Efficient Neighbour Discovery for Battery-Free IoT Using RF Information Harvesting

Saptarshi Hazra
Dept. of Electrical Engineering
Uppsala University
Uppsala, Sweden
saptarshiv2hazra@gmail.com

Fehmi Ben Abdesslem
Dept. of Computer Science
RISE
Stockholm, Sweden
fehmi.ben.abdesslem@ri.se

Thiemo Voigt
Dept. of EE, Uppsala University
Dept. of CS, RISE
Uppsala/Stockholm, Sweden
thiemo.voigt@angstrom.uu.se

Abstract—Battery-free Internet of Things (IoT) devices, powered by ambient energy harvesting, offer sustainable operation but face significant challenges due to energy intermittency, particularly for network communication tasks like neighbour discovery. Traditional discovery protocols often assume predictable energy availability and can be inefficient for energy-harvesting nodes. In this paper, we propose an efficient neighbour discovery protocol (BFND) that leverages ultra-low-power RF information harvesting to intelligently guide the transmission schedule of energy-intensive advertisements. Nodes use low-energy scans to build a channel activity map and probabilistically choose between scanning or advertising, prioritising advertisement attempts during slots with previously detected activity. We evaluate our protocol against a baseline approach using a discrete-event simulator incorporating realistic energy harvesting traces. Simulation results over 1000 runs show that our RF harvesting-enhanced protocol significantly reduces discovery latency and improves reliability. For instance, in medium energy-scarcity environments with 5 nodes, BFND achieves 100% success and is over 26× faster than the baseline. Under severe energy scarcity (representing transient RF harvesting from road traffic), the baseline fails completely with a 100% timeout rate, whereas BFND still achieves a 40.5% success rate.

Index Terms—Battery-free IoT; Neighbour Discovery; Energy Harvesting; RF Information Harvesting; Low-Power Wireless Communication; Intermittent Computing.

I. Introduction

The need for frequent battery replacements in the Internet of Things (IoT) motivates the development of battery-free devices powered by ambient energy harvesting (e.g., photovoltaic, piezoelectric). However, their deployment is challenged by unpredictable and intermittent energy availability. While previous studies have largely focused on ensuring computational task completion under intermittency, communication between battery-free devices remains a critical challenge. Specifically, a primary hurdle is ensuring two devices discover each other to establish communication links.

Challenge: Neighbour discovery enables devices in radio proximity to become aware of each other, typically requiring active advertisements and listening. A primary objective in battery-free environments is to quickly find neighbours while minimising active radio usage to conserve energy. Neighbour discovery is a well-studied problem in traditional networks [1], [2], where protocols typically require sampling the radio medium for long periods. Alternative strategies [3]–[5] employ deterministic duty cycles as a basis for developing stochastic techniques to expedite the process of identifying neighbouring devices. Translating these strategies to battery-free devices is a challenging task because nodes operate intermittently under erratic energy harvesting conditions [6]. Under the intermittent computing paradigm, programs execute in bursts punctuated by frequent power outages, which erase volatile state and disrupt hardware timekeepers [7], [8]. Consequently, the periods of active radio communication are highly sporadic and dependent on environmental conditions, making traditional slot alignment and continuous duty cycles impractical [9].

In order to infer radio channel occupancy, specific hardware for RF information harvesting using passive components has been introduced [10], [11]. Although similar in functionality to clear channel assessment used by, e.g., Noda et al. [12], RF information harvesting uses analog components for energy detection rather than active radio, leading to lower power consumption. Using an amplifier, comparable sensitivity can be achieved, while still consuming less energy.

In this paper, we propose BFND (Battery-Free Neighbour Discovery), a method that exploits the ultra-low power sensing of RF information harvesting to enable efficient neighbour discovery. Our results demonstrates that BFND enables efficient neighbour discovery across various ambient energy conditions.

Core Contributions: The main contributions of this work are:

- We design BFND, an efficient neighbour discovery protocol that utilises ultra-low-power RF information harvesting to construct a local channel activity map and schedule advertisements.
- We develop a custom discrete-event network simulator in Python that models capacitor voltage transitions, radio collisions, and protocol operational cycles for battery-free nodes.
- We perform an extensive empirical evaluation across five real-world harvesting profiles representing diverse

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

office, domestic, and outdoor activities, and node counts $N \in \{2, 3, 4, 5\}$. Our results show that BFND reduces discovery latency by up to over 26× in energy-scarce scenarios and maintains high discovery reliability, whereas the baseline protocol collapses and suffers a 100% timeout rate under severe energy constraints.

II. Related Work

Neighbour discovery is a fundamental primitive for wireless networking. We categorise the related literature into three main domains: traditional asynchronous discovery protocols, communication protocols for battery-free systems, and low-power RF sensing techniques.

A. Asynchronous Neighbour Discovery

In powered wireless sensor networks (WSNs) and mobile systems, neighbour discovery protocols traditionally operate under the assumption of continuous power availability, aiming to minimise latency and energy consumption through scheduled or stochastic duty cycles. Deterministic protocols such as Disco [3], U-Connect [4], and Searchlight [5] rely on prime-number mathematical relationships or slotted offsets to guarantee overlapping wake-up windows within a bounded period. Stochastic approaches, such as the Birthday protocol [13], use random transmit-and-listen probabilities. While highly effective for battery-powered systems, these protocols assume that nodes can wake up precisely according to scheduled slot boundaries. In battery-free systems, this assumption fails because the wake-up times and operational periods are strictly dictated by ambient energy harvesting, which is highly intermittent and unpredictable.

B. Battery-Free Systems and Intermittent Computing

Battery-free IoT nodes operate in an intermittent computing model: they execute tasks when they harvest sufficient energy into a capacitor, and power-cycle when their capacitor voltage drops below a threshold (V_{OFF}) [14].

Panda [15] designs optimal neighbour discovery algorithms under solar energy-harvesting budgets by modelling state-of-charge transitions. However, Panda assumes that nodes can maintain their state/time-alignment across cycles (or have predictability in energy harvesting), and does not leverage passive RF channel activity mapping to adapt advertisement schedules dynamically. A foundational slotted synchronisation protocol, Flync [14], enables bootstrapping and synchronisation by exploiting the invisible flicker of ambient indoor light bulbs (e.g., fluorescent and LED lights connected to the grid) as a shared clock source. Although highly effective indoors, Flync cannot be applied to outdoor environments, such as smart road infrastructure, where light flicker is unavailable. In contrast, our work focuses on leveraging RF energy harvesting, which is readily available in environments like road traffic or near cellular towers, to bootstrap discovery.

Geissdoerfer and Zimmerling also propose Bonito [16], which maintains bi-directional connections between battery-free devices by estimating charging durations, though it targets a two-device setting rather than multi-device neighbour discovery. While de Winkel et al. [17] enable standard Bluetooth low-energy communication on intermittent devices by checkpointing protocol state across power outages, they focus on single-link connections under steady charging profiles. BFND, by contrast, targets multi-node discovery under highly intermittent, non-periodic energy traces without requiring heavy checkpointing overhead. WEND [18] enables neighbour discovery among battery-free devices using wake-up receivers for low-power channel monitoring, with piggybacking of known neighbours to speed up the process. It does not, however, exploit channel activity patterns to schedule advertisements toward slots where neighbours are most likely to be active. Pulsar [19] and FreeBeacon [20] both address battery-free discovery by introducing a small number of battery-powered beacons that assign unique time slots to battery-free devices, enabling structured communication without random guessing. FreeBeacon additionally provides formal discovery guarantees using co-prime cycle lengths and extends the approach to structured data aggregation, but like Pulsar, it requires at least one battery-powered node per network. BFND avoids such infrastructure dependency by operating in a fully battery-free setting. Because protocols like WEND require specific hardware (wake-up receivers) and Pulsar/FreeBeacon rely on battery-powered infrastructure beacons, they cannot be directly evaluated in our fully symmetric, battery-free energy-harvesting setup without altering the fundamental hardware and deployment assumptions. Therefore, we evaluate BFND against FIND [14], which operates under the exact same infrastructure-free constraints.

C. Channel Sensing and Information Harvesting

Clear Channel Assessment (CCA) is commonly used in wireless protocols (e.g., Noda et al. [12]) to detect channel occupancy. Beyond channel quality measurement, CCA has also been employed as a rendezvous mechanism in receiver-initiated MAC protocols such as RIVER-MAC [21] that uses it to reduce idle listening overhead. However, conventional CCA requires activating the RF front-end and analogue-to-digital converters. This consumes milliwatts of power, which is prohibitive for frequently resetting battery-free nodes. To bypass this limitation, Hokke et al. [10] introduced RF information harvesting, which uses passive, analogue envelope-detectors to sense channel activity. This approach consumes microwatts of power by utilising harvested RF energy for detection rather than active radio components. Our work builds upon this concept. While Hokke et al. focused on medium access control, we integrate RF information harvesting into the neighbour discovery layer,

using low-energy scans to map channel activity patterns and guide advertisement schedules.

III. System Design

This section describes the hardware architecture, and power management of our battery-free IoT node, followed by the detailed design of the BFND protocol.

A. Hardware Architecture and Energy Model

We consider a battery-free IoT node powered entirely by ambient energy harvesting. The node consists of three primary subsystems: an energy-harvesting unit, a RF-sensing unit, and a microcontroller unit (MCU) with a low-power wireless transceiver. Figure 1 provides an architectural overview.

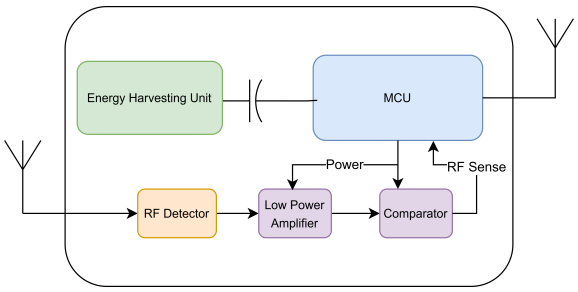


Fig. 1. Architectural overview of the battery-free IoT node, comprising the energy-harvesting unit, the RF-sensing unit, and the MCU subsystem.

Energy Harvesting Unit: The energy-harvesting unit employs transducers (e.g., piezoelectric or photovoltaic harvesters) to charge a storage capacitor of capacitance C. The instantaneous energy stored in the capacitor is:

$$E(t) = \frac{1}{2} C V(t)^2 \tag{1}$$

where $V(t)$ is the terminal voltage.

RF Sensing Unit: The RF-sensing path employs a passive envelope detector to rectify incident RF energy into a low-voltage DC signal. Due to the low amplitude of the detector's output, a low-power amplifier stage (analogous to the wake-up receiver amplifier in [11]) boosts the signal before it reaches a low-power comparator. The comparator evaluates the amplified voltage against a reference voltage (typically 50–100 mV) and outputs a high digital signal to trigger an external interrupt on the MCU. The resulting hardware interrupt wakes the MCU from sleep mode, allowing it to register channel activity without the overhead of active polling. Finally, the MCU determines the scanning schedule and powers the scan circuit through general-purpose input/output (GPIO) pins.

Microcontroller Unit: The MCU executes the discovery protocol and manages communication, operating from the capacitor voltage through a DC/DC regulator. It supports four operating actions with specific energy costs: advertisement (transmitting a neighbour-discovery beacon), active channel scanning, sleep (low-power idle), and busy-wait (active processing). The active scanning cost utilising our passive RF-sensing circuit is orders of magnitude smaller than active radio advertisement ($E_{\text{scan}} \ll E_{\text{adv}}$), which is the primary energy asymmetry we exploit.

B. MCU Power Management

The MCU operating range is governed by four voltage thresholds defined over the capacitor voltage V, summarised in Table I.

TABLE I
MCU voltage thresholds and their effects on node operation.

Threshold	Symbol	Condition	Effect
Turn-on	V_{ON}	$V \geq V_{\text{ON}}$	MCU transitions from OFF to ON
Turn-off	V_{OFF}	$V < V_{\text{OFF}}$	MCU transitions from ON to OFF
Maximum	V_{max}	$V \geq V_{\text{max}}$	Upper-bound charge signaling
Brown-out	V_{BO}	$V < V_{\text{BO}}$	MCU resets; all volatile state is lost

These thresholds satisfy $V_{\text{BO}} < V_{\text{OFF}} < V_{\text{ON}} \leq V_{\text{max}}$, establishing a hysteresis window that is essential to prevent the node from thrashing (continually booting and crashing) under low harvesting rates. Specifically, the hysteresis window between V_{ON} and V_{OFF} ensures that the node has accumulated a minimum quantum of energy to perform active tasks (such as transmitting a packet) before it boots. Meanwhile, the brownout threshold V_{BO} acts as a hardware safety boundary; if voltage drops below V_{BO}, RAM retention is no longer guaranteed, triggering a hard reset to protect the integrity of the volatile memory and ensure clean protocol re-initialisation.

When the capacitor voltage rises above V_{ON}, the MCU transitions to the ON state and begins executing the protocol. As the capacitor discharges through active operation, once V falls below V_{OFF}, the MCU transitions to the OFF state and enters a low-power sleep consuming E_{sleep} per slot. If the harvested power is insufficient to sustain even the sleep state, V continues to fall. Upon crossing V_{BO}, the MCU undergoes a hard reset: all RAM contents are cleared and the protocol state is discarded. The node then re-initialises from scratch upon the next V_{ON} crossing.

This power cycle defines the fundamental constraint of battery-free neighbour discovery: each active period is energy-bounded, and brown-out events irrecoverably discard accumulated protocol state.

C. BFND Protocol Design

BFND exploits passive RF envelope sensing to direct energy-intensive advertisements toward slots where neighbouring nodes are most likely to be active.

Time-Slotted Framework: Nodes operate under a time-slotted clock at frequency f_{clk}. Slots of duration $T_{\text{clk}} = 1/f_{\text{clk}}$ are grouped into nominal cycles of length T_{nom} slots. The current cycle offset is given by ASN mod T_{nom}, where ASN is the running slot counter. Each node maintains a slot offset S_{offset}, initialised randomly and re-randomised upon discovery.

Channel Activity Map: The node maintains a channel activity map $\mathbf{M}$ of length T_{nom}, where $M[i]$ records the successful scans observed at cycle slot i. When the node enters scan mode, it executes

$$k = \left\lfloor \frac{E_{\text{adv}}}{E_{\text{scan}}} \right\rfloor \tag{2}$$

consecutive scans in back-to-back slots. A successful scan at slot ASN increments the map:

$$M\left[\text{ASN} \bmod T_{\text{nom}}\right] \mathrel{+}= 1 \tag{3}$$

The map $\mathbf{M}$ persists across power cycles (ON/OFF transitions) and is only cleared during a brown-out reset (voltage below V_{BO}). Maintaining this channel activity map imposes minimal memory and computational overhead. For a typical nominal cycle length of $T_{\text{nom}} = 1000$ slots, storing byte-sized counters requires only 1 KB of RAM. This easily fits within the memory limits of modern low-power MCUs (such as the nRF52840, which features 256 KB of RAM) without impacting other application processes. The computational overhead is similarly negligible, restricted to simple array indexing and modulo arithmetic upon scan completions.

Advertisement Scheduling: When advertising, the node constructs the candidate slot set:

$$\mathcal{S} = \{S_{\text{offset}}\} \cup \{i : M[i] > 0\} \tag{4}$$

Given the current position $s = \text{ASN} \bmod T_{\text{nom}}$, the node schedules its advertisement at target slot S_{adv}:

$$S_{\text{adv}} = \arg\min_{s_i \in \mathcal{S}} \left[(s_i - s) \bmod T_{\text{nom}}\right], \quad (s_i - s) \bmod T_{\text{nom}} > 0 \tag{5}$$

The node sleeps until ASN reaches S_{adv}, then transmits. This focuses advertisements on slots with previously observed channel activity.

Protocol State Machine: The protocol operates as a three-state finite state machine (Figure 2). Upon transitioning to the ON state, the node enters the UNINITIALISED state and draws a Bernoulli random variable with parameter $\alpha \in (0, 1)$:

- With probability α: The node enters the ADVERTISEMENT state, schedules the next transmission at S_{adv}, sleeps until that slot, transmits, and returns to UNINITIALISED.
- With probability $1 - \alpha$: The node enters the SCAN state, executes k consecutive scans, updates $\mathbf{M}$ on successes, and returns to UNINITIALISED.

Reset and State Persistence: On any turn-off event ($V < V_{\text{OFF}}$), scheduled slots and remaining scans are discarded, and the FSM returns to UNINITIALISED. However, the channel map $\mathbf{M}$ persists across these cycles. It is only cleared on a hard brown-out reset ($V < V_{\text{BO}}$).

Advertisement Frame Structure: Drawing from [5], [14], each advertisement occupies a single slot (Figure 3). The advertising node transmits its advertisement frame, transitions to a receiving state, and reserves the end of the slot for an acknowledgement frame. This bidirectional handshake ensures that both nodes register the discovery event. BFND does not enforce network-wide slot synchronisation, which mitigates overlapping transmission collisions and increases discovery probability.

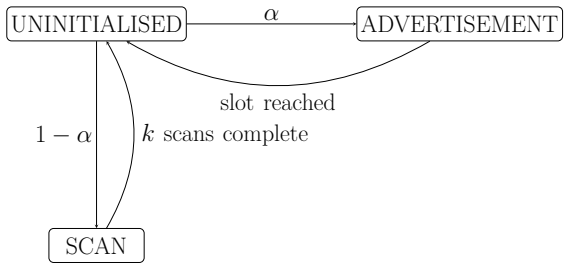


Fig. 2. BFND protocol finite state machine. Upon each turn-on event, the node probabilistically selects advertisement or scan mode and returns to UNINITIALISED upon completion.

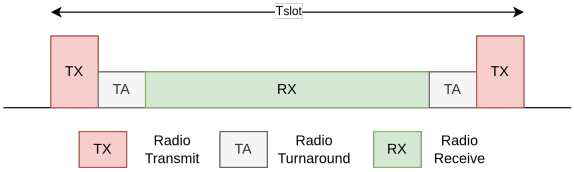


Fig. 3. Frame structure of a single advertisement slot specifying the sequence of operations.

IV. Experiments

To evaluate the performance of BFND against the baseline, we implement a custom discrete-event simulator in Python (drawing design patterns from sensor network simulation frameworks like SensorSim [22]) that models the interaction between continuous energy harvesting, capacitor state transitions, and asynchronous slot boundaries.

A. Simulation Environment and Channel Model

The simulator models a network of N battery-free devices operating in a star topology with a central node (node 0) and $N - 1$ leaf nodes. The discovery process is complete when the central node has successfully discovered all surrounding leaf nodes.

Channel and Interference Model: The wireless medium is modelled as a slotted collision channel. If multiple leaf nodes advertise concurrently, their transmissions collide at the central node. We model this multi-user interference (MUI) using a Signal-to-Interference-plus-Noise Ratio (SINR) formulation at the receiver. The SINR at a receiving node i (such as the central node) from a transmitting node j is:

$$\text{SINR}_i = \frac{P_{\text{tx}} \cdot g_{ji}}{\sigma^2 + \sum_{k \neq j} P_{\text{tx}} \cdot g_{ki}} \tag{6}$$

where P_{tx} is the transmission power, g is the channel gain (with a path-loss exponent of $\eta = 2$), and σ^2 is the thermal noise floor. If multiple nodes advertise concurrently,

the interference drops SINR_i below the demodulation threshold ($\gamma_{\text{th}} = 10$ dB), resulting in a packet collision. However, the passive RF-sensing envelope detector can still detect channel activity during collisions as long as the received signal power exceeds its sensitivity threshold ($P_{\text{sens}} = -50$ dBm), enabling nodes to build their channel activity maps even in the presence of collisions.

B. Simulator Implementation Details

Asynchronous Slot Modelling: The simulator models continuous-time asynchronous clock boundaries, clock drift δ_{drift}, and a random hardware wake-up jitter $J \in [-0.15, 0.15]$ slots. A packet transmission spans an interval of $T_{\text{pkt}} = 0.048$ slots (48 μs at 1 Mbps), and a scan action opens the receiver window for a full 1 ms slot. A successful scan occurs if the transmission interval strictly overlaps with the receiver's scan window, capturing the physical interaction of asynchronous signals.

Trace Discretisation: The simulator aggregates real-world continuous power traces (HDF5 format) from the Bonito dataset [16] into slot-level energy values. A trace records harvested power $P_{\text{harvest}}(\tau)$ in Watts at sampling resolution $\delta t = 10$ μs. The simulator aggregates these samples over a slot interval $\Delta t = 1/f_{\text{clock}} = 1$ ms:

$$E_{\text{in}}(k) = \sum_{j=k\cdot M}^{(k+1)M-1} P_{\text{harvest}}(j) \cdot \delta t \tag{7}$$

where k is the slot index, and $M = \Delta t/\delta t = 100$ is the number of trace samples per slot. Each simulation run starts at a random offset in the trace to ensure statistical independence.

C. Evaluated Protocols

Baseline Protocol (FIND) [14]: The baseline represents a standard intermittent protocol without RF sensing feedback. Upon waking, nodes sleep for a random delay before advertising. After transmitting, they enter an active busy-wait scanning state (consuming high energy $E_{\text{busy_wait}}$) until the capacitor voltage drops below V_{OFF}.

BFND Protocol: BFND uses the probabilistic state machine described in Section III-C. Depending on α, it performs low-power passive scanning to update its map $\mathbf{M}$ or sleeps until its intelligently scheduled advertisement slot S_{adv} to transmit.

D. Simulation Parameters

Table II details the parameters used across our simulations, which are modeled on the nRF52840 system-on-chip operating with a 3.0 V rail and an active DC/DC converter. The default capacitance of 17 μF represents the effective, voltage-derated capacitance of a ceramic capacitor with a nominal value of 47 μF operating at 3.0 V (a typical voltage derating curve for Multi-Layer Ceramic Capacitors under DC bias, as referenced in [16]).

We use the nRF52840 as our hardware baseline, accounting for its slightly higher sleep currents ($E_{\text{sleep}} \approx$ 14.4 nJ/slot). To maximize TX-RX overlap, we select an 800 μs receive window separated by two 48 μs transmissions, yielding a burst duration of 976 μs ($E_{adv} \approx 14\,\mu$J). For the passive scanning phase, the MCU wakes briefly to drive the external scan circuit via GPIO, drawing $P_{\text{scan}} \approx 11.3\,\mu$W [11]. The resulting per-slot scanning cost is $E_{\text{scan}} \approx 45$ nJ, reducing the energy footprint by orders of magnitude compared to active radio listening.

TABLE II
Simulation Configuration Parameters

Parameter	Description	Value
f_{clock}	Clock Frequency / Slot Rate	1000 Hz
C	Effective Capacitance	17 μF
V_{ON}	Turn-on Voltage Threshold	3.0 V
V_{OFF}	Turn-off Voltage Threshold	2.4 V
V_{max}	Maximum Voltage Threshold	3.3 V
V_{BO}	Brown-out Voltage Threshold	1.8 V
E_{adv}	Advertisement Energy Cost	14.0 μJ
E_{scan}	Passive Scanning Energy Cost	45 nJ
E_{sleep}	System Idle Sleep Energy per Slot	14.4 nJ
$E_{\text{busy_wait}}$	Active Scanning Energy per Slot	9.9 μJ
T_{nom}	Nominal Cycle Length	1000 slots
α	Advertisement Probability	0.7

E. Experimental Scenarios

We evaluate the performance and scalability of the protocols under two main sweep configurations, running 1000 independent simulations per setting:

1) Scenario Harvesting Profiles: We run simulations under five time-synchronised energy-harvesting traces from the Bonito dataset [16]: Office (stable indoor RF), Stairs (abundant dynamic energy), Washer (moderate domestic vibrations), Jogging (high-intensity movement), and Cars (highly transient road traffic RF).
2) Network Density (N): We sweep the node count $N \in \{2, 3, 4, 5\}$ to study how both protocols scale and handle packet collisions under varying densities.

Each run is simulated for up to 5000 nominal cycles ($5,000,000$ slots) to determine the discovery latency (in slots) and the success rate.

V. Results

In this section, we present the results from our performance evaluation of the proposed BFND protocol against the FIND baseline. We analyse discovery latency (in absolute slot numbers, ASN) and timeout rates across five real-world harvesting environments and network densities $N \in \{2, 3, 4, 5\}$. Each configuration runs 1000 independent simulation instances with a timeout limit of $5,000,000$ slots (5000 nominal cycles).

A. Default Configuration Performance

We first evaluate the protocols under the default simulation setup representing a typical IoT cluster: $N = 5$ nodes, capacitor $C = 17$ μF (voltage-derated from a nominal 47 μF capacitor), advertisement probability $\alpha = 0.7$, and trace-based Office harvesting.

BFND achieves a median discovery latency of $9,054$ slots, compared to $55,925$ slots for the baseline protocol, representing a $6.2\times$ speedup. Furthermore, both protocols completed successfully in all 1000 runs (0% timeout rate).

Figure 4 (left) presents a box plot summarising the discovery latency distribution for successful runs. The BFND protocol consistently achieves lower discovery latency with minimal variance compared to the wider spread and tail of the baseline. The Cumulative Distribution Function (CDF) plot in Figure 4 (right) highlights the higher probability of BFND achieving discovery by any given ASN. BFND achieves discovery in nearly all runs well before the ASN where the baseline begins to complete, demonstrating superior reliability and predictability.

This default performance highlights the physical differences in energy management: the active channel listening state (ACTION.BUSY_WAIT) of the baseline repeatedly drains the node's capacitor, causing it to brown out frequently, reset its state, and miss advertisements from neighbouring nodes. In contrast, BFND preserves charge using low-power sleep and passive envelope-detector scanning, allowing it to build its channel activity map $\mathbf{M}$ and schedule advertisements to rendezvous efficiently.

B. Impact of Advertisement Probability (α)

The advertisement probability α dictates the proportion of charge cycles a node dedicates to transmitting advertisements versus scanning the medium. We sweep $\alpha \in \{0.5, 0.7, 0.9\}$ under default configurations for $N = 2$ nodes:

Discussion: Figure 5 demonstrates that $\alpha = 0.5$ yields the lowest median discovery latency, though the performance gap between $\alpha = 0.5$ and $\alpha = 0.9$ narrows significantly as the available harvested energy decreases. In high-energy scenarios (Stairs and Office), frequent node activations guarantee that a scanning phase quickly captures a neighbor's advertisement. Consequently, an equal split between scanning and advertising ($\alpha = 0.5$) optimally minimizes latency, whereas overwhelmingly broadcasting ($\alpha = 0.9$) needlessly starves the critical schedule-learning phase. Conversely, in low- and medium-energy scenarios (Cars, Jogging, Washer), nodes activate infrequently, which drastically reduces the probability that a scanning window temporally coincides with a neighbor's advertisement. Under these energy-constrained conditions, the necessity of performing more advertisements increases to ensure that a transmission is actually present during a neighbor's scanning phase. However, over-advertising also leads to frequent misalignment and missed discovery opportunities. This is evidenced by the increased timeout percentage observed in the lowest-energy dataset (Cars) at $\alpha = 0.9$. Hence, regardless of the energy scenario, these results validate our hypothesis that introducing ultra-low-power scanning significantly improves discovery latency compared to a greedy, transmission-heavy approach.

Beyond α, performance depends on the nominal cycle length (T_{nom}) and map size. A longer cycle mitigates collisions but increases absolute latency, delaying convergence. Furthermore, in non-stationary environments with node mobility, past channel observations may become stale. Robustness could be maintained by adapting $\mathbf{M}$ with a time-decay factor or sliding window to prevent obsolete data from skewing schedules.

C. Network Density and Environmental Scenarios

We evaluate the scalability of both protocols across five real-world energy-harvesting traces: Office (stable indoor RF), Stairs (abundant dynamic kinetic energy), Washer (moderate domestic vibrations), Jogging (high-intensity human movement), and Cars (highly transient road traffic RF). We sweep the node count $N \in \{2, 3, 4, 5\}$ and summarise the median discovery latency and timeout rates in Table III.

Scalability and Collision Avoidance: The results highlight that BFND scales exceptionally well as node density increases. For instance, under the Jogging trace, increasing the node count from $N = 2$ to $N = 5$ results in a modest increase in median latency for BFND ($10,152$ to $23,995$ slots) with a consistent 0% timeout rate. In contrast, the baseline's median latency increases from $43,754$ to $638,358$ slots, with its timeout rate climbing to 4.2%.

This scalability advantage stems from BFND's collision avoidance. While FIND nodes transmit randomly, causing frequent multi-user collisions at the central node, BFND nodes construct a shared slot schedule using their channel activity maps $\mathbf{M}$. This allows nodes to coordinate their active slots, avoiding collisions and drastically reducing discovery times.

Behaviour in Abundant Energy (Stairs Trace): In the Stairs trace, ambient energy is abundant. Under these conditions, nodes require very few slots to transition from V_{OFF} to V_{ON}. Under the default $\alpha = 0.7$ configuration, the FIND baseline achieves slightly lower median latencies than BFND at low node densities (e.g., $2,538$ vs. $3,926$ slots at $N = 2$, and $5,418$ vs. $6,678$ slots at $N = 3$). This performance gap stems primarily from the cyclic nature of the BFND protocol. In high-energy scenarios, even when a node harvests sufficient energy for frequent advertisements, BFND schedules transmissions only during designated slots, which increases discovery latency.

However, parameter tuning easily mitigates this minor baseline advantage. Specifically, reducing the nominal cycle length to 250 slots in the Stairs trace enables BFND to close the gap. In this configuration, BFND achieves median latencies of $1,199$ slots at $N = 2$ and $2,032$ slots at $N = 3$, outperforming FIND's median discovery latency while maintaining tighter latency bounds and lower variance.

Failure Mechanism under Severe Scarcity (Washer, Jogging, Cars Traces): Under severe energy scarcity (Washer and Cars traces), the baseline protocol fails to discover

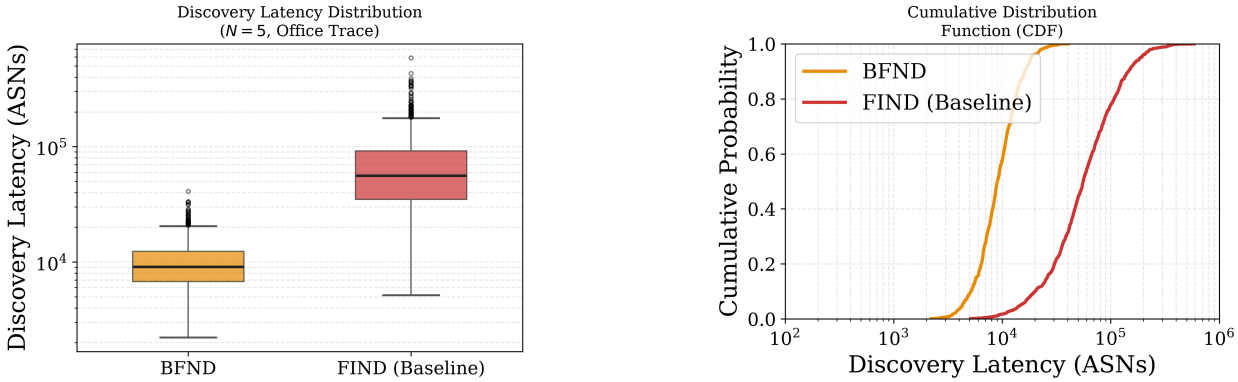


Fig. 4. Discovery latency distribution (left) and Cumulative Distribution Function (CDF) of discovery latency (right) for successful runs under the default configuration ($N = 5$, $C = 17$ μF, $\alpha = 0.7$, Office trace).

TABLE III
Neighbour Discovery Latency and Timeout Rates across Network Densities (N) and Environmental Traces (1000 Runs per Configuration)

Trace Scenario	Nodes (N)	Median Latency* (slots)		Timeout Rate (%)	
		BFND	FIND	BFND	FIND
Cars (Damped Vibrations)	2	301,527	467,903	7.3%	98.7%
	3	896,096	2,604,744	26.4%	99.8%
	4	3,032,740	–	44.0%	100.0%
	5	3,166,578	–	59.5%	100.0%
Jogging (High Intensity Vibrations)	2	10,152	43,754	0.0%	0.9%
	3	13,732	101,180	0.0%	2.9%
	4	17,364	275,505	0.0%	2.9%
	5	23,995	638,358	0.0%	4.2%
Office (Indoor Solar)	2	4,742	16,714	0.0%	0.0%
	3	7,091	34,393	0.0%	0.0%
	4	8,290	44,678	0.0%	0.0%
	5	9,054	55,925	0.0%	0.0%
Stairs (Outdoor Solar)	2	3,926	2,538	0.0%	0.0%
	3	6,678	5,418	0.0%	0.0%
	4	7,696	6,806	0.0%	0.0%
	5	8,848	8,071	0.0%	0.0%
Washer (Domestic Vibration)	2	62,246	1,690,985	0.0%	1.8%
	3	879,869	1,694,876	0.0%	2.2%
	4	882,444	1,576,339	0.0%	4.9%
	5	884,542	1,232,402	0.0%	6.9%

*Calculated over successful (non-timeout) runs.

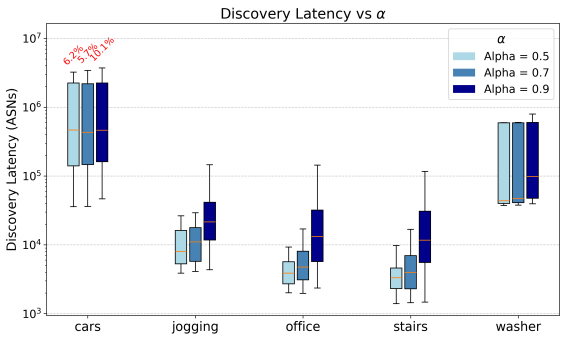


Fig. 5. Impact of α on discovery latency across different datasets. The timeout percentage (if non-zero) is shown on top of each bar.

neighbours within the allocated time. For $N = 5$ in the Cars trace (representing highly transient road traffic RF harvesting), the baseline suffers a 100% timeout rate across all 1000 runs, whereas BFND maintains a 40.5% success rate with a median latency of $3,166,578$ slots for successful runs. Under the Jogging trace, BFND is over $26\times$ faster than the baseline at $N = 5$ ($23,995$ vs. $638,358$ slots).

This significant performance gap is explained by the physical energy consumption of the protocols. When a FIND node boots, its active busy-wait scanning state (ACTION.BUSY_WAIT) consumes 9.9 μJ per slot. Under weak harvesting, this high active drain instantly depletes the storage capacitor, dropping the voltage below V_{BO} and triggering a hard brownout reset. This reset erases all volatile state, forcing the node into a loop of booting, discharging, and resetting.

Conversely, BFND uses low-power sleep (14.4 nJ) to preserve charge and passive envelope detector scanning (45 nJ) to map the channel. This minimal energy footprint allows nodes to remain ON longer, avoid brownout resets, retain their persistent channel maps, and successfully coordinate active slots for rendezvous.

VI. Conclusions

In this paper, we addressed the critical challenge of neighbour discovery in battery-free IoT networks operating under intermittent power constraints. We proposed BFND, a protocol that leverages ultra-low-power passive RF information harvesting to sense channel occupancy and construct a local channel activity map. This map allows nodes to intelligently schedule advertisements during high-activity slots, matching the wake-up times of neighbouring nodes.

We evaluated BFND against FIND, a random-duty-cycle baseline using a custom discrete-event simulator configured with parameters derived from hardware models. Under stable indoor harvesting (Office trace at $N = 5$), BFND reduces the median discovery latency by over $6.2\times$ ($9,054$ slots vs. $55,925$ slots). Under high-intensity harvesting scarcity (Jogging trace at $N = 5$), BFND is over $26\times$ faster than the baseline ($23,995$ vs. $638,358$ slots) and eliminates timeouts (0% vs. 4.2%). Under severe transient road traffic harvesting (Cars trace at $N = 5$), the baseline fails completely with a 100% timeout rate, whereas BFND still maintains a 40.5% success rate with a median latency of $3,166,578$ slots for successful runs. Our findings demonstrate that integrating low-power environmental sensing, such as RF information harvesting, is a promising and reliable path for designing communication protocols in energy-intermittent computing systems.

Future work involves translating our simulation findings to a physical hardware testbed. We plan to build hardware prototypes of the analogue envelope-detectors to validate that the assumed RF sensing, timing behaviour, and energy models accurately reflect practical deployments. Additionally, we will explore lightweight, online slot-selection algorithms running on intermittent MCUs to dynamically predict activity slot variations and evaluate the protocol under node mobility.

Acknowledgement

This work was supported by the Swedish Foundation for Strategic Research (SSF) under the project ZeroIoT: Enabling the Battery-free Internet of Things (grant CHI19-0003).

References

[1] W. Augenstein et al., "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861, Tech. Rep., September 2007.

[2] M. Bakht, M. Trowbridge, and P. Mohapatra, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1448–1459, 2014.

[3] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in SenSys, 2008.

[4] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in Proc. ACM/IEEE IPSN, ser. IPSN '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 350–361. [Online]. Available: https://doi.org/10.1145/1791212.1791253

[5] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in Proc. ACM MobiCom, ser. Mobicom '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 185–196. [Online]. Available: https://doi.org/10.1145/2348543.2348568

[6] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," IEEE Communications Surveys & Tutorials, vol. 13, no. 3, pp. 443–461, 2010.

[7] A. Colin, E. Ruppel, and B. Lucia, "Intermittent computation without hardware support or programmer intervention," in Proc. USENIX OSDI, 2016, pp. 17–32.

[8] A. Torrisi, D. Brunelli, and K. S. Yildirim, "Amalgamated intermittent computing systems," in Proc. ACM/IEEE IoTDI, 2023, pp. 112–124.

[9] E. Longman, M. El-Hajjar, and G. V. Merrett, "Multihop networking for intermittent devices," in Proc. ACM ENSsys, 2022, pp. 31–36.

[10] N. H. Hokke et al., "RF Information Harvesting for Medium Access in Event-driven Batteryless Sensing," in IPSN, 2022.

[11] G. Liu, T. Ma, W. Yin, J. Zhang, H. Gao, and B. Wu, "A 2.4 ghz band highly sensitive low power wake-up receiver," in Proc. IEEE Ucom. IEEE, 2023, pp. 116–120.

[12] C. Noda, S. Prabh, M. Alves, C. A. Boano, and T. Voigt, "Quantifying the channel quality for interference-aware wireless sensor networks," ACM SIGBED Review, 2011.

[13] 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, ser. MobiHoc '01. New York, NY, USA: Association for Computing Machinery, 2001, p. 137–145. [Online]. Available: https://doi.org/10.1145/501431.501435

[14] K. Geissdoerfer and M. Zimmerling, "Bootstrapping battery-free wireless networks: Efficient neighbor discovery and synchronization in the face of intermittency," in Proc. USENIX NSDI. USENIX Association, Apr. 2021, pp. 439–455. [Online]. Available: https://www.usenix.org/conference/nsdi21/presentation/geissdoerfer

[15] R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, "Panda: Neighbor discovery on a power harvesting budget," IEEE Journal on Selected Areas in Communications, vol. 34, no. 12, pp. 3606–3619, 2016.

[16] K. Geissdoerfer and M. Zimmerling, "Learning to communicate effectively between battery-free devices," in Proc. USENIX NSDI, 2022, pp. 419–435.

[17] J. de Winkel, H. Tang, and P. Pawełczak, "Intermittently-powered bluetooth that works," in Proc. ACM MobiSys, ser. MobiSys '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 287–301. [Online]. Available: https://doi.org/10.1145/3498361.3538934

[18] S. Mosavat, J. Göpfert, B.-C. Renner, P. J. Marrón, and M. Zella, "Cato: Wake-up receiver-based communication for batteryless devices," Sensors, vol. 25, no. 22, p. 6813, 2025.

[19] G. Liu, V. Nigade, H. Bal, and L. Wang, "A little certainty is all we need: Discovery and synchronization acceleration in battery-free iot," in Proc. ACM APNet, 2024, pp. 38–44.

[20] G. Liu, K. S. Yıldırım, and L. Wang, "Freebeacon: Efficient communication and data aggregation in battery-free iot," IEEE Transactions on Mobile Computing, 2025.

[21] M. L. Wymore and D. Qiao, "River-mac: A receiver-initiated asynchronously duty-cycled mac protocol for the internet of things," in Proc. IEEE COMPSAC, vol. 1. IEEE, 2019, pp. 860–869.

[22] S. Park, A. Savvides, and M. B. Srivastava, "SensorSim: A Simulation Framework for Sensor Networks," in Proc. ACM MSWiM, 2000, pp. 104–111.