.

[8] M. Gorlatova, T. Sharma, D. Shrestha, E. Xu, J. Chen, A. Skolnik, D. Piao, P. Kinget, I. Kymissis, D. Rubenstein, and G. Zussman, "Prototyping Energy Harvesting Active Networked Tags (EnHANTs) with MICA2 motes," in *Proc. IEEE SECON'10*, June 2010.

[9] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," in *Proc. IEEE INFOCOM'11*, Apr. 2011.

[10] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman, "Challenge: ultra-low-power energy-harvesting active networked tags (EnHANTs)," in *Proc. ACM MobiCom'09*, Sept. 2009.

[11] J. Gummeson, S. S. Clark, K. Fu, and D. Ganesan, "On the limits of effective micro-energy harvesting on mobile CRFID sensors," in *Proc. ACM MobiSys'10*, June 2010.

[12] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 4, 2007.

[13] K. Kar, A. Krishnamurthy, and N. Jaggi, "Dynamic node activation in networks of rechargeable sensors," *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 15–26, 2006.

[14] L. Lin, N. Shroff, and R. Srikant, "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1021–1034, 2007.

[15] R.-S. Liu, P. Sinha, and C. E. Koksal, "Joint energy management and resource allocation in rechargeable sensor networks," in *Proc. IEEE INFOCOM'10*, Mar. 2010.

[16] D. Niyato, E. Hossain, and A. Fallahi, "Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: performance analysis and optimization," *IEEE Trans. Mobile Comput.*, vol. 6, no. 2, pp. 221–236, Feb. 2007.

[17] D. Noh and T. Abdelzaher, "Efficient flow-control algorithm cooperating with energy allocation scheme for solar-powered WSNs," *Wireless Comm. and Mobile Comput.*, 2010, published online at: http://onlinelibrary.wiley.com/doi/10.1002/wcm.965/pdf.

[18] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[19] C. Vigorito, D. Ganesan, and A. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," in *Proc. IEEE SECON'07*, June 2007.

[20] J. Y. Yu and S. Mannor, "Online learning in Markov decision processes with arbitrarily changing rewards and transitions," in *Proc. IEEE GAMENETS'09*, May 2009.