Bootstrapping Battery-free Wireless Networks: Efficient Neighbor Discovery and Synchronization in the Face of Intermittency

Kai Geissdoerfer
TU Dresden

Marco Zimmerling
TU Dresden

Abstract

Due to their favorable size, cost, and sustainability, battery-free devices are preferable in various applications. However, battery-free devices operate only intermittently since ambient energy sources, such as light and radio-frequency signals, are often too weak to continuously power the devices. This paper addresses the unsolved problem of efficient device-to-device communication in the face of intermittency. We present Find, the first neighbor discovery protocol for battery-free wireless networks that uses randomized waiting to minimize discovery latency. We also introduce Flync, a new hardware/software solution that synchronizes indoor light harvesting nodes to powerline-induced brightness variations of widely used lamps, which we exploit to further speed up neighbor discovery. Experiments with an open-source prototype built from off-the-shelf hardware components show that our techniques reduce the discovery latency by 4.3× (median) and 34.4× (99th percentile) compared with a baseline approach without waiting.

1 Introduction

Despite technological advances, the maintenance costs and environmental impact of batteries remain a major threat to the vision of a truly ubiquitous Internet of Things [3, 11]. *Battery-free devices* that store energy harvested from light, vibrations, radio-frequency (RF) signals, and other ambient sources in a capacitor are one of the most viable alternatives today [45]. Capacitors store electrical energy in an electrical field rather than in the form of chemical energy, and thus have negligible aging effects and are sustainable [1, 6]. Moreover, their favorable size, weight, and cost points enable new applications where batteries would be inconvenient or infeasible [30].

Challenge. The power that can be harvested from ambient energy sources can vary significantly across time and space [15], and is often too weak to directly power a battery-free node, such as a smart sensor [32]. Thus, as illustrated in Fig. 1 and further discussed in detail in Sec. 7, a battery-free device first needs to buffer sufficient energy in its capacitor before it can operate for a short period of time; then the device turns off until the capacitor is sufficiently charged again. As a result, battery-free devices operate *intermittently*.

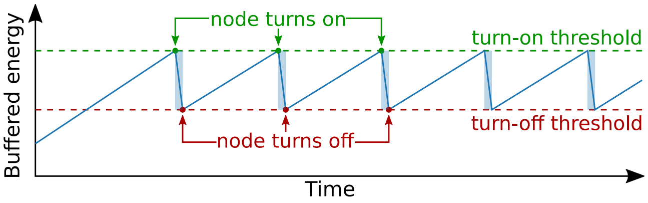


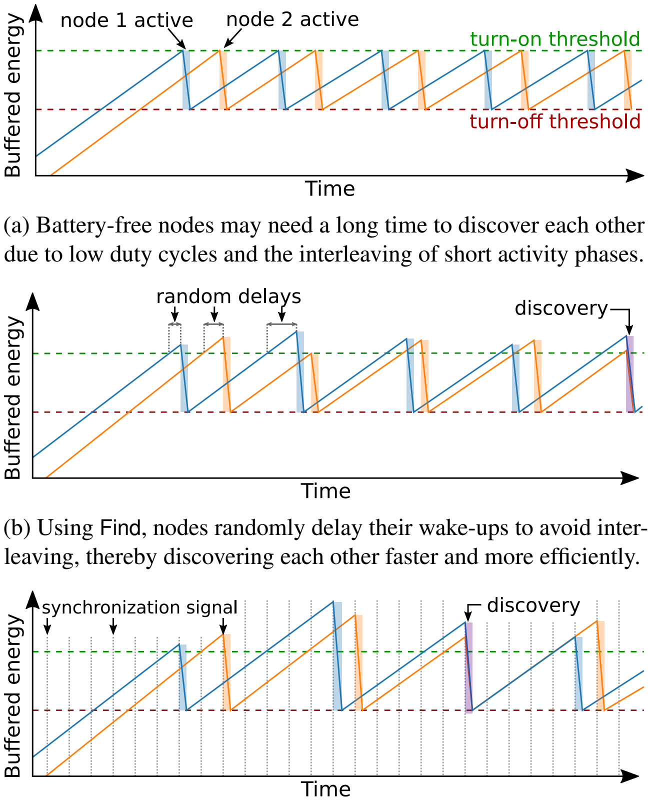
Figure 1: Because ambient power is often weak, a battery-free node must buffer energy before it can wake up and operate for a short time period. This is known as intermittent operation.

Intermittency is in stark contrast to conventional duty cycling. While duty cycling is intentionally introduced to save energy and thus predictable, intermittency is mainly dictated by uncontrollable environmental factors and thus impacts the device operation in unpredictable ways. The resulting challenges in terms of, for example, reliable time keeping [12, 18] or ensuring application progress and data consistency [8, 34] have been widely studied in the recent literature.

The impact of intermittency on wireless networking has instead received little attention. Just like in conventional battery-supported networks, direct communication between battery-free devices is desirable, for example, to increase the availability of the system [36], to enable novel applications [20, 32], and to reduce infrastructure costs [35]. However, to communicate with one another, sender and receiver must be active simultaneously for at least the airtime of one complete packet. This is challenging in battery-free networks for three reasons:

1. Battery-free nodes can only become active when they have accumulated sufficient energy in their capacitors.
2. They may only be active for a short period, which renders excessive sampling of the wireless channel infeasible.
3. Their duty cycles are often low and may change unpredictably due to varying availability of ambient energy.

For example, our prototype battery-free node needs to charge its capacitor for hundreds of milliseconds to sustain 1 ms of activity when harvesting from indoor light. Because the short



(a) Battery-free nodes may need a long time to discover each other due to low duty cycles and the interleaving of short activity phases.

(b) Using Find, nodes randomly delay their wake-ups to avoid interleaving, thereby discovering each other faster and more efficiently.

(c) Using Find + Flync, nodes implicitly align their wake-ups to an external synchronization signal, further accelerating discovery.

Figure 2: Illustration of the battery-free neighbor discovery challenge in (a) and of our proposed mechanisms to address it in (b) and (c).

activity phases of different nodes are generally interleaved, as shown in Fig. 2a, it takes a long time until nodes encounter each other. And this is not a one-time endeavor: While nodes may attempt to synchronize their activity phases at the first encounter, they lose track of time during extended periods without energy [12, 18], which forces them to re-synchronize.

This challenge is fundamental and pertains to battery-free networks regardless of the type of wireless communication: While backscatter communication can lower the energy costs compared to active radio communication, sender and receiver still need to have sufficient energy at the same time. Prior work on backscatter has primarily focused on pushing the envelope of communication range and throughput, avoiding intermittency by evaluating the designs under high ambient energy availability [20, 32] or by powering the devices via USB or batteries to not disturb the measurements [35]. To our knowledge, direct radio communication between real battery-free devices has not been explored so far, as the overhead due to intermittency is considered too demanding [36].

Contribution. We set out to bootstrap battery-free wireless networks by presenting two mechanisms that enable battery-free nodes to discover each other quickly and efficiently.

The first mechanism, Find, is a neighbor discovery protocol. As illustrated in Fig. 2b, the key idea behind Find is to address the interleaving problem by introducing random delays after the devices have sufficiently charged their capacitors before becoming active. We develop analytical models to determine an optimized delay distribution that minimizes discovery latency. At runtime, each Find node dynamically adapts the delay distribution to changes in its energy availability.

The second mechanism, Flync, is a hardware/software solution that further speeds up the discovery process. Flync phase-synchronizes solar energy harvesting devices to powerline-induced flicker of state-of-the-art lamps; the proposed circuit draws only 5 µW of power. As shown in Fig. 2c, using Find together with Flync, nodes can implicitly align their activity phases to this external synchronization signal, dramatically increasing their chances to be active at the same time.

We prototype our mechanisms on a custom-designed ultra low-power battery-free node. It is based on a state-of-the-art microcontroller (MCU) with a 2.4 GHz Bluetooth Low Energy (BLE) radio, and buffers energy harvested via three small solar panels in a tiny 47 µF ceramic capacitor.

We use 6 of our prototype battery-free nodes to conduct extensive experiments and a contact-tracing case study. We summarize our key findings as follows:

- Find provides shorter discovery latencies than greedy and naïve random node activations. Find + Flync improves on greedy by 4.3× in terms of the median latency (141 s vs. 604 s); the 99th percentile improvement is 34.4×.
- Our hardware prototype works with 14 out of 19 fluorescent, halogen, and light emitting diode (LED) lamps we tested, demonstrating that Flync is broadly applicable in indoor environments. Flync provides a stable clock signal when nodes are deployed across different rooms, carried around, or exposed to temporary shadowing.
- We conduct a contact-tracing case study in an open-air pub with Find and in an office kitchen using Find + Flync. The median time between consecutive encounters of the same two nodes is 1.5 s and 7.5 s in the outdoor and indoor environment, respectively. This shows the potential of our battery-free designs for real-world applications.

Overall, this paper makes the following contributions:

- Find, the first neighbor discovery protocol for battery-free networks. Find is agnostic to the energy harvesting modality and the type of wireless communication.
- Flync, the first solution extracting a stable clock from solar harvesting current, whose amplitude changes due to powerline-induced flicker of state-of-the-art lamps. While we use Flync in tandem with Find to speed up discovery in indoor scenarios, Flync is useful for other purposes and also applicable to battery-supported nodes.
- A novel battery-free node design including an implementation of an efficient intermittent runtime.
- Empirical evidence that the proposed techniques work well under a diverse set of real-world conditions.

2 Battery-free Neighbor Discovery

This section presents the design of Find, the first neighbor discovery protocol for battery-free wireless networks. Find empowers battery-free nodes to quickly discover each other's presence despite intermittent operation and varying ambient energy availability. It is agnostic as to how the nodes harvest energy (from solar, vibrations, RF, etc.) and as to whether they communicate using backscatter or radio communication.

The design of Find is based on the observation that the only way battery-free nodes can reliably avoid interleaving is to not wake up and become active immediately after reaching the minimum energy level required to do so. We refer to this as the greedy approach. Instead, Find delays each wake-up for a random time. A crucial question is how to choose this random delay to ensure fast and energy-efficient discovery.

To answer this question, we devise a model that captures the impact of key parameters, such as the charging time needed to reach the minimum energy level and the random delay, on the discovery latency (Sec. 2.1). Using this model, we then determine an optimized delay distribution that minimizes the discovery latency (Sec. 2.2). Finally, we describe how these considerations materialize in the practical design of the Find protocol and its runtime operation (Sec. 2.3).

2.1 Modeling Discovery Latency

Suppose that a node needs to charge for c slots until it reaches the minimum energy level required to be active for one slot. Let k_0 denote the first slot in which a node reaches the minimum energy level. Using Find, a node waits for a random delay x in units of slots before it wakes up and becomes active. We model x as a discrete random variable X with probability mass function (pmf) $p_X(x)$. During an active slot, a node fully depletes its energy storage. The probability that a node becomes active for the first time in slot k is given by

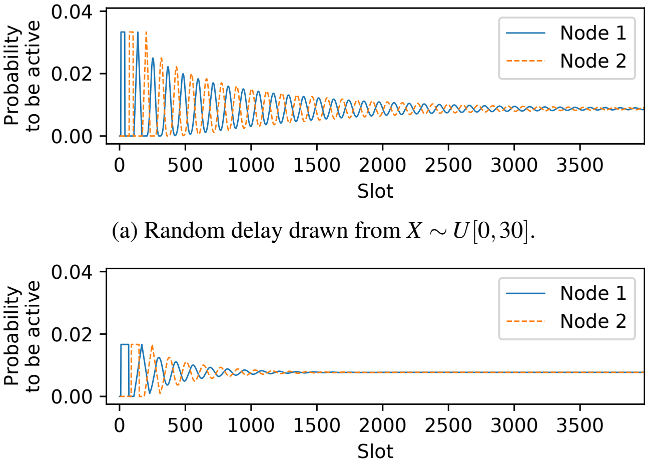
$$p_{wk,0}(k) = p_X(k - k_0) \tag{1}$$

Afterward, a node needs to recharge for c slots before it can become active again. The time of the second wake-up is the sum of the time of the first wake-up, the charging time, and the second random delay. The same reasoning applies recursively to all future wake-up times. Because the random delay is independently chosen across all wake-ups, we can use a recursive convolution to determine the probability that a node wakes up for the n-th time in the k-th slot

$$p_{wk,n}(k) = (p_{wk,n-1} * p_X)(k - c) \tag{2}$$

By summing over $n \to \infty$ we obtain the probability that a node is active in slot k

$$p_a(k) = \sum_{n=0}^{\infty} p_{wk,n}(k) \tag{3}$$



(a) Random delay drawn from $X \sim U[0,30]$.

(b) Random delay drawn from $X \sim U[0,60]$.

Figure 3: Probability of being active in a slot for two nodes with identical charging times but an initial offset in their wake-ups. The more wide-spread the random delay, the faster nodes break up their interleaved wake-up pattern at the cost of a lower average duty cycle.

To model discovery latency, we consider a fully connected network of N nodes (i.e., a clique of size N). Using a suitable sequence of message exchanges in active slots (see Sec. 2.3), one of the $M = N(N-1)/2$ bi-directional links $i \leftrightarrow j$ is discovered if nodes i and j are active in the same slot while all other nodes in the network are inactive. Otherwise, a collision occurs and no link is discovered, a typical assumption in neighbor discovery protocols [24]. The probability that link $i \leftrightarrow j$ is discovered within k slots is the complement of the probability that the link is not discovered in slots $0, \ldots, k$:

$$c_{i\leftrightarrow j}(k) = 1 - \prod_{\kappa=0}^{k} \left(1 - p_{a,i}(\kappa) \cdot p_{a,j}(\kappa) \cdot \prod_{l \neq i,j} (1 - p_{a,l}(\kappa)) \right) \tag{4}$$

$c_{i\leftrightarrow j}(k)$ can be regarded as the cumulative distribution function (cdf) of the discrete random variable describing the slot in which link $i \leftrightarrow j$ is discovered. With $p_{i\leftrightarrow j}(k)$ denoting the corresponding pmf, we compute the expected fraction of links discovered up to slot k by averaging $p_{i\leftrightarrow j}(k)$ over all M links

$$d(k) = \frac{1}{M} \sum_{i\leftrightarrow j} p_{i\leftrightarrow j}(k) \tag{5}$$

If the nodes' charging times are finite, $d(k)$ is a valid cdf, and we define the discovery latency as

$$T_{nd} = \sum_{k=0}^{\infty} (1 - d(k)) \tag{6}$$

2.2 Optimized Delay Distribution

With the above model we are able to get a better understanding of how nodes should delay their wake-ups to help discovery.

Example. Suppose two nodes i and j with the same charging time of $c = 100$ slots, but different slots k_0 in which they reach

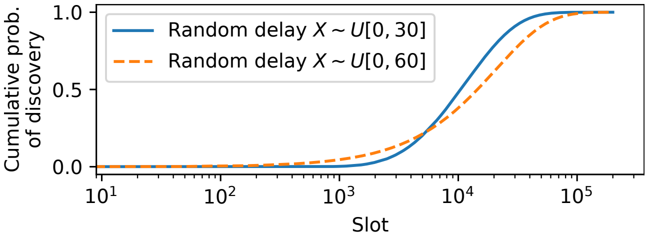


Figure 4: Cumulative distribution function of the slot in which two nodes discover each other, for the two delay distributions in Fig. 3. A more wide-spread delay performs better initially, but leads to lower performance in the long run due to a lower average duty cycle.

the minimum energy level for the first time (i.e., initial offset). Using (3) we plot in Fig. 3a for both nodes the probability of being active in a slot when they pick random delays from the discrete uniform distribution $X \sim U[0,30]$. We see that in the first thousand slots there is hardly any overlap in the activity of the nodes: Due to the initial offset, node i is likely active when node j is powered off, and vice versa. The probability of being active smears out over time and converges to an average duty cycle of $1/(c+\mathrm{E}[X]) \approx 0.0087$. Fig. 3b plots the same when the two nodes pick random delays from $X \sim U[0,60]$. Compared to Fig. 3a we find that the probability of being active smears out sooner as nodes tend to choose more wide-spread delays. However, as nodes also tend to pick longer delays, they have a lower average duty cycle of 0.0077.

Fig. 4 directly compares the two delay distributions by plotting the cdf of the slot in which nodes i and j discover each other according to (4). We observe that the more wide-spread delay induced by the second distribution $X \sim U[0,60]$ initially provides a higher probability of discovery. In the long run, however, the higher average duty cycle of the first distribution $X \sim U[0,30]$ leads to a higher probability of discovery.

Choosing a distribution. The above example suggests that a non-negative delay distribution with *high randomness and low mean* is preferable. Entropy is a commonly used measure of randomness. Maximizing the entropy of a general non-negative distribution with a given mean yields the exponential distribution [38]. Thus, in Find, we draw random delays from the geometric distribution, the discrete analogue of the exponential distribution, with scale parameter $1/r$ and pmf $(1-r)^k r$ for $k \in \{0,1,2,\ldots\}$.

To confirm our reasoning, we compare the geometric distribution against other well-known distributions, namely the discrete uniform distribution and the Poisson distribution. We sweep the scale parameter of the three distributions and compute the discovery latency using (6) for the two-node case, where nodes i and j have equal charging times (25, 100, 500, or 1000 slots). We find that the geometric distribution achieves the lowest discovery latency across all charging times. Fig. 5 shows the resulting curves for a charging time of 100 slots. The differences in the minimum discovery latencies are relatively small. One reason for this is that, according to the central limit theorem, the probability that a node wakes up for the n-th time in slot k converges to a normal distribution for large n, irrespective of the underlying delay distribution.

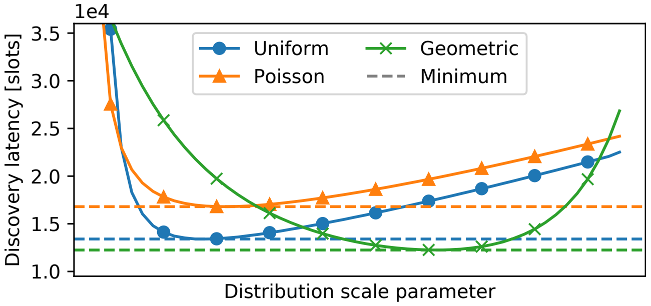


Figure 5: Discovery latency against scale parameter for three different probability distributions. The geometric distribution performs best as it yields delays with high randomness and low mean.

Determining optimized distribution parameters. Having chosen a suitable delay distribution, we now turn to the problem of determining the scale parameter that minimizes the discovery latency. To formally state the optimization problem, we consider the worst case in terms of discovery latency: all N nodes have the same charging time c, and their initial wake-up times $k_{0,i}$ are all interleaved as in Fig. 3, that is,

$$k_{0,i} = i \cdot \frac{c + 2\,\mathrm{E}[X]}{N} \tag{7}$$

where i is the node index and $\mathrm{E}[X]$ is the expected delay. For specific N and c, we minimize the discovery latency given by (6) and the initial offsets given by (7)

$$\min_{r} T_{nd}(N,c) \tag{8}$$

Numerical evaluation suggests that $T_{nd}(N,c)$ is convex (see Fig. 5) and hence straightforward to optimize. We use Brent's method [9] to approximate the scale parameter $1/r^*$ that minimizes the discovery latency. The next section explains how we adapt the scale parameter at runtime on a real node.

2.3 Practical Protocol Design

The above analysis makes a number of simplifying assumptions that do not hold in practice. For example, the charging times are generally different across nodes and vary over time. A node typically only knows its own charging time c and is unaware of the total number N of nodes in the network.

Nevertheless, prior work has shown that neighboring nodes have similar energy availability because they harvest energy from the same ambient source(s) [4, 15]. Thus, *in the absence of any prior information*, a reasonable approach for a node is to assume that its neighbors harvest the same amount of energy and thus have the same charging time c like itself.

Moreover, we found that knowledge of the number of nodes N is often not required: optimizing for the case of a two-node

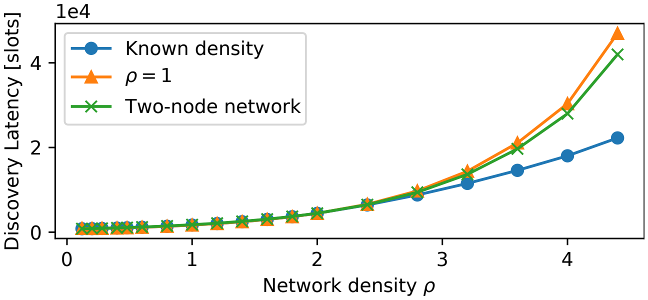


Figure 6: Discovery latency against network density ρ when optimizing for the known density, for a fixed density of ρ = 1, and for a two-node network. For ρ ≤ 2.5, all approaches perform similarly.

network yields competitive performance across a wide range of network densities. In other words, in practice it is often sufficient for a node to assume that it is has only one neighbor (although over time it may discover that it has many more). To understand why, we plot in Fig. 6 the discovery latency for a charging time of 25 slots when optimizing for (*i*) the known network density $\rho = N/c$, (*ii*) a fixed network density of $\rho = 1$, and (*iii*) a two-node network. We can see that for a network density of $\rho \leq 2.5$ the three approaches achieve almost the same performance. For realistic charging times, the network density rarely exceeds this threshold. For example, based on the charging times and beacon length in our real-world case study (see Sec. 6), a network density of $\rho = 2$ would require a network of around 4000 fully connected nodes.

Runtime operation. Prior to each wake-up, a Find node samples a geometric distribution to determine the random delay. A node dynamically adapts the scale parameter of the distribution to changes in its charging time, under the assumption that it has one neighbor with the same charging time, as explained above. To achieve an efficient runtime operation, we store a look-up table of optimized scale parameters in non-volatile memory and use inverse transform sampling to convert samples from a uniform pseudo-random number generator to the optimized, geometric distribution.

Frame structure. Taking inspiration from existing neighbor discovery protocols for battery-powered sensor nodes [5, 13], we adopt the frame structure shown in Fig. 7. During each active slot, a node first transmits a beacon, then listens for potential beacons from neighboring nodes, and finally transmits another beacon at the end of the slot. The second beacon ensures that nodes can discover bi-directional links in one common active slot. Specifically, if the slot offset $\mathcal{T}$ between two nodes (see Fig. 7) is uniformly distributed between $-T_{slot}/2$ and $T_{slot}/2$, where T_{slot} is the slot length, the probability that two nodes successfully discover each other's presence is

$$p = 1 - \frac{2 \cdot (T_{ta} + T_{tx})}{2 \cdot (T_{ta} + T_{tx}) + T_{rx}} \quad (9)$$

Here, as depicted in Fig. 7, T_{tx}, T_{rx}, and T_{ta} denote the times

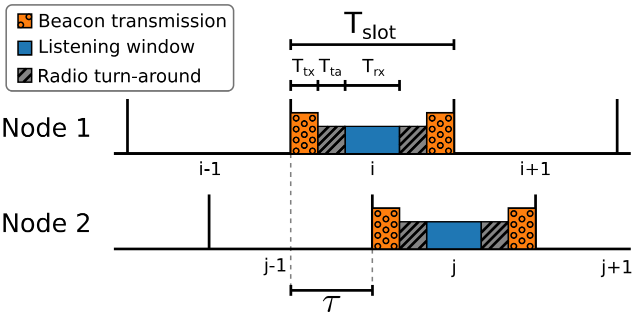


Figure 7: Find's frame structure specifying the sequence of beacon transmissions and the intermediate listening window during an active slot. Using our prototype implementation, nodes can successfully discover each other if the slot offset $\mathcal{T}$ is between 88 µs and 848 µs.

needed to transmit a beacon, to listen for potential beacons, and to switch from receive to transmit mode (or vice versa). In order to maximize the success probability according to (9), Find keeps the beacon transmission time T_{tx} as short as possible to maximize the listening window T_{rx}.

3 Further Accelerating Neighbor Discovery

Find provides fast and energy-efficient neighbor discovery in battery-free networks. Nevertheless, if the ambient energy availability is low, discovery may still take a long time due to the low duty cycles. For example, according to our model, under dim indoor light conditions it takes on average 8 min until two of our prototype battery-free nodes (see Sec. 4) discover each other. Similar observations are to be expected when nodes harvest from weak RF signals or miniature vibrations [7]. The discovery latencies in those challenging energy environments can be prohibitively long for many applications.

This section introduces an approach that facilitates, according to our model, a 10× speed-up in the above-mentioned scenario, allowing two nodes to discover each other in 45 s on average instead of 8 min at an additional cost of only 5 µW. The underlying idea is that neighboring nodes harvest energy from the same ambient source(s) and may therefore have access to a common energy signal that can be used as a time reference. In combination with Find, nodes can exploit this common time reference to align their wake-ups, thereby increasing the chances that nodes are active in the same slot.

To assess the potential of this idea, we focus in this work on harvesting energy from indoor light. While this is a popular method for powering battery-free nodes due to the ubiquity of interior lamps, the energy density of indoor light is significantly lower than that of sunlight. As such, it represents both a challenging environment for battery-free neighbor discovery and a highly relevant setting for real applications. In the following, we provide answers to three key questions:

1. What common energy signal can nodes use? (Sec. 3.1)
2. How to efficiently extract a time reference? (Sec. 3.2)
3. How to exploit this for faster discovery? (Sec. 3.3)

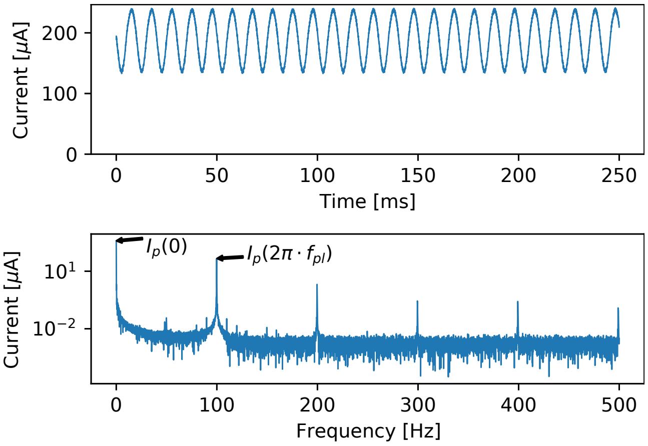


Figure 8: Time and frequency domain of solar panel current when harvesting energy from light emitted by a UP-PL30120-45W LED panel. The current varies with double the powerline frequency.

3.1 Powerline Flicker in Solar Current

When harvesting energy from indoor light, we observed that the solar panel current varies with double the powerline frequency (50 or 60 Hz depending on the region). As an example, Fig. 8 shows the solar panel current when harvesting energy from an LED panel light found in a typical office space.

Practically all indoor lamps are connected to mains power, which induces phase-synchronized brightness variations (*powerline flicker*) of the lamps through different effects. Despite their relatively high inertia, the alternating current through the filament of incandescent and halogen lamps causes temperature and, as a result, brightness variations. A similar effect occurs in gas-discharge lamps like the ubiquitous fluorescent lamps, where the alternating current through the gas modulates the brightness. Due to the exponential relation between forward voltage and brightness, voltage-controlled LEDs are also sensitive to residual ripple of the rectified supply voltage. Because the power available from a solar panel is proportional to the brightness of the incident light, it also varies with double the powerline frequency, as visible in Fig. 8.

To assess the potential of using powerline flicker as a common energy signal, we characterize the magnitude of powerline frequency induced fluctuations of the solar panel current for a wide variety of lamps. To compare lamps across diverse average brightness levels, we define the *flicker index FI* as the ratio of the amplitude of the powerline frequency component and the DC component of the solar panel current i_p

$$FI = \frac{I_p(2\pi \cdot f_{pl})}{I_p(0)} \tag{10}$$

where $I_p(\omega) = \mathcal{F}\{i_p(t)\}$ is the Fourier transform of the solar panel current and f_{pl} is the powerline frequency.

We attach an IXYS SM141K06L solar panel to a Shepherd node [15] and record 15 s of solar panel current at a sampling frequency of 100 kHz from each of the 19 lamps in Fig. 9. For each trace we compute the flicker index using (10). The results in Fig. 9 show that all lamps we tested exhibit varying levels of powerline flicker. We observe that all fluorescent and halogen lamps have a relatively large flicker index. The results for the tested LED lamps are more ambiguous. We suspect that highly integrated, bulb-shaped LED lamps tend to have high-quality current-controlled drivers with little flicker, whereas commercial panel-style LED lamps often rely on voltage-controlled drivers with significant levels of flicker.

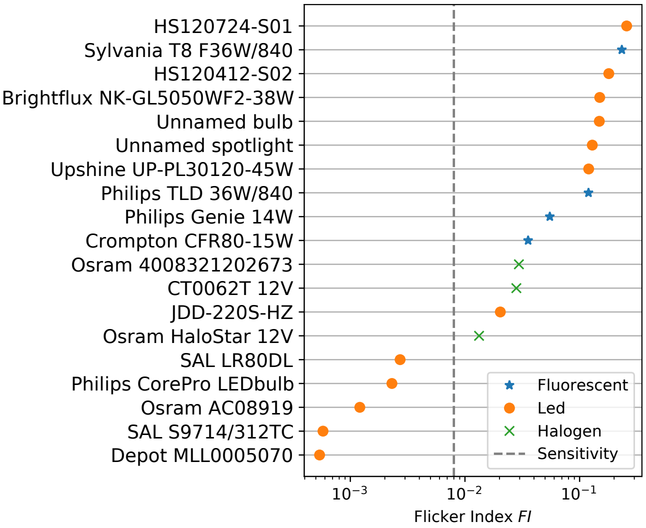


Figure 9: Flicker index for 19 tested lamps. The gray line marks the sensitivity of our Flync prototype. The proposed circuit works with all fluorescent and halogen lamps and the majority of LED lamps.

We conclude that most types of lamps exhibit significant powerline flicker, which makes this an attractive common energy signal. Next, we present our design of Flync, a hardware/software solution that extracts a frequency- and phase-synchronized clock signal from this common energy signal on distributed battery-free nodes. The dashed line in Fig. 9 is the measured sensitivity (see Sec. 5.2) of our Flync prototype, showing that the proposed design works with all fluorescent and halogen lamps and the majority of tested LED lamps.

3.2 Extracting a Clock from Solar Current

To be viable, Flync needs to provide a stable clock signal while keeping the required energy costs as low as possible.

Hardware. We propose the circuit shown in Fig. 10, which converts the modulated current signal from the solar panel into a digital clock signal that can be connected to a general purpose input/output (GPIO) pin of a MCU. The current through shunt resistor R_S causes a voltage drop that is filtered with a narrow-band bandpass filter to extract and amplify the powerline frequency component. We tune the band-pass filter to a gain of 36 dB at a center frequency of exactly double the powerline frequency, taking into account the limited gain-

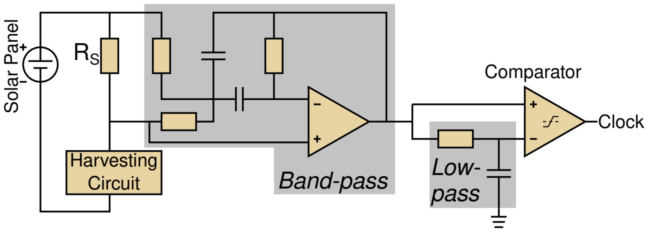


Figure 10: Flync circuit to extract a clock signal from the powerline-induced solar panel current variations (see Fig. 8 for an example).

bandwidth product of the low-power operational amplifier. The resulting signal is connected to a comparator directly and through a low-pass filter to convert it into a digital signal.

The TI TLV521 operational amplifier used in the band-pass filter has a typical current draw of 350 nA, and the TLV7031 comparator has a typical current draw of 315 nA. Including the losses over the 300 Ω shunt resistor, the Flync circuit draws a total of around 5 µW under typical harvesting conditions. This is orders of magnitude lower than the power draw of related approaches, using a light sensor and an analog-to-digital converter (ADC) (5.394 mW [31]) or an antenna to extract the signal from powerline radiation (300 µW [42]).

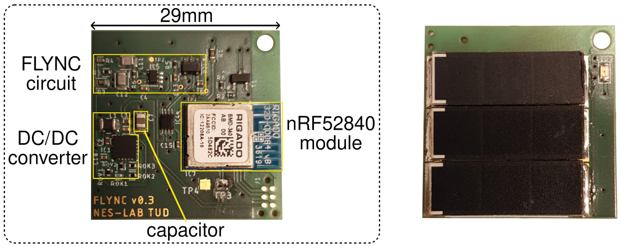
Software. To achieve a stable clock signal, we use a phase-locked loop (PLL) in combination with a proportional integral derivative (PID) controller to synchronize the MCU's real-time clock (RTC) to the powerline frequency signal extracted with our proposed circuit. In Sec. 4.2, we describe our software implementation of Flync in more detail.

3.3 Exploiting the Clock for Faster Discovery

Using Flync, neighboring battery-free nodes have access to a common clock. Nodes can use the phase information of this clock to implicitly agree on times at which they potentially become active. For the powerline flicker, this could be the rising edges of the solar panel current (see Fig. 8).

When using Find without Flync, we set the slot length to the duration of a node's active period. When using Find with Flync, we increase the slot length to $1/(2 \cdot f_{pl})$ and let nodes only become active at the beginning of a slot. This increases the probability that nodes become active in the same slot. For example, consider two nodes that randomly and uniformly wake up once within a 1 s time window. Using a slot length of 1 ms, the probability that both nodes wake up in the same slot is 1/1000. With a slot length of 10 ms, this probability is 10× higher, which speeds up the neighbor discovery process.

Flync exploits the well-behaved, widely available powerline flicker as sychronization source, but the concept applies to any phase-synchronized signal available on different nodes. Because the benefit in terms of a shorter discovery latency stems from increasing the effective slot length, the signal's period must be longer than the duration of a node's active period. The lower the frequency, the longer the slot length and the greater the potential benefit. If the period is longer than the charging time of a node, it can be divided down to avoid nodes wasting energy while waiting for the next slot.



(a) Front

(b) Back

Figure 11: Prototype battery-free node based on the nRF52840 MCU. Solar panels on the back charge a tiny capacitor that powers the node.

4 Prototype Implementation

This section describes the hardware and software components of our prototype implementation.

4.1 Hardware

We design a low-power battery-free node that integrates the circuit from Fig. 10. The node is based on a Nordic Semiconductor nRF52840 MCU, which features a 64 MHz ARM Cortex-M4F and a 2.4 GHz radio with support for Bluetooth 5.2 and IEEE 802.15.4. The node harvests energy using three 23 mm × 8 mm IXYS KXOB25-05X3F solar panels. A TI BQ25505 DC-DC boost converter steps up the voltage of the solar panels and charges a 2 mm × 1.25 mm × 1.25 mm 47 µF multilayer ceramic capacitor (MLCC). However, due to DC bias, the capacitor has only an effective capacitance of around 17 µF at 3.3 V. The BQ25505 implements a maximum power point tracking (MPPT) mechanism that aims to operate the solar panels close to their optimal voltage of around 80 % of the panels' open-circuit voltage. The MPPT circuit obtains a new reference voltage every 16 s by disabling the charger for 256 ms and sampling the panels' open-circuit voltage. Once the capacitor voltage reaches a hardware-programmable threshold of 3.3 V, the BQ25505 sets one of its pins high. This pin is connected to a TI TS5A23166 analog switch that connects the MCU to the capacitor-buffered supply voltage.

The two-layer printed circuit board (PCB) shown in Fig. 11 measures 29 mm × 29 mm. The total cost of all components is $13.89, including $8.11 for the relatively expensive, highly integrated nRF52840 module. Comparing our design to recently proposed battery-free platforms with similar capabilities in Table 1, we see that our prototype is indeed one of the first truly battery-free nodes in the sense that the energy storage is negligible in terms of cost, size, and environmental impact: The ceramic capacitor does not contain problematic materials, costs $0.024, and takes up only 0.3 % of the PCB area.

Platform	Year	Capacitor	Communication
Pible [14]	2018	220 mF super-cap	BLE
luxBeacon [22]	2019	1.5 F super-cap	BLE
Sigrist et al. [46]	2020	520 µF MLCC	BLE
Botoks [12]	2020	100 µF MLCC	868 MHz
This work	2021	47 µF MLCC	BLE PHY

Table 1: Our battery-free prototype node has a sustainable ceramic capacitor that is significantly smaller and cheaper than the energy storage of other recently proposed battery-free platforms.

4.2 Software

Next, we describe our implementation of an efficient runtime for battery-free nodes. We also detail the PLL implementation of Flync and key configuration parameters of Find.

Efficient runtime. Many existing battery-free runtimes discharge the capacitor until the voltage drops below the minimum and the MCU is powered off [12, 17]. To avoid the high energy costs of frequent hardware resets, we implement a different approach that we call *soft intermittency*. During charging, the MCU enters the lowest possible sleep mode, periodically waking up to sample the capacitor voltage with the built-in ADC. In this mode, we measure a total average power draw of 15 µW, including the power for the Flync circuitry and software processing. When the capacitor voltage reaches a software-defined turn-on threshold, the node arms the power-fail comparator, a dedicated peripheral that raises an interrupt when the capacitor voltage drops below a software-defined turn-off threshold. Then the node executes protocol and application code until it is notified by the power-fail comparator upon which it immediately transitions to deep sleep, drastically reducing its power draw until it has again buffered enough energy. While this soft intermittency approach cannot prevent hard resets when there is no energy input for several hundreds of milliseconds, it greatly increases the average efficiency without using additional comparators and switches.

Flync PLL. The comparator at the output of the circuit in Fig. 10 has a relatively small hysteresis, occasionally causing flickering at signal transitions. Furthermore, while MPPT obtains a new reference value, the harvesting current approaches zero, causing the clock signal to pause for hundreds of milliseconds. To provide a stable clock signal despite these disturbances, we implement a PLL that synchronizes the MCU's RTC to the signal extracted with the Flync circuit. We configure the GPIO peripheral to generate an interrupt on a rising edge at the GPIO pin connected to the output of the comparator of the circuit. After a reset, we wait for the first GPIO interrupt. Upon this interrupt, we set up an RTC interrupt to reset the RTC counter after the nominal powerline frequency interval. Ideally, all following GPIO interrupts should coincide with that RTC interrupt. Thus, the counter value at the time of the GPIO interrupt can be interpreted as phase deviation between the external clock signal and the local timer. We implement a control loop to continuously adjust the timer period in order to minimize the phase deviation. In this way, we obtain a highly stable interrupt that is phase-synchronized with the variations of the solar panel current and works even during the MPPT sampling or other disruptions.

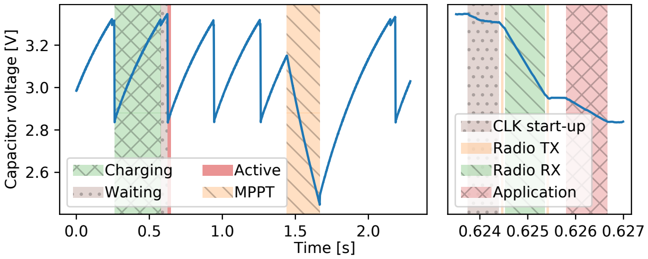


Figure 12: Example trace from a prototype node running Find.

Find settings. Each beacon in Find's frame structure shown in Fig. 7 consists of 2 B preamble, 3 B base address, 6 B payload, and 1 B cyclic redundancy check (CRC). When using the 2 Mbit BLE mode of the radio, this corresponds to a beacon transmission time of $T_{tx} = 48\,\mu\text{s}$. With 17 µF of capacitance, the time required to start the high-frequency oscillator, and a turn-around time of $T_{ta} = 40\,\mu\text{s}$, we can afford a maximum listening window of $T_{rx} = 800\,\mu\text{s}$. As a result, two nodes can successfully detect each other if they wake up with an offset $\mathcal{T}$ between 88 µs and 848 µs (see Fig. 7).

4.3 Example Real-world Trace

Fig. 12 shows capacitor voltage and activities over time while one of our prototype nodes runs Find. We see that the node charges its capacitor until reaching the turn-on threshold of 3.3 V. It wakes up and samples a random delay from Find's optimized distribution. The necessary computations cause a noticeable drop in the capacitor voltage when transitioning from charging to waiting. After the random delay, the node becomes active and quickly drains its capacitor below the turn-off threshold of 2.8 V. The overview on the left side of Fig. 12 also shows how the capacitor discharges during MPPT at around 1.5 s. The detailed view on the right side shows the individual stages while the node is active. We see that the node first starts the high-frequency clock required to run the radio. Then it sends the first beacon and starts to listen for potential beacons from other nodes. After listening for 800 µs, the node sends the trailing beacon. The remaining energy in the capacitor is assigned to the application that can run until the capacitor voltage hits the turn-off threshold.

5 Evaluation

We manufacture six prototype battery-free nodes to evaluate Find and Flync. We first look at their effectiveness in terms of discovery latency, followed by a detailed characterization of Flync's robustness and performance. Sec. 6 reports on the results of a contact tracing case study based on our techniques.

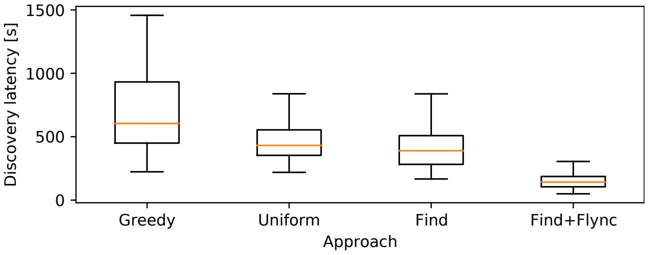


Figure 13: Discovery latency of four different approaches in a network of 6 battery-free nodes. Our techniques outperform the comparison approaches by up to 4.3× (median) and 34.4× (99th percentile).

5.1 Neighbor Discovery Performance

To fairly compare the neighbor discovery performance of our techniques against baseline approaches, we conduct experiments under controlled conditions. Sec. 6 reports on results when using Find and Flync in uncontrolled environments.

Setup. All experiments are conducted in a darkened room with a controllable light source. We place six prototype nodes next to each other on a flat surface. The nodes are programmed to output the ID of any discovered node over universal asynchronous receiver transmitter (UART), while a logic analyzer logs the output of every node. For each run, we let nodes wake up with a random initial delay, and consider the measured time until all 15 bi-directional links are discovered as the discovery latency. We compare Find and Find + Flync with a *greedy* approach, where nodes become active as soon as their capacitor voltage reaches the turn-on threshold, and a *uniform* approach, where nodes randomly delay their wake-ups by a uniformly distributed time. Overall, the measurement campaign took more than 4 days, in which we performed between 48 and 128 runs for each of the four approaches.

Results. Fig. 13 shows the measured discovery latency for each approach, including the median, the 25th and 75th percentiles, and the 1.5× of the interquartile range. Clearly, the greedy approach performs worst. This is mainly because of interleaved activity phases of the nodes, as visible from the trace in Fig. 14. If we zoom in on the first three and the last three wake-ups in the trace, we notice that nodes repeatedly wake up with the same pattern that prevents discovery despite different charging times and MPPT intervals. In Fig. 15, instead, we see that when nodes use Find to randomly delay each wake-up, they are more likely to be active at the same time. For instance, at about 4.5 s, the nodes wake up with an offset of less than 848 µs and are therefore able to successfully exchange beacons as shown in the detailed plot on the right side of Fig. 15. This explains the significant reduction in median discovery latency from 604 s with greedy to 390 s with Find, as visible in Fig. 13. We also see that Find's optimized delay distribution performs slightly better than the uniform approach (median of 431 s), which matches the magnitude of improvement predicted by our model (see Fig. 5).

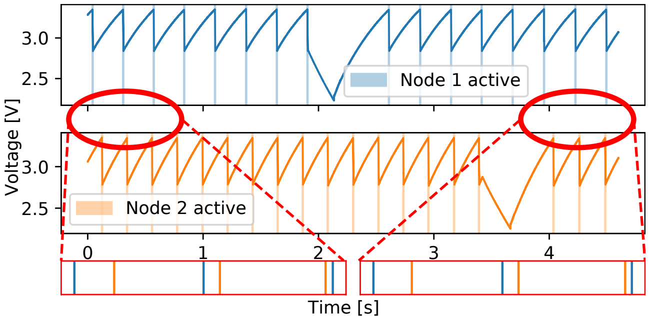


Figure 14: Interleaved activity phases of two nodes when using the *greedy* approach. The zoomed in plots on the bottom show that, despite the disturbances caused by MPPT, the two nodes repeatedly wake up with the same pattern, preventing successful discovery.

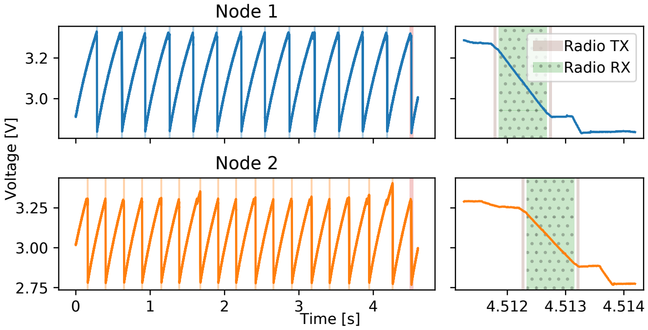


Figure 15: Using Find, nodes prevent *interleaving* by delaying each wake-up by a small random time, enabling quick discovery.

Find + Flync achieves the lowest median discovery latency of 142 s, which corresponds to an overall improvement of 4.3× (median) and 34.4× (99th percentile) compared with greedy.

5.2 Flync Sensitivity

To extract a clock signal, the Flync circuit requires a minimum magnitude of the powerline frequency component in the solar panel current. We empirically determine the corresponding minimum flicker index for our hardware prototype.

Method. The magnitude of the powerline frequency component is proportional to the DC component and decreases with smaller panel size and increasing distance from the light source. We define the worst-case minimum flicker index as the flicker index sufficient to extract a clock signal even at the lowest possible harvesting current. The latter is defined by the minimum power requirements of our prototype when running Find, the panel voltage, and the corresponding efficiency of the DC-DC converter. Our solar panels have a typical panel voltage of 1 V at the maximum power point. At this voltage, our DC-DC converter has an efficiency of 80 %. Thus, the minimum harvesting current to cover the power requirements of our prototype of about 37.5 µW is 50 µA.

We use a Keithley 2600B sourcemeter to generate a current signal with a DC offset of 50 µA while sweeping the ampli-

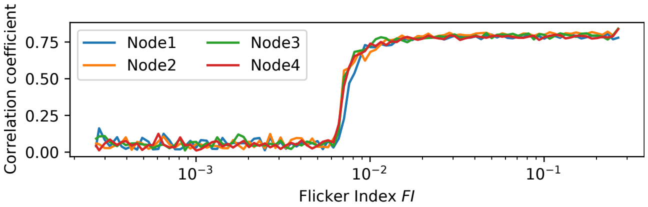


Figure 16: For a flicker index ≥0.008 Flync provides a stable clock.

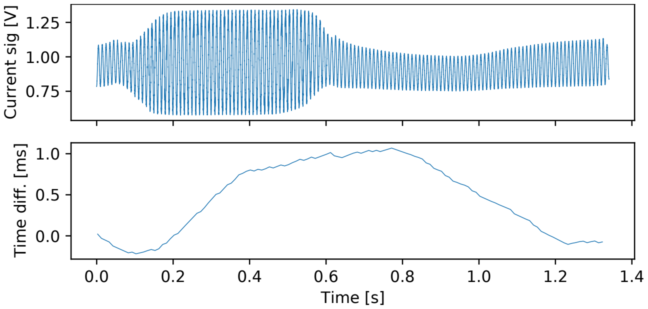


Figure 17: Current signal and time difference between two nodes while one node changes its distance and angle to the light source.

tude of the 100 Hz AC component. The current is fed to the input of our prototype that is usually connected to the solar panel. By limiting the voltage at the output of the sourcemeter to 1.25 V, the MPPT circuit regulates the input to around 1 V. For every setting of the AC amplitude, we record 5 s of clock signal with a mixed-signal oscilloscope. To quantify the quality of the clock signal, we compute the correlation coefficient between the signal and a phase-aligned 100 Hz reference. We repeat these measurements for four of our prototype nodes.

Results. The results in Fig. 16 show that there is a distinct threshold at around $FI = 0.008$ beyond which all nodes begin to output a clean clock signal. Comparing this with Fig. 9, we conclude that, with the exception of 5 LED lamps, our prototype works with the vast majority of the lamps we tested.

5.3 Flync Robustness

We now assess the robustness of Flync when a node changes its position and orientation relative to the light source, when the solar panels of a node are temporarily covered, and when electrical loads are temporarily connected to the same power strip. To this end, we experiment with two nodes powered by a desk lamp and connect them to an oscilloscope. We quantify robustness by measuring the time difference between clock edges on the two nodes. As a benchmark, we note that our implementation can tolerate a time difference of up to 848 µs.

Mobility. We keep one node static and attach the other one to the wrist of a person. The person waves, changing distance and angle between the node's solar panels and the light source.

Fig. 17 shows a period where the node moves closer and farther away from the lamp. The changes in the amplitude of the current signal affect the time difference between the nodes.

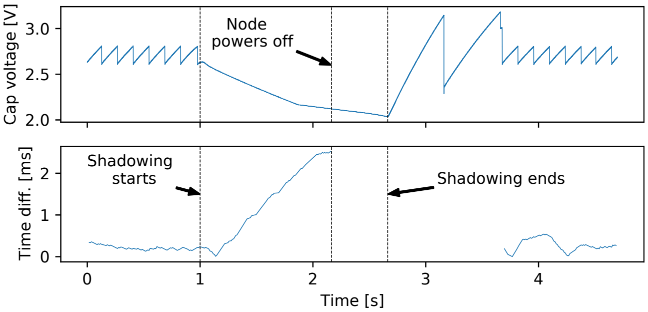


Figure 18: Capacitor voltage and time difference between two prototype nodes while temporarily covering the solar panel of one node.

The comparator that thresholds the sine wave uses a low-pass filter that reacts slowly to changes in the average amplitude. As a result, the clock signal deteriorates temporarily, causing an increased time difference of up to 1 ms. However, after a short while, the time difference recovers to previous levels.

Shadowing. To investigate the impact of shadowing, we put both nodes on a table and temporarily cover one of them by slowly moving a hand between the lamp and the node.

Fig. 18 shows that the time difference increases after covering the panel as the PLL loses its reference signal. However, without significant energy input, the node does not reach the turn-on threshold, which renders communication infeasible anyhow. As soon as the panel is uncovered, the node quickly charges up again and, after less than a second, the clock returns with a small time difference.

Electrical loads. We repeatedly switch on and off a drilling machine and a vacuum cleaner connected to the same power strip as the lamp. We do not observe any noticeable effect of the loads on the time difference between the two nodes.

5.4 Flync Jitter

In a final set of experiments, we look at the time difference between the clock signals of different nodes when these are: (*i*) powered by a single light source, (*ii*) placed in different rooms, and (*iii*) powered by different types of light sources.

Testbed. For these experiments, we built a distributed testbed of observer nodes. The observer nodes are accurately time-synchronized to within 479 ns, and record the clock signals of the attached prototype nodes with a resolution of 62.5 ns.

Single light source. We place six of our prototype nodes in the same room with a single halogen lamp. The experiments are conducted during the day, and the nodes receive a mixture of natural sunlight and artificial light from the lamp. Using our testbed, we record the clock edges of all six nodes for 1 h.

Fig. 19 shows the pairwise time difference between nodes. Because the phase offset resulting from propagation delays of light is negligible, the jitter must be introduced on each node. For example, a slight difference in the offset voltage of the comparator can lead to a significant mean difference of the

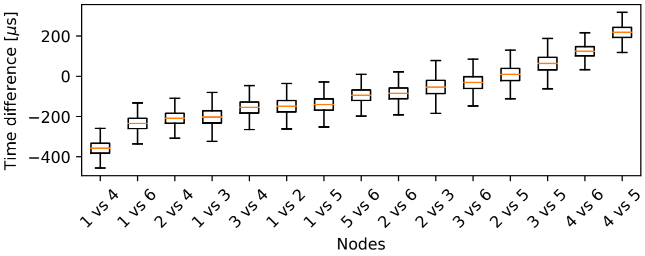


Figure 19: Pairwise time difference between clock edges on different prototype nodes when these are powered by a single light source.

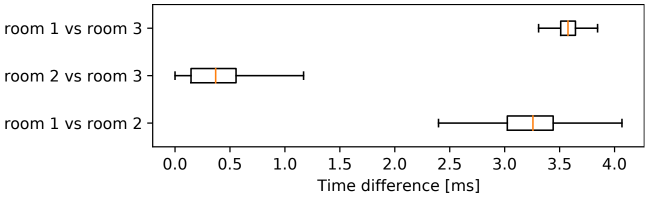


Figure 20: Pairwise time difference between clock edges on prototype nodes placed in different rooms with the same type of lamp.

resulting clock signal. Nevertheless, with 95 % of the more than five million recorded pairs below 244 µs, the jitter is well below the 848 µs tolerated by our Find implementation.

Different rooms. We conduct experiments in three rooms of an office building equipped with fluorescent tubes. The rooms are located on a long hallway with a distance of around 15 m between the middle room and the other two. We place two nodes in each room, and record with our testbed for 4 h while the nodes receive light from the tubes as well as sunlight.

Fig. 20 shows that there is a small offset between rooms 2 and 3 with 95 % of the recorded values being smaller than 700 µs. The offsets between rooms 1 and 2 and rooms 1 and 3 are centered around 3.3 ms. While residential homes are often connected to a single phase, larger apartment blocks or commercial buildings are typically fed by three-phase power. Apparently, the lights in room 1 are connected to a different phase than the lights in rooms 2 and 3, leading to a 60° phase and 3.3 ms time shift between the light intensity variations. Thus, when nodes need to discover neighbors across rooms with lights potentially connected to different power phases, they must be able to become active not only at the edge of their own Flync clock signal, but also with a 60° phase shift.

Different types of light sources. We plug an LED, a fluorescent, and two halogen lamps into the same power strip. We place one node under each lamp so that it only receives light from this lamp, and record for 30 min with our testbed.

Fig. 21 reveals large offsets between the clocks of nodes powered by different types of lamps. These offsets are due to varying phase shifts between the powerline voltage and the brightness variations of the lamp. For example, although the current through an incandescent lamp is in phase with the supply voltage, the filament may take some time to heat up and cool down, leading to the observed phase shift. Other types of lamps contain inductors or capacitive elements, a switching power supply, or an electronic ballast that cause different phase shifts. This shows that Flync does not work out of the box when different nodes are powered by different types of lamps. The static phase shifts would need to be measured during deployment or learned at runtime. On the other hand, Flync may not work reliably when individual nodes receive a mixture of light from different types of lamps. The results from the previous experiments (see Figs. 19 and 20) show that Flync works well when nodes receive a mixture of natural sunlight and artificial light from the same type of lamp.

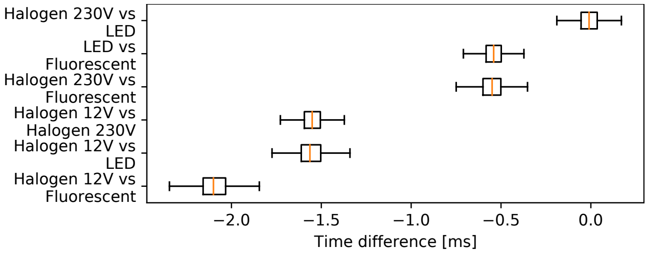
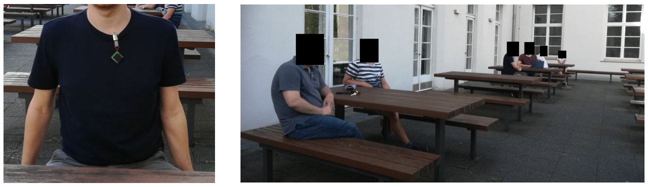


Figure 21: Pairwise time difference between clock edges on different prototype nodes when these are powered by different types of lamps.

6 Case Study: Contact Tracing

Automatic contact tracing is important to contain the spread of infectious diseases (e.g., SARS-CoV2) in a scalable manner. It allows to quickly identify contacts of an infected person and to quarantine potentially infected individuals before they become contagious. To assess the potential of our proposed designs for real-world battery-free applications, we conduct a contact tracing case study with our prototype nodes.

Setup. We attach six nodes to the shirts of human participants, as shown in Fig. 22a. The nodes run the Find protocol, logging the timestamp and ID of each discovered node to non-volatile memory. As we are only interested in relatively close contacts that would allow a virus to transmit from one person to another, we set the transmission power of the beacons to -16 dBm. We run experiments indoors and outdoors, as detailed below. After each run, we dump the content of the non-volatile memory of each node to a computer for analysis.



(a) Node on shirt. (b) Setup of experiment in an open-air pub.

Figure 22: Battery-free contact tracing.

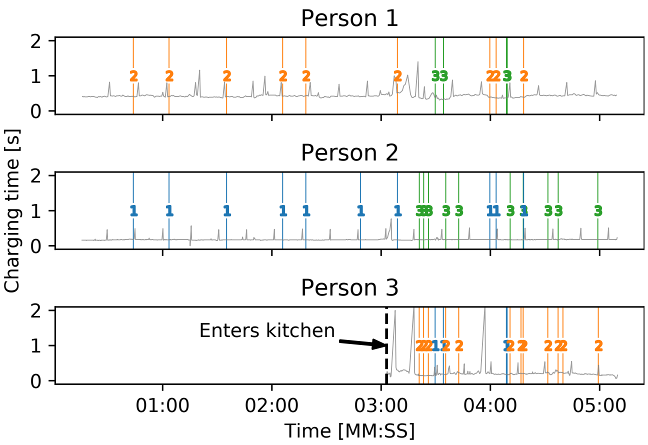


Figure 23: Charging times and rendezvous in coffee kitchen experiment. Vertical markers show rendezvous with the respective person.

Indoor experiment: coffee kitchen. Two persons sit at a table in a small coffee kitchen, roughly 1.5 m apart from each other. After 3 min a third person enters the kitchen and prepares a coffee for 2 min. The kitchen is equipped with fluorescent lamps, and we use Flync together with Find.

Fig. 23 plots the charging times and recorded rendezvous of the three nodes over time. We see a total of 49 received beacons. All contacts are logged successfully with low latency, despite the relatively long charging times of hundreds of milliseconds. Specifically, the first contact between persons 1 and 2 is detected after 43.9 s. When person 3 enters the kitchen, it takes 26.6 s and 17.9 s until the contacts with persons 1 and 2 are detected, respectively. Overall, the median time between rendezvous of the same two nodes is 7.5 s.

Outdoor experiment: open-air pub. Three pairs of persons sit at opposite sides of three tables (see Fig. 22b). Two tables are next to each other; the third table is at a distance of around 4.5 m. We perform the experiments in the morning of a slightly overcast day at an open-air pub without direct sunlight. Receiving only natural sunlight, the nodes do not make use of Flync. We conduct three consecutive 15 min runs.

We measure a total of 4426 received beacons. All contacts between persons on the same table are successfully recorded. More importantly, contacts between persons on different tables in close vicinity are also reliably detected. Due to the low transmit power, we do not see any rendezvous between the first two tables and the third remote table, which is expected and in fact desirable because we only want to trace contacts that are associated with an actual risk of virus transmission. Fig. 24 shows the histogram of the time between consecutive rendezvous between the same two nodes. As expected, the time between rendezvous is approximately exponentially distributed, and the mean is estimated between 2.61 s and 2.78 s with 95 % confidence. This means, under the given conditions, we are able to detect contacts with a resolution of around 2.67 s, allowing for fine-grained contact tracing.

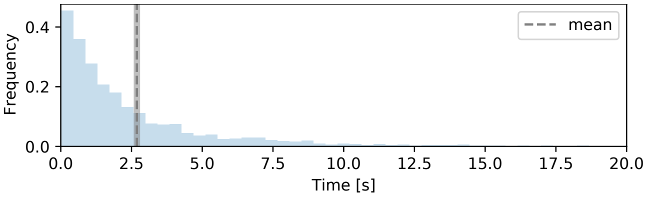


Figure 24: Histogram of the time difference between rendezvous of the same two nodes in the open-air pub experiment.

Summary. The results from our contact tracing case study show that Find and Flync are also effective under uncontrolled real-world conditions. Outdoors, energy availability is high and therefore Find alone enables fast rendezvous and fine-grained contact tracing. Indoors, Flync can compensate for the significantly lower energy density of interior light, providing decent performance even under these challenging conditions.

7 Discussion

We have presented two novel techniques that enable for the first time efficient device-to-device communication in the face of intermittency. By introducing random delays, Find breaks interleaved activity patterns of battery-free devices to discover each other faster and more efficiently. By tapping into the powerline-induced flicker of state-of-the-art lamps, Flync phase-synchronizes devices that harvest energy from indoor light. While we have exploited Flync to further speed up discovery in battery-free networks, Flync is useful for other purposes and also applicable to battery-supported devices.

Recent work tackles the intermittency problem on individual battery-free devices in terms of, for example, computing and time keeping [8,12,18,34]. We instead focus on communication between battery-free devices that operate intermittently. Like prior work, our techniques are relevant if intermittency makes traditional approaches inefficient or unreliable. To understand the scope of our work, we discuss intermittency and relevant impact factors below. Afterward, we discuss the influence of built-in randomness on our proposed techniques.

7.1 When Does Intermittency Occur?

A battery-free device goes through periods with low power requirements (e.g., system-off and sleep modes) and high power requirements (e.g., sensing, processing, and communication). Since the instantaneous power available from a harvester is often insufficient to support a battery-free device during periods with high power requirements, some form of energy storage is needed that buffers energy when the device is inactive to support a high-power workload for a short period of time.

The minimum size of the energy storage is determined by the demands of the largest atomic operation that must not be interrupted. For example, to transmit or receive a packet, the buffered energy needs to be sufficient to power the radio for at least the airtime of one complete packet; other examples of atomic operations include reading out a sensor or executing a checkpoint [21]. In our proposed Find protocol, the largest atomic operation is the frame sequence depicted in Fig. 7.

If a device with an active power draw higher than the harvesting power is equipped with an energy storage that does not support executing multiple iterations of the largest atomic operation from a single full charge, it is forced to go through periods of inactivity—the device is said to operate intermittently. Intermittency is in stark contrast to duty cycling, which is intentionally used on devices with primary or rechargeable batteries, yet the devices can become active at any point in time subject only to an upper bound on the average duty cycle. By contrast, intermittency prevents a device from becoming active at any point in time, and when a device enters and exits the inactivity phases is only partially controllable, at best.

7.2 What Factors Impact Intermittency?

Three key dimensions influence the extent of the intermittency problem: energy input, energy storage, and workload.

Energy input. An ambient energy source may exhibit intermittent behavior, including periods where it emits no energy. Clearly, a battery-free device can only harvest energy when the ambient source emits energy. In this case, provisioning a device with a harvester that provides the power required to continuously operate the device in high-power mode prevents the intermittency problem. This, however, would come with major drawbacks in terms of size, weight, and costs. For example, a battery-free device may draw only 10 µW on average but 10 mW when active, thus requiring to over-provision the harvester by a factor of 1000. While such over-provisioning is in theory always possible, it is severely limited in practice by the constraints imposed by the application requirements.

Energy storage. If permitted by the application requirements, an energy storage larger than the minimum required to execute the largest atomic operation may be used. For example, using a high-capacity rechargeable battery can prevent intermittency. Such batteries have a high energy density, but their minimum physical dimensions are typically orders of magnitude larger than those of capacitors. Batteries are also more expensive and subject to aging, losing capacity over time and eventually malfunctioning with excessive heat and leakage of potentially toxic chemicals. By contrast, capacitors have low energy density, but are extremely cheap, readily available in sizes well below 0.1 mm^3, have negligible aging effects, and do not contain problematic materials (e.g., toxic chemicals). Thus, despite advances in battery technology, alternative systems to store energy are being explored [3] and capacitors are widely regarded as a more sustainable option [11, 45].

When a device is inactive, it accumulates charge until the capacitor voltage reaches a turn-on threshold. The amount of energy that can be stored depends on the turn-on threshold, which is limited by the breakdown voltage of the capacitor and the device's maximum operating voltage. When a device is active, it discharges the capacitor until the voltage reaches a turn-off threshold, which is dictated by the device's minimum operating voltage. Thus, for the same capacitor, a device with a lower minimum operating voltage or a higher maximum operating voltage can increase the effective amount of buffered energy that can be used. This allows to either use a smaller capacitor or execute longer from a single full charge, potentially alleviating the intermittency problem.

Workload. While lower-power hardware can reduce the average power draw in sleep mode and thus the charging time, it does not generally avoid intermittency. This would require pushing also the active power below the harvesting power.

Reducing the transmission power of the radio can extend the time a device can operate from a single full charge. While this may alleviate the intermittency problem, it also reduces the communication range, which may render device-to-device communication infeasible or require multi-hop networking.

Similarly, using backscatter communication instead of active radio communication may bring the active power draw of a device below the harvesting power and thereby enable continuous operation. However, backscatter requires the presence of an external carrier and may pose limitations in terms of communication range and data rate. In particular, existing practical implementations of tag-to-tag backscatter receivers do not yet reach the point where the end-to-end power draw is negligible (i.e., below sleep power of around 1 µW) [35, 39], thus leaving a significant region in the design space of battery-free backscatter devices where intermittency occurs.

7.3 Impact of Built-in Spatial Randomness

Find tackles interleaving by letting nodes randomly and independently delay their wake-ups. This approach is particularly effective in scenarios with little built-in spatial randomness, that is, when the harvested energy exhibits limited variability *between* nodes, regardless of a potentially high temporal variability in harvested energy. We believe this holds for a broad class of battery-free application scenarios, because nodes in a confined space often harvest energy from the same ambient source(s). On the other hand, a high built-in spatial randomness may alleviate the interleaving problem. Although our case study experiments exhibit built-in spatial and temporal randomness, it remains an open question how built-in spatial randomness may influence the choice of Find's delay distribution and scale parameter as well as its overall effectiveness.

8 Related Work

Battery-free device-to-device communication. Prior work on battery-free wireless device-to-device communication is mainly theoretical [25, 51], studying the capacity limits for different energy scheduling, transmission, and decoding policies. Understanding energy issues on the receiver side [2] and the impact of intermittency have been open problems. On the other hand, practical work on tag-to-tag backscatter communication has primarily focused on physical-layer issues and considers intermittency an orthogonal problem [20, 32, 35].

Work	Type	Sensing Signal	Power
Syntonistor [42]	frequency	EM radiation	300 µW
Flight [31]	frequency	light sensor	5394 µW
Flync	freq.+phase	solar current	5 µW

Table 2: Compared with prior work using powerline frequency for synchronization, Flync provides frequency and phase synchronization from the solar panel current at significantly lower power draw.

Rendezvous and neighbor discovery protocols. Blind rendezvous is the process of establishing a communication link between nodes in a distributed system without any prior information [16]. Neighbor discovery protocols for wireless networks target a sub-class of the blind rendezvous problem with the goal of optimizing the trade-off between discovery latency and energy consumption. Deterministic protocols let nodes wake up according to a schedule based on (co-)prime numbers [13, 23], a quorum [19, 27, 28], or by systematically traversing slots [5, 50]. This way, they can provide guaranteed bounds on discovery latency [24]. Probabilistic protocols are stateless, robust to varying conditions, and offer low average discovery latency [10]. For example, the influential birthday protocol [37] and follow-up work [48, 49] analyze optimal transmit probabilities to maximize the fraction of links discovered in a given time. However, none of the existing neighbor discovery protocols are applicable to battery-free networks because they require nodes to be able to wake up at arbitrary points in time, not taking into account intermittency.

Powerline-based clock synchronization. We are not the first to exploit the powerline frequency signal for synchronization. The Syntonistor extracts a stable clock signal from electromagnetic (EM) powerline radiation using a large coil [42]. It draws 300 µW of power, 60× more than Flync. Flight samples a light sensor to synchronize a node's oscillator to the powerline-induced brightness variations of fluorescent lamps [31]. Using Flight, synchronization takes 100 ms at a power draw of 5394 µW, 1000× more than Flync. As summarized in Table 2, both approaches only synchronize the frequency of local clocks, eliminating the need to periodically compensate for clock drift, but do not exploit phase information. They also use dedicated high-power sensors, whereas Flync uses a low-power circuit to extract the signal from the current of the solar panel.

Energy harvesters as sensors. Previous work has explored the use of the harvesting current or voltage as a sensing signal for indoor positioning [41], gait recognition [33], gesture recognition [47], activity classification [44], and transport-mode detection [43]. To the best of our knowledge, we are the first to exploit context information from harvested energy for synchronization. Furthermore, Flync is the first design that extracts the sensing signal from current variations of a solar panel that is simultaneously used to power the system.

Visible light communication. Flync exploits the powerline-induced brightness variations as an intrinsic property of ubiquitous lamps. When modifying existing lighting infrastructure, it is possible to encode arbitrary data into the brightness variations. This opportunity has been used for downlink communication [40], indoor positioning [26], and battery-free duplex visible light communication [29]. By modulating light with a well-defined synchronization signal, the efficiency and applicability of Flync could be further improved. Also, our approach to harvest energy while simultaneously demodulating encoded signals from the same panel may reduce the size and power of existing visible light communication receivers.

9 Conclusions

Leaving batteries behind allows for building cheap, tiny, and maintenance-free devices that can be embedded into smart textiles, intelligent surfaces, or even the human body. In this paper, we have addressed the problem of enabling efficient battery-free device-to-device communication. Experiments with a prototype platform and implementation show that our proposed techniques empower battery-free devices to quickly and efficiently discover each other despite their unpredictable intermittent operation. By bootstrapping battery-free wireless networks, we believe that our work provides a stepping stone for future research toward full system and communication stacks for this emerging kind of networked system.

Availability

Artifacts are available to the public under a permissive MIT license at https://find.nes-lab.org/. These include a Python implementation of the Find model from Sec. 2, which can be used to reproduce the analytical results in Figs. 3 to 6, as well as the hardware design files and the firmware of our prototype implementation from Sec. 4, which we used for the experiments and case study described in Secs. 5 and 6.

Acknowledgments

Thanks to Brano Kusy, Carlos Pérez-Penichet, and Carsten Herrmann for their feedback that helped improve this paper, to Friedrich Schmidt for his help with the hardware design, to James Broadhead for the preliminary experiments, and to all participants of the contact tracing case study. Thanks also to the anonymous reviewers, and to our shepherd, Shyam Gollakota. This work was supported by the Emmy Noether project NextIoT (DFG grant ZI 1635/2-1) and cfaed.

References

[1] T52x/t530 polymer electrolytic capacitors. Technical report, KEMET Electronics Corporation, 2018.

[2] Kofi Sarpong Adu-Manu, Nadir Adam, Cristiano Tapparello, Hoda Ayatollahi, and Wendi Heinzelman. Energy-harvesting wireless sensor networks (EH-WSNs): A review. *ACM Transactions on Sensor Networks*, 14(2), 2018.

[3] Science Communication Unit at University of the West of England. Towards the battery of the future. *Science for Environment Policy*, 2018.

[4] Abu Bakar and Josiah Hester. Making sense of intermittent energy harvesting. In *Proceedings of the 6th ACM International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems (ENSSys)*, 2018.

[5] Mehedi Bakht, Matt Trower, and Robin Hilary Kravets. Searchlight: Won't you be my neighbor? In *Proceedings of the 18th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2012.

[6] James Bates, Marc Beaulieu, Michael Miller, and Joseph Paulus. Reaching the highest reliability for tantalum capacitors. Technical report, AVX Corporation, 2013.

[7] Naveed Anwar Bhatti, Muhammad Hamad Alizai, Affan A. Syed, and Luca Mottola. Energy harvesting and wireless transfer in sensor network applications: Concepts and experiences. *ACM Transactions on Sensor Networks*, 12(3), 2016.

[8] Adriano Branco, Luca Mottola, Muhammad Hamad Alizai, and Junaid Haroon Siddiqui. Intermittent asynchronous peripheral operations. In *Proceedings of the 17th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2019.

[9] R. P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4), 1971.

[10] Lin Chen and Kaigui Bian. Neighbor discovery in mobile sensing applications: A comprehensive survey. *Ad Hoc Networks*, 48, 2016.

[11] Albert Cohen, Xipeng Shen, Josep Torrellas, James Tuck, Yuanyuan Zhou, Sarita Adve, Ismail Akturk, Saurabh Bagchi, Rajeev Balasubramonian, Rajkishore Barik, Micah Beck, Ras Bodik, Ali Butt, Luis Ceze, Haibo Chen, Yiran Chen, Trishul Chilimbi, Mihai Christodorescu, John Criswell, Chen Ding, Yufei Ding, Sandhya Dwarkadas, Erik Elmroth, Phil Gibbons, Xiaochen Guo, Rajesh Gupta, Gernot Heiser, Hank Hoffman, Jian Huang, Hillery Hunter, John Kim, Sam King, James Larus, Chen Liu, Shan Lu, Brandon Lucia, Saeed Maleki, Somnath Mazumdar, Iulian Neamtiu, Keshav Pingali, Paolo Rech, Michael Scott, Yan Solihin, Dawn Song, Jakub Szefer, Dan Tsafrir, Bhuvan Urgaonkar, Marilyn Wolf, Yuan Xie, Jishen Zhao, Lin Zhong, and Yuhao Zhu. Inter-disciplinary research challenges in computer systems for the 2020s. Technical report, 2018.

[12] Jasper de Winkel, Carlo Delle Donne, Kasim Sinan Yildirim, Przemysław Pawełczak, and Josiah Hester. Reliable timekeeping for intermittent computing. In *Proceedings of the 25th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2020.

[13] Prabal Dutta and David Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2008.

[14] Francesco Fraternali, Bharathan Balaji, Yuvraj Agarwal, Luca Benini, and Rajesh Gupta. Pible: Battery-Free Mote for Perpetual Indoor BLE Applications. In *Proceedings of the 5th ACM Conference on Systems for Built Environments (BuildSys)*, 2018.

[15] Kai Geissdoerfer, Mikołaj Chwalisz, and Marco Zimmerling. Shepherd: A portable testbed for the batteryless IoT. In *Proceedings of the 17th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2019.

[16] Zhaoquan Gu, Yuexuan Wang, Qiang-Sheng Hua, and FC Lau. *Rendezvous in Distributed Systems*. Springer, 2017.

[17] Josiah Hester and Jacob Sorber. Flicker: Rapid prototyping for the batteryless Internet-of-Things. In *Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2017.

[18] Josiah Hester, Nicole Tobias, Amir Rahmati, Lanny Sitanayah, Daniel Holcomb, Kevin Fu, Wayne P. Burleson, and Jacob Sorber. Persistent clocks for batteryless sensing devices. *ACM Transactions on Embedded Computing Systems*, 15(4), 2016.

[19] Tingpei Huang, Haiming Chen, Li Cui, and Yuqing Zhang. EasiND: Neighbor discovery in duty-cycled asynchronous multichannel mobile WSNs. *International Journal of Distributed Sensor Networks*, 9(7), 2013.

[20] Vikram Iyer, Vamsi Talla, Bryce Kellogg, Shyamnath Gollakota, and Joshua Smith. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the ACM SIGCOMM Conference*, 2016.

[21] Neal Jackson, Joshua Adkins, and Prabal Dutta. Capacity over capacitance for reliable energy harvesting sensors. In *Proceedings of the 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2019.

[22] Kang Eun Jeon, James She, Jason Xue, Sang-Ha Kim, and Soochang Park. luXbeacon—A batteryless beacon for green IoT: Design, modeling, and field tests. *IEEE Internet of Things Journal*, 6(3), 2019.

[23] Arvind Kandhalu, Karthik Lakshmanan, and Ragunathan (Raj) Rajkumar. U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2010.

[24] Philipp H. Kindt and Samarjit Chakraborty. On optimal neighbor discovery. In *Proceedings of the ACM SIGCOMM Conference*, 2019.

[25] Meng-Lin Ku, Wei Li, Yan Chen, and K. J. Ray Liu. Advances in energy harvesting communications: Past, present, and future challenges. *IEEE Communications Surveys & Tutorials*, 18(2), 2016.

[26] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2014.

[27] Shouwen Lai, Binoy Ravindran, and Hyeonjoong Cho. Heterogenous Quorum-Based Wake-Up Scheduling in Wireless Sensor Networks. *IEEE Transactions on Computers*, 59(11), 2010.

[28] Shouwen Lai, Bo Zhang, Binoy Ravindran, and Hyeonjoong Cho. CQS-Pair: Cyclic quorum system pair for wakeup scheduling in wireless sensor networks. In *Proceedings of the International Conference on Principles of Distributed Systems (OPODIS)*. Springer, 2008.

[29] Jiangtao Li, Angli Liu, Guobin Shen, Liqun Li, Chao Sun, and Feng Zhao. Retro-VLC: Enabling battery-free duplex visible light communication for mobile and IoT applications. In *Proceedings of the 16th ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2015.

[30] Tianxing Li and Xia Zhou. Battery-free eye tracker on glasses. In *Proceedings of the 24th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[31] Zhenjiang Li, Wenwei Chen, Cheng Li, Mo Li, Xiang-Yang Li, and Yunhao Liu. FLIGHT: Clock calibration using fluorescent lighting. In *Proceedings of the 18th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2012.

[32] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. Ambient backscatter: Wireless communication out of thin air. In *Proceedings of the ACM SIGCOMM Conference*, 2013.

[33] Dong Ma, Guohao Lan, Weitao Xu, Mahbub Hassan, and Wen Hu. SEHS: Simultaneous energy harvesting and sensing using piezoelectric energy harvester. In *Proceedings of the 3rd IEEE/ACM International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2018.

[34] Kiwan Maeng and Brandon Lucia. Adaptive dynamic checkpointing for safe efficient intermittent computing. In *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2018.

[35] Amjad Yousef Majid, Michel Jansen, Guillermo Ortas Delgado, Kasim Sinan Yildirim, and Przemysław Pawełczak. Multi-hop backscatter tag-to-tag networks. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2019.

[36] Amjad Yousef Majid, Patrick Schilder, and Koen Langendoen. Continuous sensing on intermittent power. In *Proceedings of the 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2020.

[37] Michael J. McGlynn and Steven A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2001.

[38] Sung Y. Park and Anil K. Bera. Maximum entropy autoregressive conditional heteroskedasticity model. *Journal of Econometrics*, 150(2), 2009.

[39] Carlos Pérez-Penichet, Claro Noda, Ambuj Varshney, and Thiemo Voigt. Battery-free 802.15.4 receiver. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2018.

[40] Niranjini Rajagopal, Patrick Lazik, and Anthony Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2014.

[41] Julian Randall, Oliver Amft, Jürgen Bohn, and Martin Burri. LuxTrace: Indoor positioning using building illumination. *Personal and Ubiquitous Computing*, 11(6), 2007.

[42] Anthony Rowe, Vikram Gupta, and Ragunathan (Raj) Rajkumar. Low-power clock synchronization using electromagnetic energy radiating from AC power lines. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2009.

[43] Muhammad Moid Sandhu, Kai Geissdoerfer, Sara Khalifa, Raja Jurdak, Marius Portmann, and Brano Kusy. Towards energy positive sensing using kinetic energy harvesters. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2020.

[44] Muhammad Moid Sandhu, Sara Khalifa, Kai Geissdoerfer, Raja Jurdak, and Marius Portmann. SolAR: Energy positive human activity recognition using solar cells. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2021.

[45] Mahadev Satyanarayanan, Wei Gao, and Brandon Lucia. The computing landscape of the 21st century. In *Proceedings of the 20th ACM International Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2019.

[46] Lukas Sigrist, Rehan Ahmed, Andres Gomez, and Lothar Thiele. Harvesting-aware pptimal communication scheme for infrastructure-less sensing. *ACM Transactions on Internet of Things*, 1(4), 2020.

[47] Ambuj Varshney, Andreas Soleiman, Luca Mottola, and Thiemo Voigt. Battery-free visible light sensing. In *Proceedings of the 4th ACM Workshop on Visible Light Communication Systems (VLCS)*, 2017.

[48] Sudarshan Vasudevan, Micah Adler, Dennis Goeckel, and Don Towsley. Efficient algorithms for neighbor discovery in wireless networks. *IEEE/ACM Transactions on Networking*, 21(1), 2013.

[49] Sudarshan Vasudevan, Donald Towsley, Dennis Goeckel, and Ramin Khalili. Neighbor discovery in wireless networks and the coupon collector's problem. In *Proceedings of the 15th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2009.

[50] Keyu Wang, Xufei Mao, and Yunhao Liu. BlindDate: A neighbor discovery protocol. In *Proceedings of the 42nd International Conference on Parallel Processing (ICPP)*, 2013.

[51] Tongxin Zhu, Jianzhong Li, Hong Gao, and Yingshu Li. Broadcast scheduling in battery-free wireless sensor networks. *ACM Transactions on Sensor Networks*, 15(4), 2019.