Panda: Neighbor Discovery on a Power Harvesting Budget

Robert Margolies, *Member, IEEE*, Guy Grebla, *Senior Member, IEEE*, Tingjun Chen, *Student Member, IEEE*, Dan Rubenstein, *Member, IEEE*, and Gil Zussman, *Senior Member, IEEE*

***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**

I. Introduction

Object tracking and monitoring applications are gaining popularity within the realm of the Internet-of-Things [2]. 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., [3], [4] 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 [5]–[7]. 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.

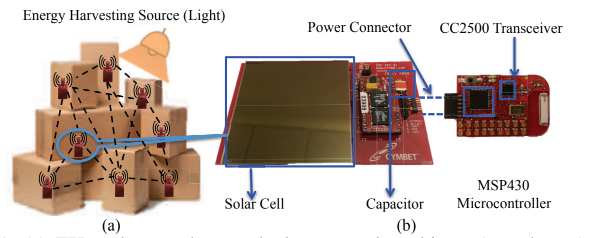


Fig. 1: (a) EH nodes can be attached to everyday objects (e.g., boxes) and utilize neighbor discovery protocols for inventory tracking and monitoring applications. (b) The prototype EH node stores harvested energy in a capacitor and powers the microcontroller and transceiver.

In this paper, *we develop an ND protocol for Commercial Off-The-Shelf (COTS) EH nodes, based on the TI eZ430-RF2500-SEH* [8] (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 [9]. 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 [10], [11]. 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., [12], [13]) 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–**

This research was supported by NSF grants CCF-09-64497 and CNS-10-54856, and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no. [PIIF-GA-2013-629740].11. A partial and preliminary version of this paper appeared in IEEE INFOCOM'16, Apr. 2016 [1].

R. Margolies, G. Grebla, T. Chen and G. Zussman are with the Department of Electrical Engineering, Columbia University, New York, NY 10027, USA (e-mail: {robm, guy, tingjun, gil}@ee.columbia.edu).

D. Rubenstein is with the Department of Computer Science, Columbia University, New York, NY 10027, USA (e-mail: danr@cs.columbia.edu).

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:

(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. We show that characteristics such as the power consumption and the time to transition between the different states (e.g., sleep to listen) are crucial to incorporate into the design of ND protocols for EH nodes.

(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 within 0.25% of the optimal for all scenarios considered.

(C4) Experimental Evaluation: Using TI eZ430-RF2500-SEH EH nodes [8], 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 [12], [14]. 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.

II. Related Work

ND for low-power wireless networks is a well studied problem (see [2], [10], [11] for a summary). The protocols can be categorized into deterministic (e.g., [12], [13], [15]–[17]) and probabilistic (e.g., [14], [18]). Probabilistic protocols (e.g., [14]) randomly rotate between the sleep, listen, and transmit states and have been shown to have a higher average ND rate, but suffer from an unbounded discovery latency [11]. Deterministic protocols select active states based on using prime numbers or "quorum" techniques. Thus, they are able to guarantee an upper bound on *discovery latency*, while the choice of parameters (e.g., prime numbers) is often limited. For reference, we describe the mechanisms for the closest related protocols (i.e., Searchlight [12] and Birthday [14]) in Appendix D.

[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.

TABLE I: Modifications to the TI eZ430-RF2500

Component	Problem/Modification
Energy Storage	On-board battery cannot be monitored; disable on-board battery and replace with an external capacitor.
Solar Panel	On-board solar cell cannot be monitored; disable on-board cell, measure power harvested by connecting a ammeter in series with solar cell from [26].
Power Consumption	Unable to track power consumed; measure consumed power with an oscilloscope across a 10Ω sense resistor, placed in series with the transceiver and the microcontroller.
12kHz Clock Source	Clock frequency varies by up to 20% for each node; rectified by manually measuring/calibrating the number of clock ticks in one second for each device.

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). 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 [11].

In our experiments, we use hardware from [8]. There are also numerous other hardware options for EH nodes [4], [19], [20], computational RFIDs [21], and mm^3-scale wireless devices [22]. Additionally, there are other radio features that achieve low energy consumption. For example, preamble-sampling and wake up radios were investigated in [23] and [24], 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 [25]). However, [8] 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 [8]. We made some modifications to the hardware as summarized in Table I.

We now describe the prototype's components:

Energy Harvesting Power Source: The prototype harvests light from a Sanyo AM 1815 amorphous solar cell [26]. The solar cell is set to a fixed harvesting voltage of 1.02V (no power point tracking techniques are used). To measure the

[2]It was shown in [15] how the aligned time slot assumption can be relaxed. Yet, practical considerations such as selecting the slot duration and avoiding collisions are not described.

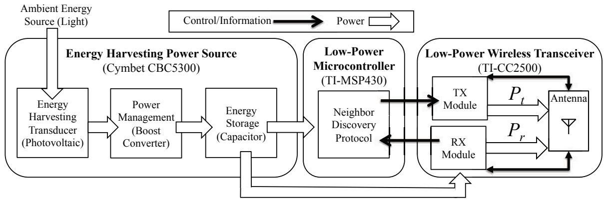


Fig. 2: A hardware model of the EH node (based on the TI eZ430-RF2500-SEH [8]).

power harvested, we place an 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.

In practice, a node can withstand variability in the energy harvesting and consumption, so long as the storage is sufficiently sized. There are numerous related works discussing the impact of finite energy storage sizes (e.g., [27]) that are out of the scope of this work. Thus, unless stated otherwise, we use a 30mF capacitor. To ensure stable voltage regulation, a software cutoff is imposed; if $V_{\text{cap}} \leq 3.6\text{V}$, 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 [28] 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) [29] 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 and is shown pictorially in Fig. 2. Yet, it is generalizable to a class of other prototypes (e.g., [4]). A summary of the nomenclature from this point forward appears in Table II.

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 Appendix A. 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 ($i, j \in \mathcal{S}$).

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

TABLE II: Nomenclature

Symbol	Description
V_{cap}	The voltage of the capacitor (V)
N	Number of nodes
P_b	Average power spending budget (mW)
P_t	Transmitting power consumption (mW)
P_r	Listening/Receiving power consumption (mW)
C_{ij}	Energy cost to transition from state i to j (µJ).
M	Discovery-packet duration (ms)
λ	Rate of exponential distribution (ms^{-1})
l	The duration of the listening period (ms)
ρ	Expected renewal duration (ms)
Y	Denotes role of node in the renewal
$\eta()$	Expected energy spending (µJ)
$\Phi()$	Expected power spending in a renewal (mW)
χ	Expected duration of idle listening (ms)
U	Discovery rate (second^{-1})

Unfortunately, the prototype does not have explicit power awareness (unlike, e.g., [4]). 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 [30]. Hence, for an EH node harvesting more power (e.g., brighter light source), 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 [12], [13], subject to a duty cycle. The discovery latency, or time between successive discoveries, is very much related to the discovery rate. In fact, the inverse of the average discovery rate is indeed the average discovery latency. Thereby, maximizing the average discovery rate is quite similar to minimizing the worst case latency.

In practice, both the average discovery rate and the worst-case latency are important to applications [11]. Reduced worst-case latency is important in cases where nodes are only collocated for short periods of time. However, in general, applications (such as the one described in Section I) must be designed to handle the occasional missed-discovery and would receive greater benefit from a higher average discovery rate. As such, we focus on maximizing the average discovery rate and in Section VII, we also consider the discovery latency 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.

[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.

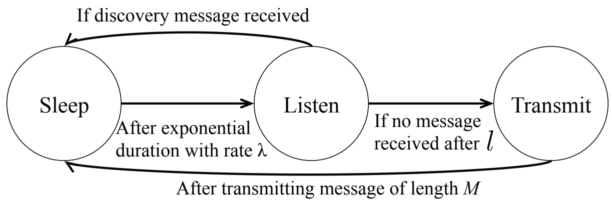


Fig. 3: Panda protocol outline: EH nodes transition between radio states (sleep, listen, and transmit) to maintain within a power budget.

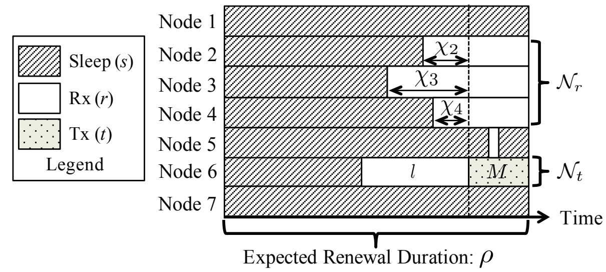


Fig. 4: Renewal process representing one renewal cycle for $N = 7$ nodes: all nodes begin in a memoryless sleep state and the renewal restarts after the first node to wake up completes its transmission.

A. Protocol Description

Fig. 3 depicts the state transition diagram for the Panda protocol, from sleep to listen to transmit and then back to sleep. 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 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. Therefore, a node's transmission will never collide with an ongoing transmission from a node that is within wireless communication range. 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 (e.g., [31]).

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.

We note that, as this is the first attempt to develop an ND protocol explicitly for EH nodes, it is natural to consider the homogeneity assumption (A1). Additionally, several other works also make assumptions similar to (A2) and (A3) (e.g., [12], [13] only consider ND for a link, $N = 2$, with no collisions or packet loss).

Using these assumptions, we now use techniques from renewal theory [32] to analyze Panda for a network of N nodes. The renewal process is shown pictorially in Fig. 4. 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. 4; 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. 4. The expected idle listening time of a node in $\mathcal{N}_r$ is denoted by χ. Fig. 4 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. 4.

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

$$\rho = 1/(\lambda N) + l + M. \tag{1}$$

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 [32], we obtain:

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

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

[5]The discovery message duration, M, is fixed, stemming from the fixed size of the node ID contained in the message.

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 $1-\mathrm{e}^{-\lambda l}$. Hence,

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

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:

$$Y=0 \text{ if } n\in\mathcal{N}_t; Y=1 \text{ if } n\in\mathcal{N}_r; Y=2 \text{ else}. \tag{4}$$

The expected power consumed (mW) for a node in which $Y=y$ is denoted as $\Phi(y)$. It is computed as, $\Phi(y) = \Pr(Y=y)\eta(y)/\rho$, where $\eta(y)$ represents the expected amount of energy (µJ) consumed by a node in a renewal in which $Y=y$ and ρ is the expected renewal duration. The expected power consumed in a renewal must meet the power budget, and thus, $\Phi(0)+\Phi(1)+\Phi(2)\le P_b$. The remainder of this section is used to derive $\Phi(0)$, $\Phi(1)$, and $\Phi(2)$. We will often refer to $\Phi(1)$ as the *discovery power*.

$$\eta(0) = C_{sr} + P_r l + P_t M + C_{ts}, \tag{5}$$
$$\Pr(Y=0) = 1/N. \tag{6}$$

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}). By definition of the renewal, there will be exactly one transmitter in a renewal and due to assumption (A1), $\Pr(Y=0)=1/N$.

$$\eta(1) = C_{sr} + P_r(\chi+M) + C_{rs}, \tag{7}$$
$$\Pr(Y=1) = \frac{N-1}{N}(1-\mathrm{e}^{-\lambda l}). \tag{8}$$

For a receiving node, the expected energy consumption is defined in (7). A receiving node consumes energy to wake up from sleep (C_{sr}), idle listen before the message transmission (for a duration of χ), and receive for the duration of the message M, and then return to sleep (C_{rs}). As shown in Fig. 4, χdenotes the expected duration of idle listening before receiving a message. We derive it without loss of generality, by assuming the transmitter in a renewal ($\mathcal{N}_t$, e.g., node 6 from Fig. 4) enters the listen state at $t=0$, and at $t=l$, it transmits the discovery message. Let x denote the idle listening time for a given node where x is exponentially distributed with $0<x<l$. We look to find $\chi = \mathbb{E}\left[x|x<l\right]$, i.e.,

$$\begin{aligned}\chi &= \int_{t=0}^{+\infty} \Pr(x>t|x<l)\,\mathrm{d}t \\ &= \int_{t=0}^{l} \frac{(1-e^{-\lambda l})-(1-e^{-\lambda t})}{1-e^{-\lambda l}}\,\mathrm{d}t = \frac{1}{\lambda} - \frac{l e^{-\lambda l}}{1-e^{-\lambda l}}\end{aligned}$$

Eq. (8) defines the likelihood of $n\in\mathcal{N}_r$. Conditioned on $n\notin\mathcal{N}_t$ w.p. $(N-1)/N$, a node successfully receives the message, if it starts listening in a period of length l preceding the transmission. Ssince the sleep duration is exponentially distributed, this is an event occurring with probability, $(1-\mathrm{e}^{-\lambda l})$.

Throughout this paper, we assume that nodes which sleep for the entire renewal (e.g., nodes 1 and 7 in Fig. 4), 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 Appendix A, we show how it can be relaxed.

Panda Configuration Algorithm (PCA)

---

for $K = [0, \epsilon, 2\epsilon, \ldots, \lfloor\frac{\rho_{\max}}{\epsilon}\rfloor\epsilon]$ **do**
    Find (λ, l) that maximize (12) s.t. (14)
    if λ,l satisfy (10) **then**
        Compute the discovery rate U
return (λ, l) that maximize U, denoted as λ_A, l_A, and U_A.

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:

$$\max_{\lambda,l} \quad U = (N-1)(1-\mathrm{e}^{-\lambda l})/\rho \tag{9}$$
$$\text{s.t.} \quad \Phi(0)+\Phi(1)\le P_b, \tag{10}$$

where (9) 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[6] 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, the following Taylor-series approximation is useful:

$$\mathrm{e}^{-x}\ge 1-x \text{ for } x\ge 0, \text{ and } \mathrm{e}^{-x}\approx 1-x \text{ for } x\approx 0. \tag{11}$$

We substitute x with λl in (11),[7]

$$U \le (N-1)\lambda l/\rho := \overline{U}. \tag{12}$$

B. Panda Configuration Algorithm (PCA)

The Panda Configuration Algorithm (PCA) returns a configuration of λ and l that satisfy (10). 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 (11) to obtain $(1-\mathrm{e}^{-\lambda l})\le\lambda l$, which leads to,

$$\Phi(1)\le\overline{\Phi}(1) := \frac{N-1}{N\rho}\lambda l\left(P_r(\chi+M)+C_{sr}+C_{rs}\right). \tag{13}$$

[6]The non-convexity of the optimization problem is straightforward to prove by taking second order partial derivatives.

[7]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 (11) is a good approximation.

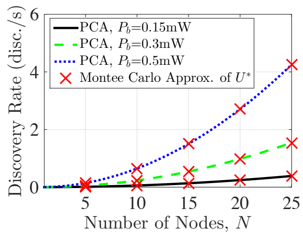


Fig. 5: The performance of the PCA for varying number of nodes, N: the discovery rates resulting from the PCA (U_A) compared to a Monte Carlo approximation of U^*.

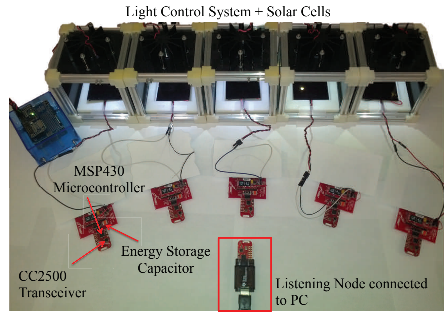


Fig. 6: Panda experimental setup: 5 EH nodes harvest energy through the solar cells with neighbor discovery rates monitored by a listening node.

The relaxed power budget constraint is then,

$$\text{s.t.} \quad \Phi(0) + \overline{\Phi}(1) \leq P_b. \tag{14}$$

The PCA analytically computes the values of (λ, l) that maximize $\overline{U}$ by solving for λ in terms of $l$in (13), 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).

Fig. 5 demonstrates the discovery rate of Panda for various power budgets and number of nodes. The performance of PCA is compared to the discovery rate provided by Monte Carlo solution to (9), denoted as U^*.[8] In all test cases considered in Section VII (see Table V), the PCA was within 0.25% of the discovery rate of the Monte Carlo simulation demonstrating the near-optimality of the PCA for the parameters we consider.

VI. Panda-Dynamic (Panda-D)

As described in Section III, the objective of the ND protocol is to maximize the discovery rate, subject to a power budget. Panda is analyzed and optimized 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 attempts to maximize the discovery rate by operating with the same behavior as Panda, transitioning between the sleep, receive, and transmit states. However, to achieve energy neutrality in the general setting with the relaxed assumptions, the rate of the exponential sleep duration is dynamic, and is adapted based on the voltage of the capacitor,[9] 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 0.01mW when $V_{\text{cap}} = 3.6\text{V}$, and is P_b when $V_{\text{cap}} = 3.8\text{V}$. From the two points, the desired power consumption of the node, P_{des}, is computed as a linear function of the capacitor voltage (V_{cap}),

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

Based on the desired power consumption P_{des}, a node adjusts its sleep duration. As mentioned above, we cannot explicitly relate the sleep duration to the power consumption for a node. Instead, we 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 V_{cap} by solving $P_{\text{est}} = P_{\text{des}}$,

$$1/\lambda = \frac{P_r l + P_t M + C_{sr} + C_{ts}}{P_{\text{des}}(V_{\text{cap}})} - l - M. \tag{15}$$

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 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), (A3).

VII. Experimental Performance Evaluation

We now evaluate Panda using a testbed, pictured in Fig. 6, composed of TI eZ430-RF2500-SEH [8] 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. Additionally, we present Panda's performance with varying parameters (e.g., transmission power, message length). Then, we evaluate Panda-D in scenarios with non-homogeneous power harvesting and multihop topologies.

[8]The Monte Carlo solution generates over 10^8 random configuration parameters and returns the (λ, l) that satisfies (10) with the largest ND rate.

[9]A similar adaptation mechanism was also proposed in [30].

TABLE III: Discovery message structure.

Byte	Data
0	Packet length (18 bytes)
1	Type
2-11	Neighbor table
12-13	Capacitor voltage
14-15	Debugging information
16	Transmissions counter
17	Originating node ID

TABLE IV: Measured prototype parameters.

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

A. Protocol Implementation

In accordance with Panda, the microcontroller builds the discovery message and sends it to the low-power transceiver. Table III illustrates the structure of the discovery message. 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 total length of a discovery message is 18 bytes and the resulting transmission duration of the discovery message is $0.92\mathrm{ms}$.

In order to characterize the energy costs, we measure the power consumption of the microcontroller and transceiver using an oscilloscope. Fig. 7(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 IV.

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 Appendix B.

The parameters in Table IV 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 $P_b = 0.15, 0.3, 0.5\mathrm{mW}$; these are aligned with other solar harvesting budgets [9]. 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-G, 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 nodes powered by AAA batteries operate using the same configuration parameters and have identical behaviors (i.e., the source of power does not affect the behavior of Panda). However, we carefully logged the power consumption of the EH node by including control information in the discovery message (see Table III).

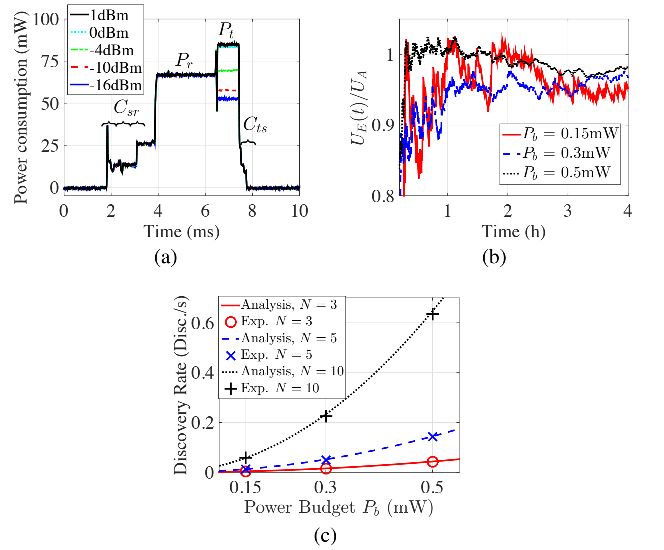


Fig. 7: (a) Power consumption and transition costs for different transmission power levels for a node transitioning between the sleep, receive, and transmit states, and back to sleep. (b) Panda experimental evaluation with varying power budgets, P_b: Convergence of the experimental discovery rate (U_E) to the analytical discovery rate (U_A) for $N = 5$. (c) Panda's discovery rate with varying power budgets, P_b, and number of nodes, N

TABLE V: Panda experimental parameters: (λ_A, l_A) generated using the PCA for every input (N, P_b) pair and the resulting analytical (U_A) and experimental (U_E) discovery rate.

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

We utilize a listening node consisting of a microcontroller and 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 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. In Fig. 7(b), we observe the experimental discovery rate, U_E over time for $N = 5$ and $P_b = 0.15, 0.3, 0.5\mathrm{mW}$. Based on the results shown in Fig. 7(b), all experiments were conducted for up to 96 hours.

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 [4]. To mitigate these affects and facilitate repeatable and controllable experiments, we designed a software

TABLE VI: Neighbor table for $N = 5$, $P_b = 0.3$mW after 4 hours. Entry (i,j) shows the number of discoveries of node j by node i.

ND Table	1	2	3	4	5	Total RX
1	0	35	42	32	42	152
2	24	0	23	45	38	130
3	39	36	0	21	33	129
4	36	35	46	0	32	149
5	38	42	42	42	0	164

controlled light system which we describe in Appendix C.

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 Appendix C. 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 V. 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 [11]. 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. 7(c), we plot the experimental and analytical discovery rate for each value of (N, P_b) shown in Table V 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 in Table VI, we confirm that all nodes discover one another and exhibit similar per link discovery rates.

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. 8(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 [12], [13], [15] focused on minimizing the worst case discovery latency for a link. We compare the discovery latency of Panda, shown in Fig. 8(a), to previous work. However, as mentioned in Section II, previous work considers a duty cycle constraint instead of a power budget

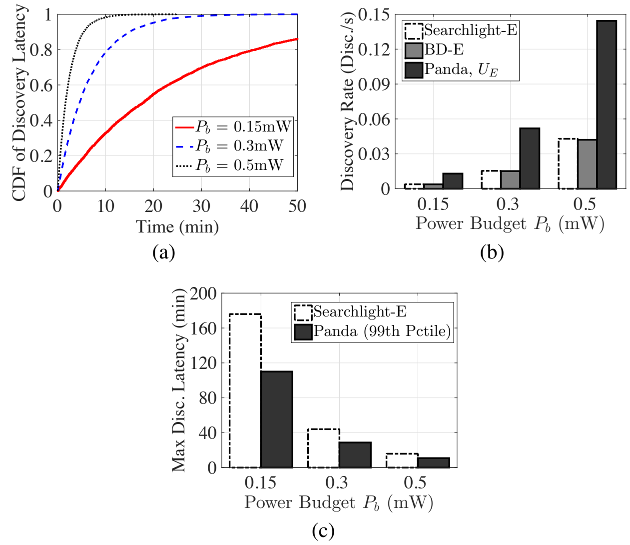


Fig. 8: Panda experimental evaluation for $N = 5$: (a) CDF of per link discovery latency; comparison to SearchLight-E [12] and BD-E [14] of (b) the discovery rate and (c) the worst case latency.

(P_b). To provide a means of comparison, we use the following equation to relate the power constraint to a duty cycle.

$$P_b = \text{Duty Cycle}(\%) \cdot \text{Average Active Power (mW)} \quad (16)$$

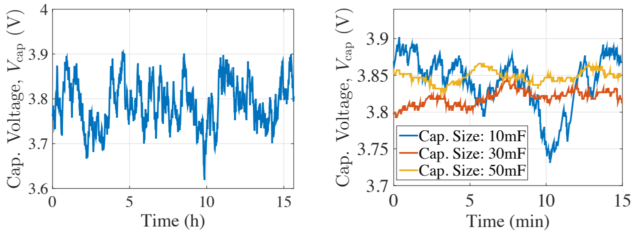
We compare to the deterministic Searchlight protocol [12], which minimizes the worst case discovery latency [13]. We also compare to the the well-known probabilistic Birthday (BD) protocol [14]. To account for the power budget, we modify these protocols based on (16) (with details explained in Appendix D) and denote them as Searchlight-E and BD-E. Based on previous work [13], we set the slot size for Searchlight-E and BD-E to 50ms and add an overflow guard time of 1ms.

In Fig. 8(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.[10]

Furthermore, in Fig. 8(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 99^{th} percentile of discovery latency outperformed the Searchlight-E protocol worst case bound by up to 40%.

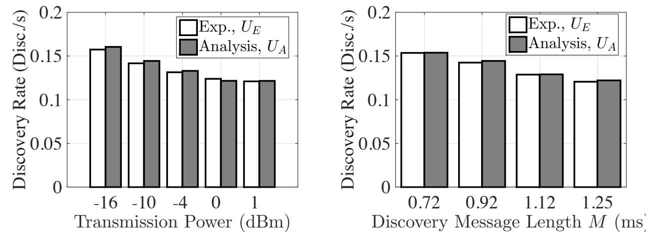
Note that the Searchlight protocol was proven to minimize the worst case discovery latency. However, as shown through our evaluation, Panda outperforms Searchlight-E by a factor of 3x in terms of average discovery rate. Moreover, in most cases (over 99%), the discovery latency is below the worst case 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.

[10]As described in Appendix D, 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.



(a) (b)

Fig. 9: Panda experimental evaluation for $N = 5$: (a) capacitor voltage level (V_{cap}) for a node with a 30mF capacitor and $P_b = 0.5\text{mW}$, and (b) capacitor voltage level of nodes with varying capacitor sizes over 15 minutes with $P_b = 0.15\text{mW}$.



(a) (b)

Fig. 10: Panda performance evaluation for $N = 5$ and $P_b = 0.5\text{mW}$: Experimental and analytical discovery rates under varying (a) transmission power (P_t) and b) discovery message length (M).

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 stochastic, and therefore, it is expected that the energy stored will vary over time. In Fig. 9(a), we show the capacitor voltage over time for a node with $N = 5$ and $P_b = 0.5\text{mW}$. 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.

Furthermore, in Fig. 9(b), we experiment with varying capacitor sizes ranging from 10-50mF. As expected, smaller capacitors have added variation in the voltage level. Therefore, smaller capacitors can reach the upper (fully charged) or lower (empty) voltage limits more frequently than larger capacitors. In practice, the capacitor should be sized with respect to the variation in the power consumption and power harvested.

F. Panda Design Considerations

We now consider Panda's performance for varying transmit power and discovery message durations.

1) Transmit Power, P_t: The transmission power can be set in software. A larger transmission power can result in more geographical coverage, but also consumes more energy. In Fig. 10(a), we consider $N = 5$ and $P_b = 0.5\text{mW}$ and observe how the discovery rate changes with varying transmission powers. A larger transmission power requires nodes to sleep longer before transmitting, resulting in less discoveries. Note that for this experiment the energy costs from Table IV no longer hold and we remeasured them to compute the configuration parameters.

2) Discovery Message Duration, M: The discovery message requires $M\text{ms}$ to be transmitted and contains the node ID and neighbor table information. By adjusting the modulation/coding of the radio or the data content, the packet length can be shortened. A shorter packet length results in less time transmitting as well as less time listening for messages. As shown in Fig. 10(b), smaller packet sizes result in an increase in the discovery rate. This presents an application design decision if the contents of the packet can be adjusted to obtain a desired discovery rate.

G. Panda-Dynamic

We now evaluate Panda-D (described in Section VI). The only input to Panda-D is the estimated power harvesting rate, $P_b = 0.15\text{mW}$, and the capacitor voltage V_{cap}. From (15), the average duration of the exponential sleep is then computed as,

$$\frac{1}{\lambda} = \frac{382.2238}{V_{\text{cap}} - 3.5857} - 2.9843 \text{ (ms)}. \tag{17}$$

Thus, the node scales its power consumption based on V_{cap}. For example, at $V_{\text{cap}} = 3.6\text{V}$ 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 (17).

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. 6. Specifically, we consider a network of $N = 3$ nodes in close proximity with a power harvesting rate of $P_b = 0.15\text{mW}$.

As shown in Fig. 11(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 ≈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. 11(b).

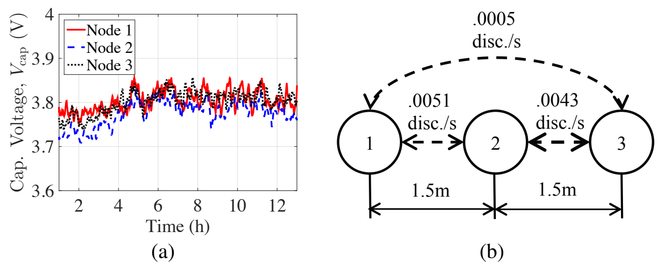
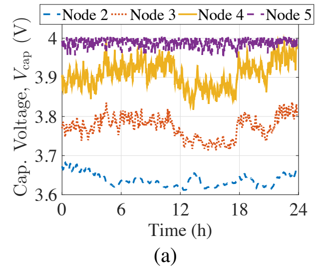


Fig. 11: Panda-D experimental evaluation for $N = 3$ and homogeneous power harvesting $P_b = 0.15\text{mW}$: (a) Capacitor voltage, V_{cap}, in a clique topology. (b) Per link experimental discovery rates for a line topology after 50 hours.



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

(b)

Fig. 12: 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.

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 $P_b = 0.15\text{mW}$ (as described in Section VII-B). After 50 hours, the resulting discovery rate is shown on each link in Fig. 11(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 $P_b = 0.15\text{mW}$. 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 $P_b = 0.15\text{mW}$ and $N = 5$.

For each of the 4 Panda-D nodes, the capacitor voltage, V_{cap}, is shown in Fig. 12(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. 12(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 Appendix E, we treat each link with non-homogenous power 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 [12] and BD-E [14] 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. Additionally, we will consider alternate formulations, for example, a scheme that rotates from sleep to extremely short listen followed by a long preamble message before transmitting the discovery message; such a scheme leverages cases where listening to the channel is very costly relative to transmitting (e.g., [4]). Also, we will consider optimizing Panda for finite energy storage sizes.

Finally, we will transform Panda into an aggregate-throughput maximizing MAC layer protocol. Panda is a natural choice at the MAC layer for applications requiring information dissemination in infrastructure-less environment (i.e., gossip-style routing at the network layer [33], and data aggregation at the transport layer, such as compressive sensing [34]), as it already maximizes the neighbor discovery rate, which can be transformed into a communication rate.

References

[1] R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, "Panda: Neighbor discovery on a power harvesting budget," in *Proc. IEEE INFOCOM'16*, Apr. 2016.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman, "Energy harvesting active networked tags (EnHANTs) for ubiquitous object networking," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 18–25, 2010.

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

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] "EnOcean Energy Harvesting Wireless Sensor Solutions and Networks," www.enocean.com/.

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

[22] 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.

[23] 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.

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

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

[26] "SANYO amorphous solar cell AM-1815," farnell.com/datasheets/87124.pdf, 2008.

[27] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1180–1189, Mar. 2012.

[28] "Texas Instruments MSP430 mixed signal microcontroller," ti.com/lit/ds/symlink/msp430f2274.pdf, 2006.

[29] "Texas Instruments low-cost low-power 2.4 GHz RF transceiver (SWRS 040C)," www.ti.com/lit/ds/swrs040c/swrs040c.pdf, 2015.

[30] 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.

[31] B. Dezfouli, M. Radi, S. A. Razak, K. Whitehouse, K. A. Bakar, and T. Hwee-Pink, "Improving broadcast reliability for neighbor discovery, link estimation and collection tree construction in wireless sensor networks." *Computer Networks*, vol. 62, pp. 101–121, Apr. 2014.

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

[33] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *IEEE/ACM Trans. Netw.*, vol. 14, no. 3, pp. 479–491, 2006.

[34] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 37–59, 2012.

TABLE VII: Panda idle power consumption for every input (N, P_b) pair from Table V: the probability of a node waking in the middle of a transmitted packet and the expected portion of the power budget (P_b) consumed.

N	P_b (mW)	Pr{A node wakes up in the middle of a packet)}	Expected Energy per Renewal (µJ)	Pct of P_b (%)
3	0.15	0.34e-3	0.0302	0.034
	0.3	0.69e-3	0.0605	0.068
	0.5	1.15e-3	0.1010	0.112
5	0.15	0.41e-3	0.0363	0.068
	0.3	0.83e-3	0.0728	0.135
	0.5	1.38e-3	0.1215	0.223
10	0.15	0.47e-3	0.0410	0.151
	0.3	0.94e-3	0.0822	0.300
	0.5	1.57e-3	0.1376	0.495

Appendix A: Incorporating Idle Power Consumption

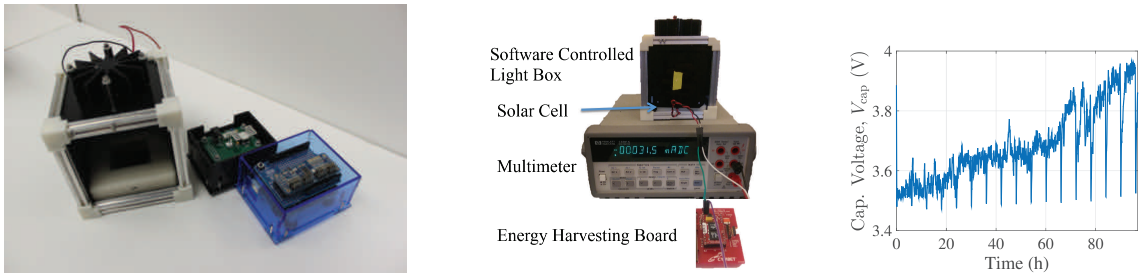
In this work, we disregard the power cost of nodes in the sleep state. In this section, we explain how these costs can be incorporated. As described in Section VII, the idle cost of the microcontroller is normally 1.6µW. This draw is constant for all states (sleep, listen, and idle). As such, to incorporate it into our model, it is simply subtracted from the power budget P_b.

In Section IV-D, we ignore the expected amount of energy (µJ) consumed by a node when it begins to listen while a packet is currently being transmitted (exemplified by Node 5 in Fig. 4). In this case, the node spends energy to transition to and from the sleep state, as well as listen for a short fixed Clear Channel Assessment (CCA) period, denoted as t_{CCA}. The energy consumption is then $C_{sl} + P_r \cdot t_{\text{CCA}} + C_{ls}$. This event occurs in a renewal with probability given by $\frac{N-1}{N}(\mathrm{e}^{-\lambda l})(1-\mathrm{e}^{-\lambda M})$. Firstly, the node must not be the transmitter w.p. $\frac{N-1}{N}$. As the node is asleep when the transmitter begins to listen, it must then sleep for at least l duration. Finally, given that it is in the sleep state when the transmitter begins to transmit, it must then wakeup before the message is transmitted (duration M).

For the evaluations in Section VII, the idle power costs are summarized in Table VII for the experimental parameters originally presented in Table V. As can be seen, this probability is quite small (under 0.2%). Thereby, the percentage of the power budget consumed, on average, is always less than 0.5% of the power budget, and therefore can be ignored. We note, however, that the PCA can easily be modified to incorporate this idle power consumption.

Appendix B: Importance of Switching Costs

In this work, we incorporate the costs to switch to and from different radio states (sleep, receive, transmit). In Table VIII, we demonstrate the importance of accounting for these costs (which are commonly overlooked in related work). In the table, the PCA is used to compute the parameters under the assumption that $C_{ij} = 0\,(\forall i, j \in \{s, r, t\})$. As indicated in the table, the discovery rate improves by 2-3x compared to



(a) (b) (c)

Fig. 13: (a) Software controlled light system including a dark box enclosure, high-power LED driver, and an Arduino-based light controller. (b) Energy harvesting characterization measurement setup. (c) Capacitor voltage while adjusting the light levels over 4 days to find the energy neutral point.

TABLE VIII: Panda performance evaluation: the discovery rate U_A resulting from the PCA, and the actual power consumed when ignoring the switching costs.

N	P_b (mW)	U_A, w/ Switching Costs (disc./s)	U_A, w/o Switching Costs (disc./s)	Power Consumed (mW)
3	0.15	0.0039	0.010	0.26
	0.3	0.0156	0.038	0.52
	0.5	0.0434	0.107	0.86
5	0.15	0.0130	0.032	0.26
	0.3	0.0519	0.128	0.52
	0.5	0.1440	0.359	0.87
10	0.15	0.0584	0.144	0.26
	0.3	0.2330	0.581	0.52
	0.5	0.6470	1.630	0.87

the discovery rate when including the switching costs from Table IV. However, the power consumed by transition causes the power budget to be exceeded by up to 80%. Therefore, ignoring the switching costs may improve the discovery rate, but also results in significantly higher power consumption.

Appendix C: Software Controlled Light System and Harvesting Inefficiencies

We develop an advanced software controlled light system (shown in Fig. 13(a)) that uses a Java-based script and Arduino-based light control modules to precisely control the irradiance (light energy intensity) generated by LEDs. The system produces 1024 irradiance levels from 0–14$\mathrm{mW/cm^2}$ and the level can be changed every 100ms. Dark box enclosures and 3D printed mounting fixtures ensure full control over the light conditions at the solar cells. This guarantees that our experimental evaluations *are based on the same energy inputs*.

Furthermore, we conduct extensive experiments utilizing a UV818 photodetector to carefully calibrate the irradiance of the light control system. We characterize the power harvested under both the ambient light and the software controlled light setup. The measurement setup is shown in Fig. 13(b). The solar cell is connected to an ammeter, and together with the cell voltage, the harvested power is easily computed. Using the software controlled light system, we record the mapping of irradiance to power harvested and can control the power harvested by the solar cell.

However, the actual power that is stored depends on numerous inefficiencies in the power harvesting circuitry. Specifically, the Cymbet CBC5300 up-converts the power harvested from the solar cell voltage to the capacitor charging voltage of 4V, which consumes some overhead energy. In addition, there are inefficiencies in the regulation circuit which regulates an output voltage of 3.5V to power the load. These inefficiencies are difficult to characterize as they vary based on uncontrollable external factors such as the temperature and component variations.

In our evaluations (see Section VII), nodes were given light levels which corresponded to their power budget P_b. To accomplish this, given the inefficiencies described above, we conduct a 4-day experiment in which nodes operated using Panda, yet we varied the light levels every 6 hours. An example of the capacitor voltage for one node in this experiment is shown in Fig. 13(c). Each valley represents a 10-minute "dark" period where the light is completely off before changing to the next light levels.

With limited light levels (i.e., hours 0-20 in Fig. 13(c)), the capacitor voltage operates near the minimum implying that the node is consuming more energy than it harvests. With larger light levels (i.e., hours 80-100), the node is harvesting more energy than it consumes and thus the capacitor voltage reaches its upper limit. However for the range of lights corresponding to 20-80 hours in Fig. 13(c), the node has a relatively stable voltage, implying that it is consuming power (on average) as the same rate it harvests; *energy neutrality* is obtained.

By performing this experiment for all nodes, we found the light levels at which each node is energy neutral. The neutral light levels varied significantly. Furthermore, by comparing the power harvested by the solar cell to the power budget (P_b), we found that the efficiency of the storage process to be between 40% and 60%. This emphasizes the need to incorporate energy storage feedback into the ND protocol, as is done by Panda-D.

Appendix D: Searchlight-E and BD-E

In this section, we describe the Searchlight [12] and Birthday [14] protocols, as well as how they are adapted to obey an power budget, termed Searchlight-E and BD-E. Both protocols divide time into slots, and in each time slot, a node must decide whether to be *active* or *idle*.[11] The percentage of active slots is termed the *duty cycle*. During active slots, nodes send a packet transmission, then listen to the channel for messages from neighbors, and end the slot with another packet transmission [15]. Thus, when the active slots of any two nodes

[11]In its original form, the Birthday protocol further divided active slots into a listen slot and a transmit slot. However, since Birthday was published, Disco [15] demonstrated the *beacon slot* which combines the listen and transmit slots. Thus, for simplicity, we adapt Birthday to use beacons.

TABLE IX: Panda-D Discovery Rate Approximation for the $N = 5$ experiment from Fig. 12: Error rate (%) of the experimental per-link discovery rate compared to (18).

	1	2	3	4	5
1	-	-17	8	-15	-9
2	-32	-	-22	-23	-20
3	12	12	-	-4	23
4	-17	-20	-17	-	12
5	-11	13	16	11	-

are un-aligned and only partially overlap, a mutual discovery occurs. Within this model, the ND protocol designates each slot as active or idle such that they the node maintains the duty cycle constraint.

In Birthday, each node randomly decides whether to be active in a time slot with probability p. Clearly, p is the effective duty cycle. In Searchlight, two slots are active per cycle of t slots. The node is always active in the first slot of cycle, which is referred to as the anchor slot. Clearly if the anchor slot of two nodes overlap, then they will continue to discover each other in each cycle. Since nodes are unsynchronized, it is likely that the anchor slots of the two nodes will not overlap and, by construction, the offset of the two nodes must be less than $t/2$ slots. Thus, each node activates a second time in a cycle in what is called a *probe* slot. The probe slot begins in slot 2 of the cycle is successively incremented until slot $t/2$ at which point it is guaranteed to have discovered the neighboring node.

We now analyze the power consumption for each protocol. We will denote the time slot duration as d_s and the energy consumed per active slot is,

$$E_{\text{slot}} = \left(2P_tM + P_r\left(d_s - 2M\right) + C_{st} + C_{ts}\right).$$

The power budget for Searchlight results in two active slots per cycle of length t and is written as, $2E_{\text{slot}}/(td_s) \leq P_b$. In Birthday, each node transmits a beacon in a slot w.p. p and the power budget is simply, $pE_{\text{slot}}/d_s \leq P_b$. In our evaluation, we select t and p such that the power budget is fully consumed and term these protocols, Searchlight-E and BD-E.

We note that there are numerous aspects of related protocols [12]–[16] which have not been considered. Specifically, existing works do not consider collisions occurring due to not listening before transmitting. Furthermore, numerous practical parameters are not considered such as the setting of the slot size. As can be seen above, the slot size impacts the average power consumption. In our simulation of Searchlight-E and BD-E, we ignore collisions and set the slot size to $d_s = 50\text{ms}$ with a guard time of 1ms, as was done in [13].

Appendix E: Approximate Analysis of Panda-D

We now approximate the directional discovery rate U_{ij} (i.e., the rate at which node i is discovered by node j) for a link between node i and j, under non-homogeneous power budgets λ_i and λ_j, respectively. Using similar analysis as in Section IV, we obtain

$$U_{ij} = \frac{\lambda_i}{\lambda_i + \lambda_j}(1 - e^{-\lambda_j l})/(\frac{1}{\lambda_i + \lambda_j} + l + M). \qquad (18)$$

To evaluate this approximation, we apply it to the non-homogenous power harvesting experiment described in Section VII-G(iii), with the total number of discoveries on each directional link presented in Fig. 12(b).

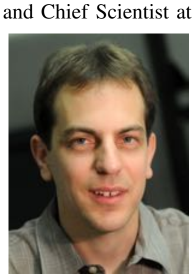
Recall that the sleep rate for a node i, λ_i, is dynamically changing in Panda-D. We estimate the sleep rate based on the experimental average capacitor voltage using (15). In Table IX, we compute the error rate between the experiment per-link discovery rate and (18). The approximation is quite crude (typically within 25%). Yet it can still be used as a rough approximation of the per-link discovery rate. We remark here that the relatively high errors come from: (1) the small number of discoveries, (2) each node is operating independently without knowledge of N a-priori, and (3) errors in the ADC capacitor voltage sampling.

Robert Margolies (S'13-M'16) is a Senior Inventive Scientist at AT&T Research Labs in Bedminster, NJ. He received the B.S. degree (Summa Cum Laude) from Rensselaer Polytechnic Institute in 2010 and the M.S. and Ph.D. degrees from Columbia University in 2011 and 2015, all in Electrical Engineering. His research interests are in cross-layer PHY and MAC layer design for ultra-low-power networks, networks of energy harvesting nodes, and opportunistic cellular scheduling. Robert has received a number of awards including the NSF Graduate Research Fellowship.

Guy Grebla is a senior software engineer at Google, New York, NY. He received the B.A., M.A., and Ph.D. degrees in Computer Science from the Technion—Israel Institute of Technology, completing his Ph.D. studies in 2013. He was a postdoctoral research scientist in the Electrical Engineering department at Columbia University, New York, NY (2013-2015). He is an IEEE Senior Member and serves as a Technical Program Committee (TPC) member of IEEE INFOCOM'16, IEEE INFOCOM'17, and ACM MOBIHOC'15.

Tingjun Chen (S'16) received the B.Eng. degree in Electronic Engineering from Tsinghua University, Beijing, China, in 2014. He received the M.S. degree in Electrical Engineering from Columbia University, New York, NY, USA, in 2015, and is currently a Ph.D. student in Electrical Engineering at Columbia University. His research interests are in algorithms, optimization, MAC layer design, and system implementation in wireless networks, energy harvesting networks, and full-duplex networks. He received the Wei Family Private Foundation Fellowship and the Columbia University Electrical Engineering Armstrong Memorial Award.

Dan Rubenstein Dan Rubenstein is an Associate Professor in the Department of Computer Science at Columbia University. He received a PhD in computer science from University of Massachusetts, Amherst. His research interests are in network technologies, applications, and performance analysis. He is an editor for IEEE/ACM Transactions on Networking, is program chair of ACM Sigmetrics 2011, and has received an NSF CAREER Award, IBM Faculty Award, the three paper awards from ACM SIGMETRICS 2000, IEEE ICNP 2003, and ACM CoNext 2008. He was a visiting scientist at Google for the 2011 year, and Chief Scientist at Silver Lining Systems for 2012.



Gil Zussman (S'02-M'05-SM'07) received the Ph.D. degree in electrical engineering from the Technion in 2004 and was a postdoctoral associate at MIT in 2004–2007. He is currently an Associate Professor of Electrical Engineering at Columbia University. He is a corecipient of 6 paper awards including the ACM SIGMETRICS'06 Best Paper Award and the 2011 IEEE Communications Society Award for Outstanding Paper on New Communication Topics. He received the Fulbright Fellowship, the DTRA Young Investigator Award, and the NSF CAREER Award, and was a member of a team that won first place in the 2009 Vodafone Foundation Wireless Innovation Project competition.