Panda: Neighbor Discovery on a Power Harvesting Budget

Robert Margolies*, Guy Grebla*, Tingjun Chen*, Dan Rubenstein†, Gil Zussman*

*Electrical Engineering and †Computer Science, Columbia University, New York, NY 10027

{robm, guy, tingjun, gil}@ee.columbia.edu, danr@cs.columbia.edu

Abstract—**Object tracking applications are gaining popularity and will soon utilize Energy Harvesting (EH) low-power nodes that will consume power mostly for Neighbor Discovery (ND) (i.e., *identifying* nodes within communication range). Although ND protocols were developed for sensor networks, the *challenges posed by emerging EH low-power transceivers were not addressed*. Therefore, we *design an ND protocol tailored for the characteristics of a representative EH prototype*: the TI eZ430-RF2500-SEH. We present a generalized model of ND accounting for unique prototype characteristics (i.e., energy costs for transmission/reception, and transceiver state switching times/costs). Then, we present the Power Aware Neighbor Discovery Asynchronously (Panda) protocol in which nodes transition between the sleep, receive, and transmit states. We analyze Panda and select its parameters to maximize the *ND rate* subject to a homogeneous power budget. We also present Panda-D, designed for *non-homogeneous* EH nodes. We perform extensive testbed evaluations using the prototypes and study various design tradeoffs. We demonstrate a small difference (less then 2%) between experimental and analytical results, thereby confirming the modeling assumptions. Moreover, we show that Panda improves the ND rate by up to 3x compared to related protocols. Finally, we show that Panda-D operates well under non-homogeneous power harvesting.**

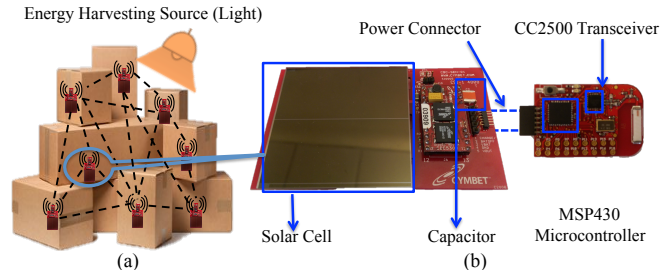*Index Terms*—**Neighbor discovery, energy harvesting, wireless**

Fig. 1. (a) EH nodes can be attached to everyday objects (e.g., boxes) and utilize neighbor discovery protocols for inventory tracking and monitoring applications. (b) The prototype EH node stores harvested energy in a capacitor and powers the microcontroller and transceiver.

I. INTRODUCTION

Object tracking and monitoring applications are gaining popularity within the realm of the Internet-of-Things [1]. Emerging low-power wireless nodes that can be attached to physical objects are enablers for such applications. Often, these nodes are meant to interact with a reader, but architectures are emerging that handle scenarios where no reader may be present, or where the number of nodes overwhelms the readers' availability. These scenarios can be supported by Energy Harvesting (EH) tags (e.g., [2], [3] and references therein) that are able to communicate peer-to-peer and are powered by an ambient energy source (e.g., light).

Such EH nodes will enable tracking applications in healthcare, smart buildings, assisted living, manufacturing, supply chain management, and intelligent transportation as discussed in [4], [5]. An example application, illustrated in Fig. 1(a), is a large warehouse that contains many inventory items, each of which is equipped with an EH node. Each node has an ID that corresponds to the physical object (item). The nodes utilize a Neighbor Discovery (ND) protocol to *identify* neighbors which are within communication range, and therefore, the system can collect information about the objects' whereabouts. A simple application is identifying misplaced objects: often when an item is misplaced (e.g., in a furniture warehouse, a box of

table parts is moved to an area with boxes of bed parts), its ID is significantly different from the IDs of its neighbors. In such a case, the misplaced node can, for instance, flash a low-power LED to indicate that it is lost.

In this paper, *we develop an ND protocol for Commercial Off-The-Shelf (COTS) EH nodes, based on the TI eZ430-RF2500-SEH* [6] (shown in Fig. 1(b)). The nodes harvest ambient light to supply energy to a low-power microcontroller and transceiver. To maintain *perpetual* tracking of the (potentially) mobile objects, ND must be run continuously with the node operating in an *ultra-low-power mode* that consumes power at the rate of power harvested [7]. Our objective is **to maximize the rate in which nodes *discover* their neighbors, given a constrained power budget at each EH node.**

ND has always been an important part of many network scenarios [8], [9]. Yet, to consume power at the rate of power harvested, EH nodes require *extremely limited power budgets*: we show that, even with optimized power spending, the duty cycles are between 0.1-0.6%. Therefore, numerous assumptions from related works (e.g., [10], [11]) no longer hold, including that switching times (between the sleep, receive, and transmit states) draw negligible power and that the power costs to send and receive are identical (see Section II for details). Furthermore, in the envisioned applications, the node's main task is to perform ND, and thus, the power consumed by ND is the dominant component of the power budget.

Hence, we design, analyze, and experiment with **Panda–Power Aware Neighbor Discovery Asynchronously**,[1] an ND protocol that *maximizes the average discovery rate under a given power budget*. The main contributions of this paper are:

---

[1]The protocol name, Panda, relates to the animal as both EH nodes and Pandas spend the majority of their time sleeping to conserve energy.

(C1) Radio Characterization: We model a generic ultra-low-power EH node that captures the capabilities of our prototype (Fig. 1(b)). We also study, for the first time, important properties of the radio in the context of ND.

(C2) Panda Protocol: We develop the Panda protocol in which an EH node discovers its neighbors by transitioning between the sleep, receive, and transmit states at rates that satisfy a power budget. Furthermore, we present Panda-Dynamic (Panda-D), which extends Panda's applicability to non-homogeneous power harvesting and multihop topologies.

(C3) Protocol Optimization: Using techniques from renewal theory, we derive closed form expressions for the discovery rate and the power consumption. We develop the Panda Configuration Algorithm (PCA) to determine the node's duration in each state (sleep, receive, transmit), such that the discovery rate is maximized, while meeting the power budget. The solution obtained by the PCA is numerically shown to be between 94–99.9% of the optimal for all scenarios considered.

(C4) Experimental Evaluation: Using TI eZ430-RF2500-SEH EH nodes [6], we show that the real-life discovery rates are within 2% of the analytically predicted values, demonstrating the practicality of our model. Moreover, we show that Panda's *experimental* discovery rate is up to 3 times higher than the discovery rates from simulations of two of the previously best known low-power ND protocols [10], [12]. Furthermore, we demonstrate that Panda-D adjusts the rate of ND for scenarios with non-homogenous power harvesting and multihop topologies.

The rest of the paper is organized as follows. In Section II we discuss related work. In Section III we present the system model. In Sections IV and V, we present and optimize Panda, respectively. In Section VI, we present the Panda-D protocol. In Section VII, we evaluate Panda experimentally. We conclude in Section VIII. Due to space constraints, several results are omitted and can be found in a technical report [13].

II. RELATED WORK

ND for low-power wireless networks is a well studied problem (see [1], [8], [9] for a summary). The protocols can be categorized into deterministic (e.g., [10], [11], [14]–[16]) and probabilistic (e.g., [12], [17]). Deterministic protocols focus on guaranteeing an upper bound on *discovery latency*, while the choice of parameters (e.g., prime numbers) is often limited. On the other hand, the most well-known probabilistic protocol [12], has a better average ND rate, but suffers from an unbounded discovery latency. Our probabilistic protocol, Panda, is fundamentally different: other protocols (i) are constrained by a duty cycle, instead of a power budget, (ii) do not account for channel collisions (e.g., when two nodes transmit at the same time), (iii) rely on each node maintaining synchronized time slots,[2] or (iv) do not consider practical hardware energy consumption costs (i.e., the power consumed by the radio to transition between different states).

[2]It was shown in [14] how the aligned time slot assumption can be relaxed. Yet, practical considerations such as selecting the slot duration and avoiding collisions are not described.

To the best of our knowledge, *Panda is the first ND protocol for EH nodes* and the *first attempt to maximize the discovery rate, given a power budget*. As such, Panda will operate with duty cycles between 0.1-0.6%, which is an order of magnitude lower than those typically considered in prior works [9].

In our experiments, we use hardware from [6]. There are also numerous other hardware options for EH nodes [3], [18], computational RFIDs [19], and mm^3-scale wireless devices [20]. Additionally, there are other radio features that achieve low energy consumption. For example, preamble-sampling and wake up radios were investigated in [21] and [22], respectively, for WSNs. However, the added power consumption of these features makes them impractical for the EH nodes we study. Furthermore, numerous options for low-power wireless communication exist (e.g., Bluetooth Low Energy [23]). However, [6] is one of the increasingly popular low-power EH nodes which seamlessly support wireless protocol development.

III. SYSTEM MODEL

In this section, we describe our prototypes, based on which, we introduce the notation and the system model.

A. Prototype Description

The prototype is shown in Fig. 1(b) and is based on the commercially available TI eZ430-RF2500-SEH [6]. We made some modifications to the hardware, which are summarized in [13]. We now describe the prototype's components:

Energy Harvesting Power Source: The prototype harvests light from a Sanyo AM 1815 amorphous solar cell. The solar cell is set to a fixed harvesting voltage of 1.02V (no power point tracking techniques are used). To measure the power harvested, we place a ammeter in series with the solar cell.

Energy Storage: The energy harvested by the solar cells is stored in a capacitor and the voltage is denoted by V_{cap}. The voltage is regulated to 3.5V to power the node. We modified the board design to enable experimentation with varying capacitor sizes. Unless stated otherwise, we use a 30mF capacitor. To ensure stable voltage regulation, a software cutoff is imposed; if $V_{cap} \leq 3.6$V, the node enters and remains in a low-power sleep state until enough power is harvested such that V_{cap} exceeds the cutoff.

Low-Power Microcontroller: A TI-MSP430 microcontroller is used to provide computational capabilities. These include (i) sampling the capacitor voltage using an analog to digital converter (ADC), (ii) operating a low-power 12kHz clock with an idle power draw of 1.6μW to instruct the node to enter and exit an ultra-low-power sleep state, and (iii) receiving and sending messages to the radio layer.

Low-Power Transceiver: The prototype utilizes a CC2500 wireless transceiver (a 2.4GHz transceiver designed to provide low-power wireless communication) to send and receive messages. The transceiver operates at 250kbps and consumes 64.85mW while in receive state. The transmission power can be set in software and we utilize levels between -16 and 1dBm, with a resulting power consumption between 53.25 and 86.82mW. At these levels, nodes within the same room typically have little or no packet loss.

B. System Model

The model is based on the prototype, yet it is generalizable to a class of other prototypes (e.g., [3]). A node can be in one out of three states, denoted by the set $\mathcal{S} = \{s, r, t\}$ for sleep (s), receive[3] (r), and transmit (t). A node in state $i \in \mathcal{S}$ consumes power of P_i. Since the power consumption in sleep state is negligible, we assume $P_s = 0$ throughout the paper and remark that all results can be easily applied for $P_s > 0$, as described in [13]. For the power budgets we consider, the energy consumed by the radio to transition between different states is non-negligible. Hence, we denote by C_{ij} the energy (µJ) consumed to switch from state i to state j ($i, j \in \mathcal{S}$).

Unfortunately, the prototype does not have explicit power awareness (unlike, e.g., [3]). Therefore, we impose a power budget, P_b (mW) on each node. The power budget is set such that *energy neutrality* is achieved: nodes consume power (on average) at the power harvesting rate [24]. Hence, for an EH node harvesting more power (e.g., brighter light source), the power budget P_b is higher.

We denote by N the number of nodes in the network and present two important definitions:

Definition 1: The **discovery message** is a broadcast packet containing the ID of the transmitter.[4] A **discovery** occurs when a node receives a discovery message from a neighbor. Multiple discoveries can occur per discovery message transmission.

Definition 2: The **discovery rate**, denoted by U, is the expected number of discoveries in the network per second. The objective of the ND protocol is to *maximize the discovery rate, subject to a given power budget.* This is in contrast to other works which seek to minimize the worst case discovery latency [10], [11], subject to a duty cycle. As such, in Section VII, we also consider the discovery latency, or time in between discoveries, as a secondary performance metric.

IV. THE PANDA PROTOCOL

In this section we describe and analyze Panda, an asynchronous ND protocol, which operates under a power budget.

A. Protocol Description

To ensure perpetual operation under the power budget P_b (mW), nodes initialize in a low-power sleep state to conserve energy. To maximize the discovery rate, Panda follows a probabilistic approach in which nodes sleep for an exponential duration with rate λ (ms^{-1}). The probabilistic sleep duration prevents unwanted synchronization among subsets of nodes.

Following sleep, nodes awaken and listen to the channel for discovery messages from their neighbors for a fixed duration of l (ms). If a message is received, the node remains in the listen state until it completes reception of this message. If

---

[3]We refer to the *receive* and the *listen* states synonymously as the power consumption of the prototype in both states is similar.

[4]In practice, the discovery message may include information on already discovered neighbors, thus enabling *indirect* discoveries. However, we do not consider these indirect discoveries.

no transmission is heard while in the listen state, the node transmits its discovery message of fixed duration M(ms).[5]

Note that in Panda, similar to CSMA, nodes always listen before they transmit, and therefore, there are no packet collisions between two nodes in wireless communication range of one another. Additionally, after a message is transmitted, the node returns to the sleep state. Hence, there is no acknowledgement of the discovery. This is because coordinating acknowledgement messages among multiple potential receivers can be costly, requiring additional listening by the transmitter and possibly collision resolution.

B. Analysis

While Panda can operate in general scenarios, for analytical tractability, we assume the following:
(A1) All nodes are homogeneous, namely, have the same power budget P_b and the same hardware.
(A2) Every pair of nodes can exchange packets (clique topology) with no packet errors due to noise.
(A3) The number of nodes, N, is known a priori.

These assumptions are applicable to some systems and envisioned applications. For example, when tracking boxes in a room (Fig.1(a)), these assumptions are close to reality as nodes in close proximity harvest similar amounts of energy, have few packets lost, and the number of nodes can be estimated a priori. However, for scenarios in which these assumptions do not hold, in Section VI, we present Panda-Dynamic which is based on relaxed assumptions and discuss the implications.

Using these assumptions, we now use techniques from renewal theory [25] to analyze Panda for a network of N nodes. The renewal process is shown pictorially in Fig. 2. The renewal initiates with all nodes in the sleep state and ends after one node completes its transmission, whether the message is heard or not. The sleep duration for each node follows a memoryless exponential distribution. Therefore, for all analytical purposes, all nodes effectively initiate their sleep state at the start of the renewal.

In each renewal, the *first* node to wake up begins its listen state, and after a duration l, it transmits its discovery message. This is exemplified by node 6 in Fig. 2; we denote by $\mathcal{N}_t$ the set containing a single transmitting node in a renewal.

Nodes that are in the receive state (r) when a message transmission begins, will stay in this state until the transmission is completed and then switch to the sleep state (s). We denote by $\mathcal{N}_r$ the set of such nodes and $|\mathcal{N}_r|$ the size of the set, exemplified by nodes 2-4 in Fig. 2. The expected idle listening time of a node in $\mathcal{N}_r$ is denoted by χ. Fig. 2 shows examples of idle listening durations for nodes 2–4, denoted as χ_i. Any node which wakes up in the middle of the message transmission immediately senses the busy channel and returns to the sleep state. An example is node 5 in Fig. 2.

When the transmission is completed, all nodes are in sleep state and the renewal restarts. The average renewal duration is

---

[5]The discovery message duration, M, is fixed, stemming from the fixed amount of data contained in the message.
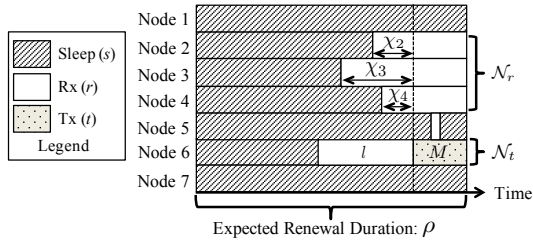
Fig. 2. Renewal process representing one renewal cycle for $N = 7$ nodes: all nodes begin in a memoryless sleep state and the renewal restarts after the first node to wake up completes its transmission.

the time it takes for the first node to wake up (occuring with rate $N\lambda$), listen for a duration l, and transmit a message for a duration of M. Hence, the expected renewal duration ρ is:

$$\rho = 1/(\lambda N) + l + M. \qquad (1)$$

C. Discovery Rate

Recall that the objective of Panda is to maximize the discovery rate, U (see Def. 2). Considering U as the reward function and applying the elementary renewal theorem for renewal-reward processes [25], we obtain:

$$U := \lim_{t \to \infty} \frac{u(t)}{t} = \frac{\mathbb{E}[|\mathcal{N}_r|]}{\rho}, \qquad (2)$$

where $u(t)$ represents the number of discoveries (as defined by Def. 1) by time t and ρ is computed by (1).

There are $N - 1$ nodes who are not the transmitter in the renewal, each of which is equally and independently likely to discover the transmitter. A discovery occurs if the node wakes up from sleep within a period of time l after the transmitting node ($\mathcal{N}_t$) wakes up, an event with probability $1 - e^{-\lambda l}$. Hence,

$$\mathbb{E}[|\mathcal{N}_r|] = (N - 1)(1 - e^{-\lambda l}). \qquad (3)$$

D. Energy Consumption

Since all nodes are homogenous (A1), we let n denote an arbitrary node and define a random variable Y that indicates the set $(\mathcal{N}_t, \mathcal{N}_r)$ in which the node resides in the renewal:

$$Y = \begin{cases} 0, & n \in \mathcal{N}_t \\ 1, & n \in \mathcal{N}_r \\ 2, & \text{otherwise.} \end{cases} \qquad (4)$$

We let $\eta(y)$ represent the expected amount of energy (µJ) consumed by a node in a renewal in which $Y = y$. Thus,

$$\eta(0) = C_{sr} + P_r l + P_t M + C_{ts}, \qquad (5)$$

$$\eta(1) = C_{sr} + P_r(\chi + M) + C_{rs}. \qquad (6)$$

Eq. (5) defines the energy consumption of the transmitting node, which consumes energy to wake up from sleep (C_{sr}), listen for a period of l, transmit a message of length M, and then return to sleep (C_{ts}). For a receiving node, the expected energy consumption is defined in (6) and consists of idle listening before the message transmission (with χ denoting the expected duration of idle listening, shown in Fig. 2). The derivation of χ can be found in [13]. Then, the node listens for the duration of the message M. Throughout this paper, we assume that nodes which sleep for the entire renewal (e.g., nodes 1 and 7 in Fig. 2), and those which wake up briefly and

sense a busy channel (e.g., node 5), do not consume power, and thus $\eta(2) = 0$. In [13], we show how it can be relaxed.

The computation of $\Pr(Y = y)$ for $y = 0, 1$ is as follows. By definition of the renewal, there will be exactly one transmitter in a renewal and due to assumption (A1), $\Pr(Y = 0) = 1/N$. Each of the remaining $N - 1$ nodes successfully receive the message, if they start listening in a period of length l preceding the transmission. Hence, since the sleep duration is exponentially distributed, $\Pr(Y = 1) = (1 - e^{-\lambda l})(N - 1)/N$.

Define $\Phi(y) = \Pr(Y = y)\eta(y)/\rho$ and note its units are (mW); we will often refer to $\Phi(0)$ as the *probing power* while $\Phi(1)$ is referred to as the *discovery power*. As described above, $\eta(2) = 0$, and thus $\Phi(2) = 0$. The expected power consumed in a renewal must meet the power budget, $\Phi(0) + \Phi(1) \leq P_b$.

V. OPTIMIZATION OF PANDA

Clearly, the choice of the sleep rate (λ) and the listen duration (l) determines the power consumption of the node as well as the discovery rate U. First, we demonstrate that an analytical solution is difficult to obtain. Next, we describe the Panda Configuration Algorithm (PCA) which obtains the *configuration parameters* (λ, l) for Panda. Finally, we demonstrate that the PCA obtains a nearly-optimal discovery rate.

A. Problem Formulation and Preliminaries

Finding (λ^*, l^*) that maximizes U is formulated as follows:

$$\max_{\lambda, l} \quad U = \frac{(N - 1)(1 - e^{-\lambda l})}{\rho} \qquad (7)$$

$$\text{s.t.} \quad \Phi(0) + \Phi(1) \leq P_b, \qquad (8)$$

where (7) is derived using (2) and (3). Recall that ρ is computed from (1) and $\Phi(y)$ is computed using the results from Section IV-D. The problem as formulated above is non-convex and non-linear, and is thereby challenging to solve.

In the following subsections, we will attempt to find nearly-optimal Panda configuration parameters (λ, l). We now provide several observations on the specific structure of the problem which are used throughout this section. First, we find the following Taylor-series approximation useful:

$$e^{-x} \geq 1 - x \text{ for } x \geq 0, \text{ and } e^{-x} \approx 1 - x \text{ for } x \approx 0. \qquad (9)$$

We substitute x with λl in (9),[6]

$$U \leq \frac{(N - 1)\lambda l}{\rho} := \overline{U}. \qquad (10)$$

B. Panda Configuration Algorithm (PCA)

The Panda Configuration Algorithm (PCA) returns a configuration of λ and l that satisfy (8). To find a configuration with the highest discovery rate, the PCA utilizes a relaxed problem formulation as follows. An upper bound on the discovery power, $\overline{\Phi}(1)$, is computed by using (9) to obtain $(1 - e^{-\lambda l}) \leq \lambda l$, which leads to,

$$\Phi(1) \leq \overline{\Phi}(1) := \frac{N - 1}{N\rho} \lambda l \left(P_r(\chi + M) + C_{sr} + C_{rs} \right). \qquad (11)$$

---

[6]Limited power budgets cause EH nodes to be in the sleep state much longer than in the listen state. Thus, $\lambda l \approx 0$ and (9) is a good approximation.

Panda Configuration Algorithm (PCA)

1: **for** $K = [0, \epsilon, 2\epsilon, \ldots, \lfloor \frac{\rho_{\max}}{\epsilon} \rfloor \epsilon]$ **do**
2:     Find (λ, l) that maximize (10) subject to (12)
3:     **if** λ, l satisfy (8) **then**
4:         Compute the discovery rate U
5: **return** (λ, l) that maximize U, denoted as $\lambda_A, l_A,$ and U_A.

The relaxed power budget constraint is then,

$$\text{s.t.} \quad \Phi(0) + \overline{\Phi}(1) \leq P_b. \tag{12}$$

The PCA analytically computes the values of (λ, l) that maximize $\overline{U}$ by solving for λ in terms of l in (11), and then finding the critical points where $\mathrm{d}\overline{U}/\mathrm{d}l = 0$. For computation tractability, the PCA replaces χ with a constant K in $\overline{\Phi}(1)$. The PCA uses the fact that, in practice, a node's sleep time is upper bounded, introducing an upper bound on the renewal duration $\rho_{\max}$. Thereby, the PCA sweeps values between $0 \leq \chi \leq \rho_{\max}$, and returns the best solution (i.e., the one that maximizes U). We denote the discovery rate that the PCA obtains by U_A and the configuration parameters by (λ_A, l_A).

C. An Upper Bound

To compute an *upper bound* on Panda's optimal discovery rate, we first derive a *lower bound* on the discovery power in an optimal solution, denoted by $\underline{\Phi}^*(1)$.

Theorem 1: The discovery power in an optimal solution, $\Phi^*(1)$, satisfies,

$$\Phi^*(1) \geq \frac{U_A}{N}(P_r M + C_{rs} + C_{sr}) := \underline{\Phi}^*(1), \tag{13}$$

where U_A is the discovery rate returned by the PCA.
The proof of Theorem 1 is in [13]. Using (10) and (13), an upper bound optimization problem is formulated as,

$$\max_{\lambda, l} \quad \overline{U}^* = (N-1)\lambda l/\rho \tag{14}$$

$$\text{s.t.} \quad \Phi(0) + \underline{\Phi}^*(1) \leq P_b. \tag{15}$$

Note that (15) effectively considers only a portion of the discovery power. This implies that the upper bound solution may be infeasible as it will incur an average power spending value higher than P_b. However, a solution maximizing (14) is in fact an upper bound on the optimal solution U^*, and is denoted by $\overline{U}^*$. To solve for $\overline{U}^*$, we first solve for λ with respect to the power constraint in (15). Then, we solve $\mathrm{d}\overline{U}^*/\mathrm{d}l = 0$ to obtain the listen time that maximizes $\overline{U}^*$ (for details, see [13]).

D. Performance of the PCA

We now compare the discovery rate from the PCA (U_A) to the upper bound discovery rate computed in Section V-C ($\overline{U}^*$). In this section, we will refer to the ratio $U_A/\overline{U}^* \leq 1$ as the *approximation ratio*. Values close to 1 imply that U_A is close to $\overline{U}^*$, and therefore, also close to the true optimal U^*.

Recall that the upper bound is computed by ignoring part of the discovery power, and therefore, violating the power constraint. Hence, when the discovery power $\Phi(1)$ in the optimal solution is indeed negligible (≈ 0), the upper bound $\overline{U}^*$ is close to the true optimal, U^*. Therefore, as the discovery power decreases, the approximation ratio approaches 1.
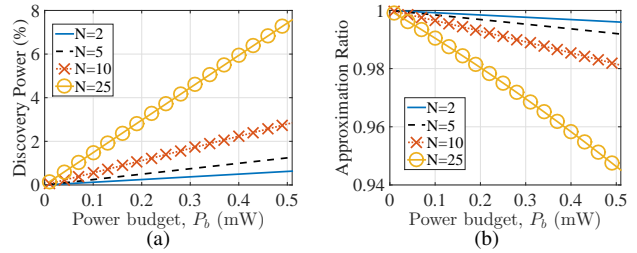

Fig. 3. The performance of the PCA for varying power budgets, P_b, and number of nodes, N: (a) discovery power as the percentage of the power budget ($\Phi(1)/P_b$), and (b) the approximation ratio ($U_A/\overline{U}^*$).

In Fig. 3 we show both the discovery power and the approximation ratio resulting from the configuration parameters returned by the PCA for varying $N \in \{2, 5, 10, 25\}$. In Fig. 3(a), we consider the discovery power as a proportion of the total power budget P_b. As shown, smaller values of N or P_b result in a smaller proportion of discovery power.

Fig. 3(b) shows the approximation ratio $U_A/\overline{U}^*$ as a function of the power budget P_b. First, note that the approximation ratio is always greater than 94% for all parameters considered. Therefore, because $U_A \leq U^* \leq \overline{U}^*$, the discovery rate provided by the PCA is within 6% of the optimal. Additionally, for larger values of N or P_b, the approximation ratio decreases. In this domain, the discovery power is larger (see Fig. 3(a)), and therefore, the discovery rate returned by the PCA (U_A) is further from the upperbound ($\overline{U}^*$), but not necessarily further from the optimal (U^*).

VI. PANDA-DYNAMIC (PANDA-D)

Panda is analyzed assuming that nodes are homogenous (A1), are arranged in a clique (A2), and the number of nodes N is known a priori (A3). However, when these assumptions do not hold, the expected power consumption of a node operating with Panda (see Section IV-D) will vary and the power budget is no longer satisfied. Therefore, in this section, we present Panda-Dynamic (Panda-D).

Panda-D operates with the same behavior as Panda, transitioning between the sleep, receive, and transmit states. However, to handle the varying power consumption with the relaxed assumptions, the rate of the exponential sleep duration is dynamic, and is adapted based on the voltage of the capacitor,[7] V_{cap}. Thereby, if a node consumes too much power, its voltage will decrease and it will adapt by staying in the sleep state for longer durations.

Formally, the configuration parameters for Panda-D are computed as follows. In this case, P_b represents an *estimated* power budget for each node, yet we allow for each node to harvest power at varying rates around P_b. The sleep duration is scaled such that the nodes' anticipated power consumption is $0.01\mathrm{mW}$ when $V_{\mathrm{cap}} = 3.6\mathrm{V}$, and is P_b when $V_{\mathrm{cap}} = 3.8\mathrm{V}$. From the two points, the desired power consumption of the

---

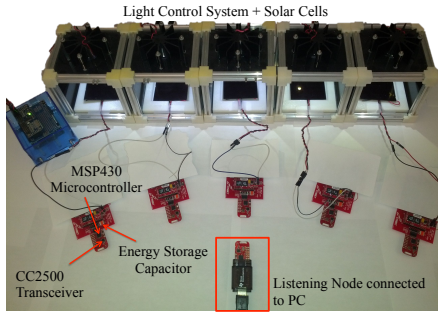[7]A similar adaptation mechanism was also proposed in [24].

Fig. 4. Panda experimental setup: 5 EH nodes harvest energy through the solar cells with neighbor discovery rates monitored by a listening node.
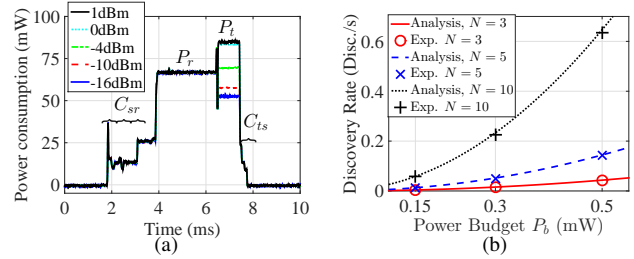

Fig. 5. (a) Power consumption for a node transitioning between the sleep, receive, and transmit states, and back to sleep. (b) Panda's discovery rate with varying power budgets, P_b, and number of nodes, N.

node, P_{des}, is computed as a linear function of the capacitor voltage (V_{cap}),

$$P_{\text{des}}(V_{\text{cap}}) = \frac{P_b - 0.01}{3.8 - 3.6}\left(V_{\text{cap}} - 3.6\right) + 0.01,\ 3.6 \leq V_{\text{cap}} \leq 4.$$

Based on the desired power consumption P_{des}, the node adjusts its sleep duration. As mentioned above, we cannot explicitly relate the sleep duration to the power consumption for a node. Instead, we will estimate the power consumption by ignoring the *discovery power*. That is, we assume that a node always follows the sleep, receive, transmit cycle and is spending on average at rate,

$$P_{\text{est}} = \frac{\eta(0)}{1/\lambda + l + M} = \frac{P_r l + P_t M + C_{sr} + C_{ts}}{1/\lambda + l + M}.$$

The average sleep duration, $1/\lambda$, is computed as a function of the capacitor voltage (V_{cap}) by solving $P_{\text{est}} = P_{\text{des}}$,

$$\frac{1}{\lambda} = \frac{P_r l + P_t M + C_{sr} + C_{ts}}{P_{\text{des}}(V_{\text{cap}})} - l - M. \tag{16}$$

We remark that the listen time l is obtained using the PCA with $N = 2$ (i.e., we try to maximize the discovery rate for each *directional link*).

We claim that the robustness of Panda-D is two-fold. First, it is power aware and thus nodes can operate under different and varying power harvesting rates, relaxing (A1). Additionally, it does not require any a priori knowledge of the size or topology of the network, relaxing (A2) and (A3).

VII. Experimental Performance Evaluation

We now evaluate Panda using a testbed, pictured in Fig. 4, composed of TI eZ430-RF2500-SEH [6] prototypes (described in Section III-A). First, we evaluate Panda in the context of the model presented in Section III-B. We compare Panda's experimental discovery rate, denoted by U_E, to related work. Then, we evaluate Panda-D in scenarios with non-homogeneous power harvesting and multihop topologies. Due to space constraints, we present Panda's performance with varying parameters (e.g., transmission power, message length) in [13].

A. Protocol Implementation

In accordance with Panda, the microcontroller builds the discovery message and sends it to the low-power transceiver. The message contains debugging information, the source ID of the transmitting node, and the node's capacitor voltage (which is sampled from the ADC). Additionally, the message includes the number of discoveries from each neighbor since the initialization of the experiment, referred to as the node's *neighbor table*. The transmission duration of the discovery message is 0.92ms.

In order to characterize the energy costs, we measure the power consumption of the microcontroller and transceiver using an oscilloscope. Fig. 5(a) shows the power levels for a node transitioning between the sleep, receive, and transmit states. We compute the average power consumption and transition energy for each state, with values summarized in Table I.

We note that the transition times to and from the sleep state are non-negligible (in some cases a few ms). To account for this, these transition times are considered as part of the sleep state and, are therefore, subtracted from the actual sleep duration. We elaborate further on the importance of incorporating these switching costs in [13].

The parameters in Table I compose the inputs to the PCA, which computes the rate of the exponential sleep λ_A and the duration of the listen state l_A as well as an expected discovery rate U_A. These *configuration parameters* are loaded into the nodes for experimental evaluation in which we observe the discovery rate as well as the power consumption.

B. Testbed and Experimental Setup

We consider networks of 3, 5, and 10 nodes ($N = 3, 5, 10$). We consider power budgets of $P_b = 0.15,\ 0.3,\ 0.5$mW; these are aligned with other solar harvesting budgets [7]. Initially, to confirm the practicality of Panda when assumptions (A1), (A2), and (A3) hold, we place the nodes in close proximity with a homogenous power budget. In Section VII-F, we will evaluate Panda-Dynamic (Panda-D) and relax these assumptions by considering a multihop topology and non-homogenous power harvesting.

To facilitate experimental evaluation with up to $N = 10$ nodes, in addition to an EH node shown in Fig. 1(b), we also incorporate nodes powered by AAA batteries into the experiments. Both the EH node and the node powered by AAA batteries operate using the same configuration parameters and hence have identical behaviors. However, we carefully logged the power consumption of the EH node by including control information in the discovery message.

We utilize a listening node consisting of a microcontroller and a transceiver set to a promiscuous sniffing mode to log experimental results. Powered by a USB port on a monitoring PC, the listening node reports all received messages to the

Param.	P_t	P_r	M	C_{sr}	C_{rs}	C_{tr}
Value	59.23mW	64.85mW	0.92ms	74.36μJ	13.48μJ	4.83μJ

N	P_b (mW)	λ_A^{-1}(ms)	l_A(ms)	Duty Cycle (%)	U_A (Disc./s)	U_E (Disc./s)	Error (%)	Run Time (h)
3	0.15	1778.68	2.066	0.168	.0039	.0038	-1.35	36
	0.3	887.39	2.070	0.336	.0156	.0154	-1.23	36
	0.5	530.88	2.075	0.561	.0434	.0438	1.07	48
5	0.15	1777.18	2.068	0.168	.0130	.0132	1.43	96
	0.3	885.91	2.075	0.337	.0519	.0518	-0.33	60
	0.5	529.43	2.084	0.564	.1443	.1427	-1.15	18
10	0.15	1773.49	2.075	0.169	.0584	.0589	0.89	18
	0.3	882.32	2.089	0.340	.2332	.2341	0.38	18
	0.5	525.97	2.107	0.572	.6470	.6510	0.62	18

PC for storage and post processing. The experimental discovery rate, U_E, is computed by dividing the total number of discoveries since the initialization of the experiment by the experiment duration. Clearly, the time until which the experimental discovery rate converges depends on the rate of discovery. Based on empirical results, all experiments are conducted for up to 96 hours (see [13] for details).

The light levels are set to correspond to each of the power budgets, P_b. However, the performance of the solar cells vary significantly due to external effects such as aging, orientation, and temperature [3]. To mitigate these affects and facilitate repeatable and controllable experiments, we designed a software controlled light system which we describe in [13].

Additionally, as mentioned in Section III, the prototype is not power aware. That is, although we can accurately measure the power harvested by the solar cell, it is difficult to control the energy actually stored in the capacitor, due to numerous inefficiencies of the harvesting circuitry, which are further described in [13]. As such, we empirically estimated the harvesting inefficiency to be 50% and adjust the light levels to provide each node energy according to the value of P_b chosen.

C. Discovery Rate

For each (N, P_b) pair, we evaluate Panda, with the experimental parameters summarized in Table II. First, we note that Panda's duty cycle is typically between 0.1–0.6%, which is significantly lower than the duty cycles considered in related protocols [9]. Additionally, note the accuracy of the analytical discovery rate, U_A, computed from (2), compared to the experimental discovery rate, U_E. On average, the error between them is $\approx 1\%$. This confirms the practicality of Panda and the model described in Section III.

In Fig. 5(b), we plot the experimental and analytical discovery rate for each value of (N, P_b) shown in Table II and observe the effect of varying N and P_b. As expected, the discovery rate increases as P_b increases. The number of nodes N is directly correlated with the discovery rate, as indicated in (2) and (3). As such, the discovery rate increases as N increases. Additionally, by tracking each nodes' neighbor

table, we confirm that all nodes discover one another and exhibit similar per link discovery rates (see [13] for details).

D. Discovery Latency and Comparison to Related Work

The discovery latency is the time between consecutive discoveries for a *directional link*. It can be an important parameter for numerous applications where nodes are only within communication range for short periods of time. Although the objective of Panda is to maximize the discovery rate, in Fig. 6(a), we show the CDF of the discovery latency for each *directional link* in an experiment with $N = 5$ and varying power budgets. Clearly, the average discovery latency decreases as the average discovery rate increases. Thus, for a higher power budget, the discovery latency decreases.

Previous work [10], [11], [14] focused on minimizing the worst case discovery latency for a link. We compare the discovery latency of Panda, shown in Fig. 6(a), to previous work. However, as mentioned in Section II, previous work considers a duty cycle constraint instead of a power budget (P_b). To provide a means of comparison, we use the following equation to relate the power constraint to a duty cycle.

$$P_b = \text{Duty Cycle}(\%) \cdot \text{Average Active Power (mW)} \quad (17)$$

We compare to the deterministic Searchlight protocol [10], which minimizes the worst case discovery latency [11]. We also compare to the the well-known probabilistic Birthday (BD) protocol [12]. To account for the power budget, we modify these protocols based on (17) (with details explained in [13]) and denote them as Searchlight-E and BD-E. Based on previous work [11], we set the slot size for Searchlight-E and BD-E to 50ms and add an overflow guard time of 1ms.

In Fig. 6(b), we compare the average discovery rate for Panda vs. simulations of the Searchlight-E and BD-E protocols. We found that Panda typically outperforms the Searchlight-E and BD-E protocols by over 3x in terms of the average discovery rate.[8]

Furthermore, in Fig. 6(c), we consider the *worst case* discovery latency and show that although Panda has a non-zero probability of having any discovery latency, for the experiments we considered, the 99th percentile of discovery latency outperformed the Searchlight-E protocol worst case bound by up to 40%.

Note that the Searchlight protocol was proven to minimize the worst case discovery latency. However, as shown through our evaluation, Panda outperforms Searchlight-E by a factor of 3x in terms of average discovery rate. Moreover, in most cases (over 99%), the discovery latency is below the worst case bound from Searchlight-E. This emphasizes the importance of incorporating a detailed power budget, as is done in Panda, as opposed to a duty cycle constraint.

E. Power Consumption

Using Panda, a node consumes power at a rate of up to P_b (mW), *on average*. However, the power consumption is

---

[8]As described in [13], the simulations of Searchlight-E and BD-E do not account for packet errors or collisions. As such, the discovery rates for these protocols is likely to be lower in practice.
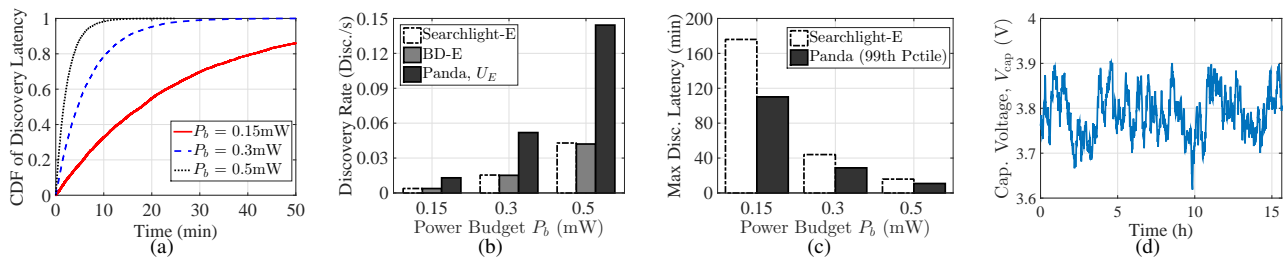
Fig. 6. Panda experimental evaluation for $N = 5$: (a) CDF of per link discovery latency; comparison to SearchLight-E [10] and BD-E [12] of (b) the discovery rate and (c) the worst case latency; (d) capacitor voltage for a node with a 30mF capacitor and $P_b = 0.5$mW.

stochastic, and therefore, it is expected that the energy stored will vary over time. In Fig. 6(d), we show the capacitor voltage over time for a node with $N = 5$ and $P_b = 0.5$mW. Energy neutrality is demonstrated by the oscillation in the energy level within the limits of the capacitor storage. Recall from Section III that if the energy drains below a software induced threshold of 3.6V, the node temporarily sleeps for 10s to regain energy. These periods of additional sleep affect the discovery rate and, as indicated by the accuracy of the experiments, these occurrences are rare. In [13], we describe the implications of varying capacitor sizes (from 10-50mF).

F. Panda-Dynamic

We now evaluate Panda-D (described in Section VI). The only input to Panda-D is the estimated power harvesting rate, $P_b = 0.15$mW, and the capacitor voltage V_{cap}. From (16), the average duration of the exponential sleep is then computed as,

$$\frac{1}{\lambda} = \frac{382.2238}{V_{\text{cap}} - 3.5857} - 2.9843 \text{ (ms)}. \qquad (18)$$

Thus, the node scales its power consumption based on V_{cap}. For example, at $V_{\text{cap}} = 3.6$V and 4V, the node will sleep on average for 26.75 and 0.92 seconds, respectively.

To estimate the average sleep duration for a given node in Panda-D, we compute the average value of V_{cap} over the course of an experiment. Based on the this value, the average sleep duration is estimated from (18).

Panda-D does not require a priori information of the number of neighbors, N. Therefore, throughout this section, (A3) is relaxed. Below, we observe the performance of Panda-D first when (i) nodes remain in a clique topology with homogenous power budgets. Then we consider Panda-D (ii) in a multihop topology (relaxing (A2)), and finally (iii) in non-homogenous power harvesting scenario (relaxing (A1)). Relaxing all assumptions together requires running a live real-world experiment and is a subject of future work.

(i) Comparison to Panda: We first evaluate Panda-D with an experimental setup similar to the one shown in Fig. 4. Specifically, we consider a network of $N = 3$ nodes in close proximity with a power harvesting rate of $P_b = 0.15$mW.

As shown in Fig. 7(a), the capacitor voltage for all 3 nodes stays approximately near 3.8V. As described in Section VI, the average power consumption at 3.8V is approximately P_b. Therefore, in this scenario, Panda-D and Panda have similar power consumption and discovery rates. As such, the experimental discovery rate of Panda-D is within 1% of the analytical estimate of Panda.

(ii) Multihop Topologies: Previously, we assumed that all nodes form a clique topology with no packet losses (A2) and the number of nodes N known (A3). Indeed, for the experiments conducted above with a transmission power of -10dBm, we found that nodes within $\approx$20m could be treated as a clique topology with over 99% packet success rates.

However, to evaluate a non-clique topology and relax (A2) and (A3), we manually reconfigured the transmission power to -26dBm and set 3 nodes in a line topology with distance between nodes 1-2 and 2-3 of 1.5m, as shown in Fig. 7(b). In this configuration, nodes rarely receive messages from their two-hop neighbors. Nodes run Panda-D and are given light levels corresponding to the power harvesting rate of $P_b = 0.15$mW (as described in Section VII-B). After 50 hours, the resulting discovery rate is shown on each link in Fig. 7(b).

The two extreme nodes (nodes 1 and 3) have very few discoveries from one other, due to the distance between them. However, the node in the middle (node 2) forms an effective clique of size 2 with each of its neighbors. We therefore can analyze the discovery rate per link. For example, the discovery rate of the link between nodes 1 and 2 is 0.0051 disc./s, which is within 1% of the analytical discovery rate for a clique with $N = 2$ and $P_b = 0.15$mW. Therefore, even with non-clique topologies, each link that is within communication range can be analyzed as a network with $N = 2$. This implies that issues such as the hidden-node problem do not significantly affect the performance of Panda.

(iii) Non-Homogeneous Power Harvesting: We now consider nodes 2–5 using Panda-D with light levels corresponding to power harvesting of 0.075, 0.15, 0.225, 0.3mW, respectively. Node 1 is a *control* node running Panda with $P_b = 0.15$mW and $N = 5$.

For each of the 4 Panda-D nodes, the capacitor voltage, V_{cap}, is shown in Fig. 8(a) and settles based on the power harvesting. Variations in the settling voltage stem from the dynamic average sleep duration at different power harvesting levels. For example, node 5 is given a light level of 0.3mW, and therefore, has a shorter sleep duration than node 2 (light level of 0.075mW). Correspondingly, Fig. 8(b) shows the neighbor table: entry (i,j) represents the number of discoveries of node j by node i over the experiment duration. Due to non-homogeneity, the discovery rate for each link depends on the power harvested; nodes with larger power budgets discover their neighbors, and are discovered, more frequently.

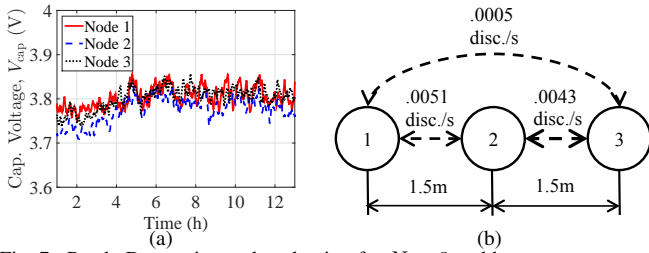In [13], we treat each link with non-homogenous power

Fig. 7. Panda-D experimental evaluation for $N = 3$ and homogeneous power harvesting $P_b = 0.15$mW: (a) Capacitor voltage, V_{cap}, in a clique topology. (b) Per link experimental discovery rates for a line topology after 50 hours.

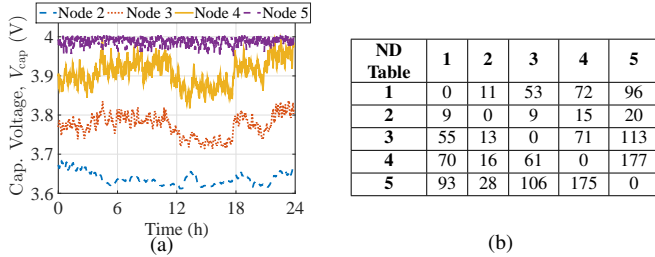| ND Table	1	2	3	4	5
1	0	11	53	72	96
2	9	0	9	15	20
3	55	13	0	71	113
4	70	16	61	0	177
5	93	28	106	175	0

Fig. 8. Panda-D experimental evaluation for non-homogeneous power harvesting with $N = 5$ over 24 hours: (a) Capacitor voltage, V_{cap}, and (b) resulting neighbor table.

harvesting as a clique ($N = 2$), and estimate its discovery rate; the approximation is within 20% of the experimental value.

VIII. Conclusions and Future Work

We designed, analyzed, and evaluated Panda, an ND protocol for EH nodes. By accounting for specific hardware constraints (e.g., transceiver power consumption for transmission, reception, and state switching), Panda adheres to a *power budget*. Using renewal theory, we developed the Panda Configuration Algorithm (PCA) to determine the nodes' sleep and listen durations which maximize the discovery rate; the PCA achieves a nearly-optimal discovery rate (over 94%).

We evaluated Panda using TI eZ430-RF2500-SEH EH nodes. The real-life accuracy was consistently within 2%, demonstrating the practicality of our model. Furthermore, Panda outperformed the closest related protocols Searchlight-E [10] and BD-E [12] by achieving a discovery rate that was up to 3x higher. Finally, we showed that a version of the protocol, Panda-Dynamic, was able to adapt to scenarios with non-homogeneous power harvesting and multihop topologies.

Panda can be readily applied to nodes with a non-rechargeable battery, where the power budget is set based on the desired lifetime. Future work will consider relaxing additional assumptions of our model. Primarily, we will attempt to *optimize* Panda-D in the presence of nodes with heterogenous power budgets in non-clique topologies.

Acknowledgements

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[2] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Mar. 2015.

[3] R. Margolies, M. Gorlatova, J. Sarik, G. Stanje, J. Zhu, B. Vigraham, M. Szczodrak, L. Carloni, P. Kinget, I. Kymissis, and G. Zussman, "Energy-harvesting active networked tags (EnHANTs): Prototyping and experimentation," *ACM Trans. Sens. Netw.*, vol. 11, no. 4, Nov. 2015.

[4] J. Wang and D. Katabi, "Dude, where's my card?: RFID positioning that works with multipath and non-line of sight," in *Proc. ACM SIGCOMM'13*, Aug. 2013.

[5] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. R. Smith, "Ambient backscatter: wireless communication out of thin air," in *Proc. ACM SIGCOMM'13*, Aug. 2013.

[6] "Texas Instruments MSP430 solar energy harvesting development tool," focus.ti.com/docs/toolsw/folders/print/ez430-rf2500-seh.html, 2012.

[7] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," *IEEE Trans. on Mobile Comput.*, vol. 12, no. 9, pp. 1853–1865, Sept. 2013.

[8] M. Radi, B. Dezfouli, K. A. Bakar, S. A. Razak, and M. Lee, "Network initialization in low-power wireless networks: a comprehensive study," *The Computer Journal*, vol. 57, no. 8, pp. 1238–1261, 2014.

[9] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1448–1459, 2014.

[10] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *Proc. ACM MobiCom'12*, Aug. 2012.

[11] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in *Proc. IEEE INFOCOM'14*, 2014.

[12] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. ACM MobiHoc'01*, Oct. 2001.

[13] R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, "Panda: Neighbor discovery on a power harvesting budget," *ArXiv e-prints*, Jan. 2016. [Online]. Available: http://arxiv.org/abs/1601.06474

[14] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. ACM SenSys'08*, Nov. 2008.

[15] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. ACM/IEEE IPSN'10*, Apr. 2010.

[16] A. Purohit, B. Priyantha, and J. Liu, "WiFlock: Collaborative group discovery and maintenance in mobile sensor networks," in *Proc. ACM/IEEE IPSN'11*, Apr. 2011.

[17] L. You, Z. Yuan, P. Yang, and G. Chen, "ALOHA-like neighbor discovery in low-duty-cycle wireless sensor networks," in *Proc. IEEE WCNC'11*, Mar. 2011.

[18] J. Taneja, J. Jeong, and D. Culler, "Design, modeling, and capacity planning for micro-solar power sensor networks," in *Proc. IEEE IPSN'08*, Apr. 2008.

[19] P. Zhang and D. Ganesan, "Enabling bit-by-bit backscatter communication in severe energy harvesting environments," in *Proc. USENIX NSDI'14*, Apr. 2014.

[20] Y.-s. Kuo, P. Pannuto, G. Kim, Z. Foo, I. Lee, B. Kempke, P. Dutta, D. Blaauw, and Y. Lee, "Mbus: A 17.5 pJ/bit/chip portable interconnect bus for millimeter-scale sensor systems with 8 nW standby power," in *Proc. IEEE CICC*, 2014.

[21] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proc. IEEE ISCC'04*, July 2004.

[22] A. Dunkels, "The ContikiMAC radio duty cycling protocol," Dec. 2011, Swedish Institute of Computer Science.

[23] "Texas Instruments CC2640 SimpleLink bluetooth smart wireless MCU," ti.com/lit/ds/symlink/cc2640.pdf, 2015.

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

[25] E. Cinlar, "Markov renewal theory," *Advances in Applied Probability*, vol. 1, no. 2, pp. 123–187, 1969.