Learning to Communicate Effectively Between Battery-free Devices

Kai Geissdoerfer and Marco Zimmerling, *TU Dresden*

https://www.usenix.org/conference/nsdi22/presentation/geissdoerfer

This paper is included in the Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation.

April 4–6, 2022 • Renton, WA, USA

978-1-939133-27-4

Open access to the Proceedings of the 19th USENIX Symposium on Networked Systems Design and Implementation is sponsored by

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Learning to Communicate Effectively Between Battery-free Devices

Kai Geissdoerfer
TU Dresden

Marco Zimmerling
TU Dresden

Abstract

Successful wireless communication requires that sender and receiver are operational at the same time. This requirement is difficult to satisfy in battery-free networks, where the energy harvested from ambient sources varies across time and space and is often too weak to continuously power the devices. We present Bonito, the first connection protocol for battery-free systems that enables reliable and efficient bi-directional communication between intermittently powered nodes. We collect and analyze real-world energy-harvesting traces from five diverse scenarios involving solar panels and piezoelectric harvesters, and find that the nodes' charging times approximately follow well-known distributions. Bonito learns a model of these distributions online and adapts the nodes' wake-up times so that sender and receiver are operational at the same time, enabling successful communication. Experiments with battery-free prototype nodes built from off-the-shelf hardware components demonstrate that our design improves the average throughput by 10–80× compared with the state of the art.

1 Introduction

The last few years have seen rapid innovation in battery-free systems [40], culminating in a number of real-world applications [1, 12, 27]. These systems pave the way toward a more sustainable Internet of Things (IoT) [7] by enabling small, cheap, and lightweight devices to perform complex tasks (e.g., DNN inference [20]) off ambient energy while using tiny, environmentally friendly capacitors as energy storage [40]. However, to replace today's trillions of battery-powered IoT nodes, battery-free devices must learn to communicate.

Challenge. The power that can be harvested from solar, vibrations, or radio signals is typically insufficient to continuously operate a device. A traditional energy-neutral device buffers harvested energy in a rechargeable battery and can *freely control* its average duty cycle to avoid power failures. Instead, a battery-free device cannot avoid power failures, and has *very limited control* over when the power failures begin and end. Fig. 1 illustrates this so-called *intermittent operation*. After executing for a short time, a battery-free device is *forced* to become inactive and wait for a long, fluctuating time until its capacitor is sufficiently charged again. For example, when harvesting energy from indoor light, our prototype battery-free nodes need to stay off and recharge, on average, for hundreds of milliseconds before they can operate for at most 1 ms.

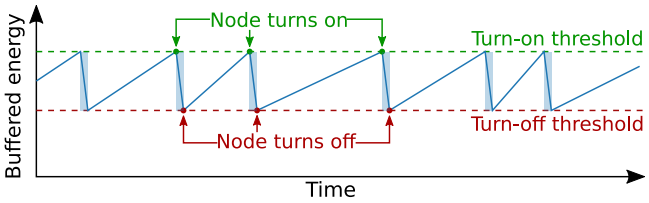
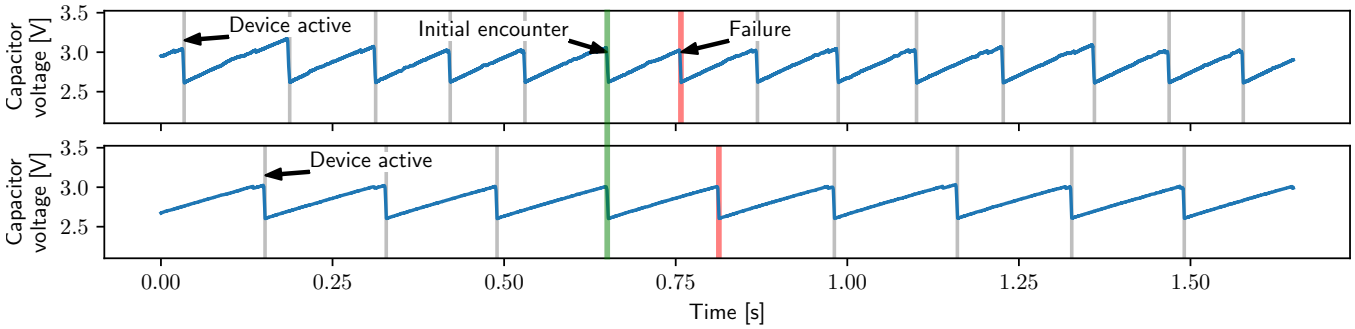


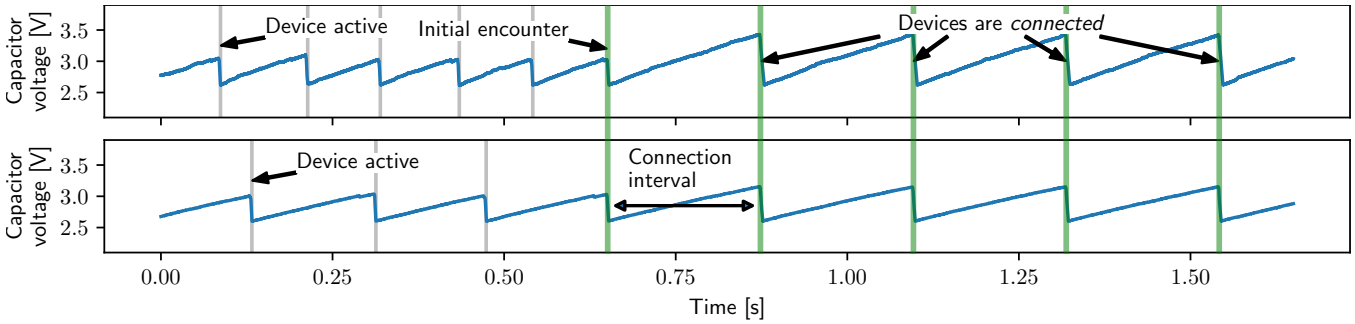
Figure 1: Because ambient power is often weak, a battery-free node must buffer energy before it can wake up and operate for a short time period. This is known as intermittent operation.

Many techniques have been developed to deal with intermittency on a *single* battery-free device [6, 11, 34], but how to communicate *between* intermittently powered devices is one of the most pressing problems yet to be solved [22, 28, 48]. This is due to the fact that device-to-device communication is a fundamental building block for a variety of network and system services, including optimal clock synchronization [26], ranging and localization [9, 21], sensor calibration [41], distribution and coordination of sensing and computing tasks [32], collaborative learning [47], and efficient and reliable wireless networking [25]. Realizing these services across battery-free devices has the potential to enable novel and more sustainable IoT and sensor network applications, from automatic contact tracing to planetary-scale environmental monitoring.

To be able to communicate, sender and receiver must be active and have enough energy for at least one complete packet transmission *at the same time*. However, since the nodes' activity phases are generally interleaved and short compared to their charging times, as visible from the real-world trace in Fig. 2a, it often takes thousands of wake-ups until two nodes encounter each other and communication becomes possible [19]. Moreover, after an encounter, the nodes quickly get out of sync if they become active immediately after a



(a) Because of their short and interleaved activity phases, battery-free devices often need a long time with hundreds of wake-ups until they encounter each other. Even after an initial encounter, the devices quickly get out of sync, rendering communication inefficient and unreliable.



(b) With Bonito, devices learn and exchange statistical models of their charging times and agree on a connection interval that ensures that both devices have sufficient energy at the same time. Maintaining a connection over multiple encounters enables efficient and timely communication.

Figure 2: The challenge of efficient battery-free device-to-device communication in (a) and our proposed protocol in (b).

recharge, as stipulated by the state of the art [8, 33] and apparent in Fig. 2a. This is because ambient energy varies across time and space [3], which leads to fluctuating and different charging times between the nodes.

Besides establishing a first encounter [19], active radio communication has been considered too demanding for battery-free devices [36]. Conversely, work on backscatter communication has focused on physical-layer issues, such as improving range and throughput, purposely considering high-energy environments, batteries, or cables to continuously power the devices in the experiments to avoid intermittency [29, 35, 38, 49]. However, when running off ambient energy, duty cycling of the backscatter transceivers becomes necessary [14, 29, 43]—and, without a battery, the intermittency problem occurs.

Contribution. This paper presents Bonito, the first connection protocol for battery-free wireless networks. Bonito provides reliable and efficient bi-directional communication despite the time-varying intermittency of battery-free devices.

The real-world trace in Fig. 2b illustrates the high-level protocol operation. Unlike the state of the art, Bonito enables two battery-free nodes, after an initial encounter, to maintain a *connection* across multiple consecutive encounters. To this end, Bonito continually adapts the *connection interval*, which is the time between the end of an encounter and the beginning of the next encounter. A shorter connection interval provides more communication opportunities in the long run. However, a connection interval that is shorter than any of the nodes' charging times breaks the connection and requires the nodes to wait for a long time until they encounter each other again. Thus, the challenge is to keep the connection interval as short as possible without losing the connection, which is difficult in the face of time-varying charging times.

One of our key insights is that, depending on the scenario and energy-harvesting modality, the charging time of a battery-free node approximately follows well-known probability distributions. We leverage this insight in Bonito by letting each node *continuously learn and track* the parameters of a model that approximates the distribution of locally observed charging times against non-stationary effects (e.g., changes in mean or variance). Then, to maintain an efficient and reliable connection, the nodes exchange at every encounter their current model parameters and jointly adapt the connection interval.

We implement Bonito on a custom-designed ultra low-power battery-free node. Our prototype is built from off-the-shelf components, including an ARM Cortex-M4 microcontroller that features a 2.4 GHz Bluetooth Low Energy (BLE) radio. The node harvests energy from a solar panel or a piezoelectric harvester, using a 47 µF capacitor as energy storage.

To evaluate Bonito through testbed experiments and fairly compare it against two baselines, we use up to 6 Shepherd ob-

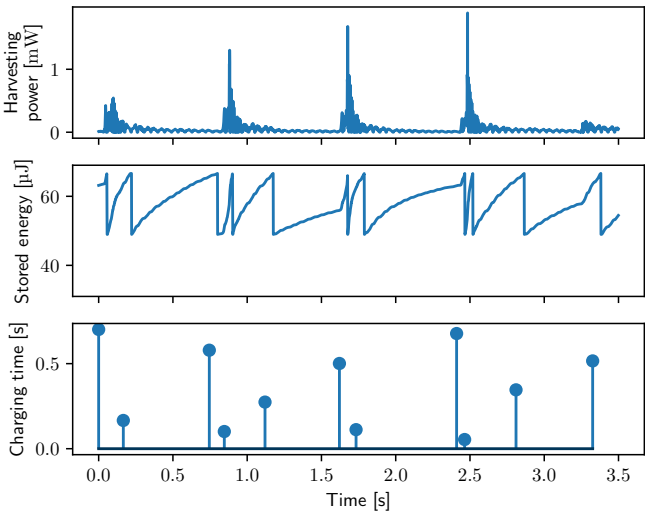
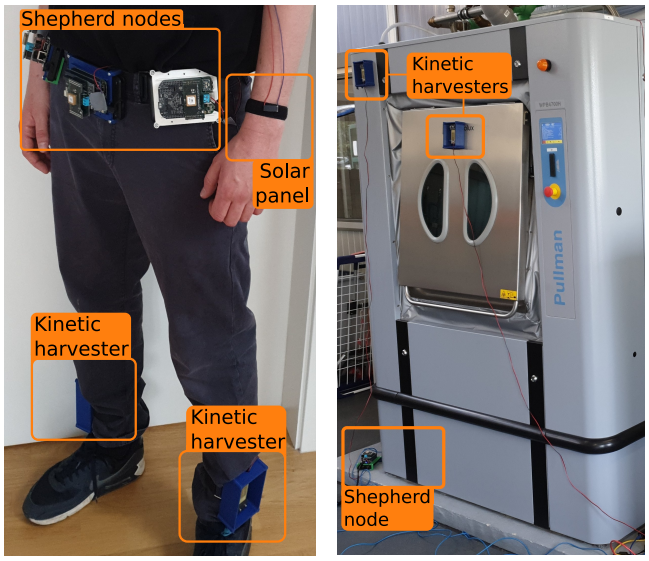


Figure 3: The top plot shows an example trace of real kinetic harvesting power during jogging (see picture in Fig. 4a). The middle and bottom plots show the corresponding energy stored in the capacitor and the resulting charging times of a simulated battery-free device.



(a) Runner with full measurement setup for the *jogging* dataset.

(b) Washing machine with partial setup for the *washer* dataset.

Figure 4: Pictures from two of the five scenarios in which we use synchronized Shepherd nodes [18] to record energy-harvesting traces.

servers [18] to record and replay real-world energy-harvesting traces from 5 diverse scenarios. Our results show, for example, that Bonito maintains connections for hundreds of consecutive encounters, and that it outperforms the state of the art by 10–80× in terms of throughput. We also conduct a case study that demonstrates the utility of Bonito for accurate and timely occupancy monitoring in homes and commercial buildings.

Overall, this paper contributes the following:

- We collect 32 h of energy-harvesting traces from 5 different scenarios. Our analysis of these traces provides new insights into spatio-temporal intermittency patterns.
- We design the Bonito protocol. Bonito enables, for the first time, reliable and efficient communication between intermittently powered battery-free devices.
- We demonstrate an efficient implementation of Bonito on a prototype node with a 3.1 mm^3 ceramic capacitor.
- Results from testbed experiments and an occupancy monitoring case study provide evidence that Bonito performs well under a diverse range of real-world conditions.

2 Motivation

While previous work on intermittency has focused on individual battery-free devices [6,11,33] or discovery of neighboring devices [19], reliable and efficient device-to-device communication is still an open challenge. By *device-to-device communication* we mean the regular exchange of application data between two battery-free devices after they have successfully discovered each other through a first encounter [19].

To motivate the need for our work, we consider the scenario of battery-free wearables. Fig. 3 shows real-world data from a piezoelectric energy harvester that is attached to the ankle of a person (see Fig. 4a). The upper plot shows the harvesting power while the person is jogging, recorded by a Shepherd node. Shepherd is a measurement tool that records time-synchronized voltage and current traces from one or more energy-harvesting nodes with high rate and resolution [18]. The power spikes correspond to when the foot strikes the ground, with significantly lower harvesting power during the rest of the stride cycle. Based on trace-driven simulations, the middle plot shows the corresponding amount of harvested energy stored in an ideal 17 µF capacitor powering a battery-free device that turns on when the capacitor voltage exceeds 3 V and turns off when the capacitor voltage falls below 2 V. We see that when the device powers up, the stored energy is quickly consumed, forcing it to turn off already after about 1 ms. While powered off the harvesting power exceeds the standby power, so energy is accumulated and the capacitor voltage rises again. Compared to the short activity phases, the time needed to charge the capacitor, shown in the bottom plot of Fig. 3, is much longer and varies significantly over time.

The variability of a node's charging time is a function of its location and the associated energy environment, that is, how much power the harvester delivers at any given time. Thus, two battery-free devices, even when they are physically close to each other, have a different energy environment and therefore experience different charging times.

As an example, Fig. 5 plots the charging times of two devices during jogging over one hour. One device is powered by a piezoelectric harvester attached to the left ankle of a person, while the other device is powered by the same type of harvester attached to the right ankle of the person (see Fig. 4a). Each point in Fig. 5 indicates the charging times of both de-

Dataset	Energy Source	Harvester Part Number	Duration	#Devices	#Links	#Wake-ups	Model
Jogging	Human motion	MIDE S128-J1FR-1808YB	1 h	3	10	13252	Exponential
	Outdoor solar	IXYS KXOB25-05X3F		2		119127	Normal
Stairs	Outdoor solar	IXYS KXOB25-05X3F	1 h	6	15	359002	Normal
Office	Indoor light	IXYS SM141K06L	1 h	5	10	98324	Gaussian mixture
Cars	Car vibrations	MIDE S128-J1FR-1808YB	2 h	6	15	8517	Exponential
Washer	Machine vibrations	MIDE S128-J1FR-1808YB	45 min	5	10	22224	Normal

Table 1: Overview of energy-harvesting datasets we record in a variety of scenarios.

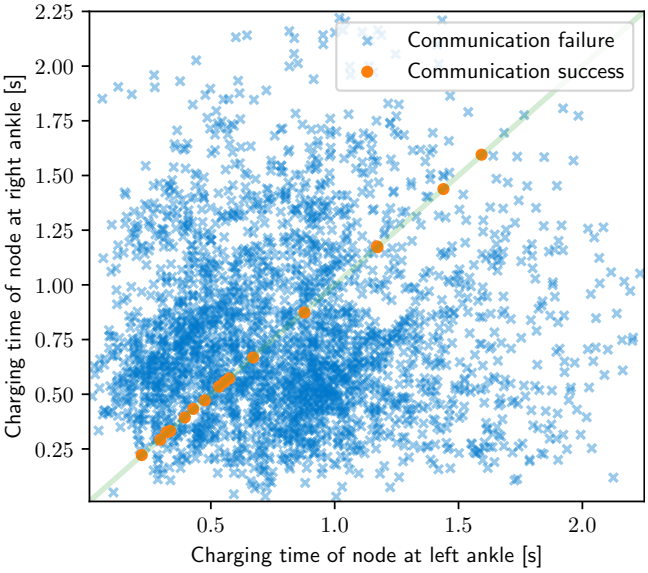


Figure 5: Charging times of two battery-free devices powered by kinetic harvesters attached to a jogger's ankles (see Fig. 4a). Using the greedy approach, the devices communicate successfully only in 0.04 % of the cases in which the charging times are almost identical.

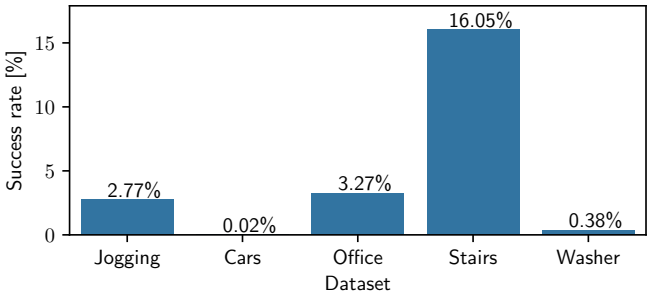


Figure 6: Success rate of greedy approach in trace-driven simulations, averaged across all pairs of devices (i.e., links) in a given dataset.

vices when they begin to charge their 17 µF capacitors at the same time from the same initial charge. We observe that in many instances the two nodes have vastly different charging times. This means that if nodes become active as soon as they reach the turn-on threshold, which is the state-of-the-art approach, called *greedy* and illustrated in Fig. 2a, the nodes often wake up with an offset that prevents communication, despite a successful encounter at the previous wake-up. Indeed, the success rate for the two nodes in Fig. 5 is less than 0.04 %. This leads to poor communication reliability and efficiency as the nodes more often than not fail to exchange their data.

To assess the generality of these observations, we record distributed energy-harvesting traces in diverse scenarios using multiple Shepherd nodes [18]. Table 1 lists the main characteristics of the five datasets we collected:

- The full *jogging* dataset comprises traces from two participants, each equipped with two piezoelectric harvester at the ankles and a solar panel at the left wrist (see Fig. 4a). The two participants run together for an hour in a public park, including short walking and standing breaks.
- For the *stairs* dataset, we recorded traces from six solar panels that are embedded into the surface of an outdoor stair in front of a lecture hall. Over the course of one hour, numerous students pass the stairs, leading to temporary shadowing effects on some or all of the solar panels.
- The *office* dataset comprises traces from five solar panels mounted on the doorframe and walls of an office with fluorescent lights. During the one-hour recording, people enter and leave the office and operate the lights.
- The *cars* dataset contains traces from two cars. Each car is equipped with three piezoelectric harvesters mounted on the windshield, the dashboard, and in the trunk. The cars drive for two hours in convoy over a variety of roads.
- The *washer* dataset includes five traces from piezoelectric harvesters mounted on a WPB4700H industrial washing machine, as shown in Fig. 4b, while the machine runs a washing program with maximum load for 45 min.

Fig. 6 plots for each dataset the average success rate across all pairs of devices (i.e., communication links) in the scenario. Even in the most favorable scenario, *stairs*, where the solar panels receive a fairly constant and similar energy input from natural sunlight, we find that the greedy approach succeeds in only 16 % of the cases. In all other scenarios, the success rate ranges below 3.5 %. Our experiments on real battery-free nodes in Sec. 5 confirm these trace-driven simulation results.

3 The Bonito Protocol

This section describes the Bonito protocol. The Bonito protocol enables two battery-free devices to stay connected after a first encounter, which can happen either coincidentally or with the support of a neighbor discovery protocol [19].

3.1 Overview

Bonito aims to make nodes repeatedly encounter each other so they can exchange application data reliably and efficiently, as shown in Fig. 2b. To ensure that nodes wake-up with a time offset small enough for a successful encounter, they agree at every encounter on a new *connection interval* T_C. This is the time between the end of the current encounter and the beginning of the next (i.e., planned) encounter.

Main idea and approach. For two nodes i and j with known charging times c_i and c_j, the shortest possible connection interval T_C^* is simply the maximum of their charging times

$$T_C^* = max(c_i, c_j) \tag{1}$$

If a shorter connection interval $T_C < T_C^*$ is used, then one node does not reach the required energy level to become active by T_C. Thus, the encounter fails, preventing the nodes from agreeing on the next connection interval—the connection is *lost*. A lost connection entails that the nodes often need to wait for a long time until they encounter each other again to resume communication. However, choosing a longer connection interval $T_C > T_C^*$ to mitigate the risk of a lost connection adds unnecessary delay as nodes, after having reached the required energy level, are forced to wait before they wake up at T_C.

The key challenge is to determine the connection interval T_C such that both nodes have enough energy while introducing only minimal delay. This is difficult as the charging times c_i and c_j are unknown and time-varying, as discussed in Sec. 2.

Using a probabilistic approach, we address this problem as follows. Let p be the probability that nodes i and j have sufficient energy to become active after a connection interval T_C. This corresponds to the probability that the nodes' charging times, c_i and c_j, are shorter than the connection interval T_C. Modeling c_i and c_j as random variables with a strictly monotonically increasing joint cumulative distribution function (cdf) $F_{i,j}$, this translates into

$$p = F_{i,j}(c_i = T_C, c_j = T_C) \tag{2}$$

Solving for T_C yields the minimum connection interval that guarantees, with a user-defined probability p, a successful encounter of the two nodes at their next wake-up

$$T_C = F_{i,j}^{-1}(p) \tag{3}$$

where $F_{i,j}^{-1}$ is the inverse joint cdf of c_i and c_j.

Base protocol. In practice, the joint cdf $F_{i,j}$ is rarely known a priori. Moreover, $F_{i,j}$ can only be estimated online by the nodes based on full knowledge of each other's charging times. Unfortunately, this requires frequent communication between battery-free nodes—precisely what Bonito intends to enable.

To circumvent this chicken-and-egg problem, we assume that the charging times, c_i and c_j, are statistically independent. In this case, the joint cdf $F_{i,j}$ is the product of the marginal cdfs F_i and F_j. The marginal cdfs can be estimated locally by each node from observations of their own charging times.

Based on these insights, we propose the following main steps of the Bonito protocol:

1. Each node i continuously estimates the marginal cdf F_i of its charging time based on local measurements.
2. When two nodes i and j encounter each other, they exchange their current estimates of F_i and F_j.
3. Using the same inputs (i.e., the marginal cdfs F_i and F_j and the user-defined probability p), both nodes compute the same new connection interval T_C according to (3).
4. Both nodes become active and communicate after the new connection interval T_C, and continue with step 2.

In this way, Bonito adapts the connection interval to changes in the energy environment, effectively enabling battery-free nodes to stay connected across several hundreds of subsequent encounters, as demonstrated by our experiments in Sec. 5.

To achieve this performance, we first need to answer the following key questions in our design of Bonito:

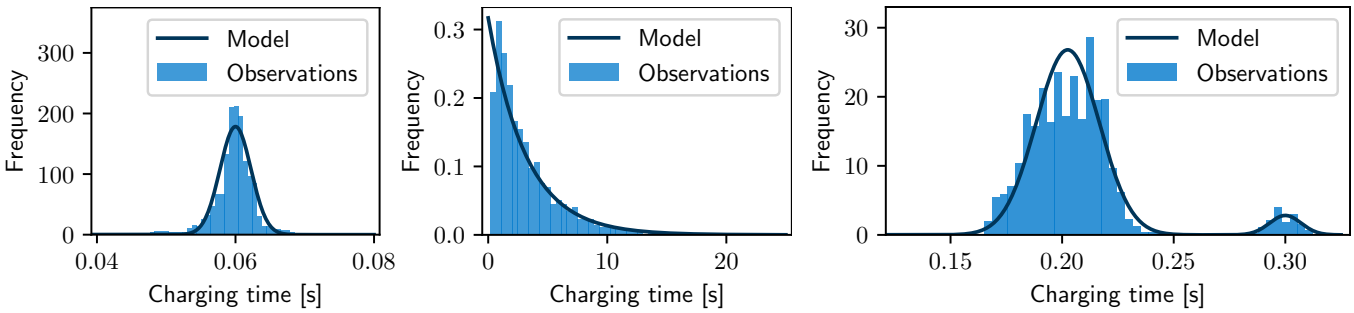
- How to compactly represent and exchange the marginal cdfs F_i and F_j in the face of limited energy (Sec. 3.2)?
- How to learn and track online an accurate estimate of F_i against a changing energy environment? (Sec. 3.3)
- How to efficiently compute the inverse joint cdf $F_{i,j}^{-1}(p)$ to obtain the connection interval T_C? (Sec. 3.4)

3.2 Modeling Charging Time Distributions

Because of the small energy storage, battery-free devices can only exchange a limited amount of data during an encounter. Thus, the marginal cdfs F_i and F_j must be represented in a compact form in order to be able to exchange them.

Unlike the common belief that the duration of a recharge is completely random [11, 31], we make the empirical observation that, in the scenarios we considered, the nodes' charging times can be faithfully modeled by well-known distributions. The rightmost column of Table 1 lists the models we use for each dataset. To illustrate, Fig. 7 plots representative charging time distributions and the corresponding models for the stairs, cars, and office datasets. Non-stationary effects like a time-varying mean are removed in the plots as these are effectively handled by our online learning approach detailed in Sec. 3.3.

We observe in Fig. 7a that when harvesting energy from outdoor solar with a constant harvesting voltage, the charging time can be modeled by a normally distributed random variable. The intuition is that temporary environmental effects, such as shadowing and change in incidence angle, let the charging time vary around a certain value. Fig. 7b shows that an exponential distribution is often a good fit when harvesting kinetic energy. This can be explained by the decaying



(a) Stairs dataset, normal distribution. (b) Cars dataset, exponential distribution. (c) Office dataset, Gaussian mixture model.

Figure 7: Charging time distributions of individual nodes. The nodes' charging times can be modeled by well-known distributions.

response of a piezoelectric harvester to the distinct impulses of a car during driving (e.g., acceleration, breaking, bumps) or a person during jogging (see Fig. 3). In the washer scenario, instead, we find that the continuous shaking of the industrial washing machine over long periods induces approximately normally distributed charging times. Looking at Fig. 7c, we see that in the office scenario the charging times are mostly distributed around a certain value. However, the maximum power point tracking (MPPT) of the DC-DC converter used in this scenario, which periodically disconnects the charger for a short time, leads to a second peak. We approximate this distribution with a Gaussian mixture model (GMM).

These observations motivate us to model the marginal cdf F_i of a node's charging time in the scenarios we considered through the parameters of a normal distribution (2 parameters), an exponential distribution (1 parameter), or a GMM (6 parameters for two Gaussians and two weights). The last column of Table 1 lists the corresponding model for each of the datasets. The jogging dataset contains traces from different types of harvesters: We use an exponential distribution to model the charging times of kinetic harvesting nodes and a normal distribution for the solar harvesting nodes. During an encounter, a node only needs to share the type of model and the current estimates of the model parameters.

3.3 Learning Distribution Parameters Online

We now turn to the problem of estimating the parameters of a given charging time distribution based on local observations. Given a sample of n independent and identically distributed observations, the log-likelihood $\mathcal{L}(\theta \mid x)$ and the corresponding maximum likelihood estimator $\widehat{\theta}$ are given by

$$\mathcal{L}(\theta \mid x) = \ln\left(\prod_{i=1}^{n} f_\theta(x_i)\right) = \sum_{i=1}^{n} \ln f_\theta(x_i) \tag{4}$$

$$\widehat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta \mid x) \tag{5}$$

where $f_\theta(x_i)$ is the conditional probability to observe x_i if the underlying distribution is parameterized with θ.

Unfortunately, vanilla maximum likelihood estimation is not viable in our setting. First, the observations of the charging time become available only one by one at runtime, yet the nodes do not have enough memory and energy to recompute the estimator with every new observation. Further, the charging time distributions are non-stationary. For instance, the dashed lines in Fig. 8 reveal trends in the mean and standard deviation of a node's charging time from the stairs dataset. Thus, an approach is needed that dynamically adjusts the parameter estimates to changing energy harvesting conditions.

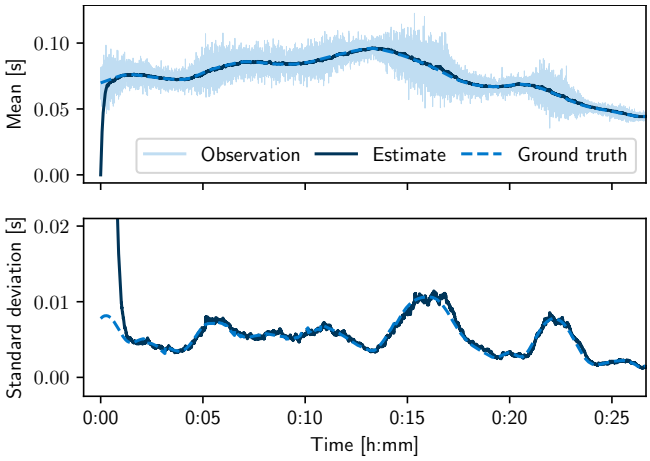


Figure 8: Varying mean and standard deviation over a moving window of one of the trace from the stairs dataset reveal non-stationarity. Using SGD, the changing distribution parameters are tracked online.

To address these problems, Bonito learns the distribution parameters online using stochastic gradient descent (SGD), which has become a popular method for training a wide range of machine learning models [5]. Compared to a sliding window based approach, SGD is less computationally demanding as the parameter update is only computed for the current observation rather than for a set of past observations that also have to be kept in memory. If the gradient of $\mathcal{L}(\theta \mid x)$ is known, one solution to (5) is to iteratively adjust $\widehat{\theta}$ along the gradient, known as gradient descent

$$\nabla\mathcal{L}(\theta \mid x) = \nabla \sum_{i=1}^{n} \ln f_\theta(x_i) \tag{6}$$

$$\widehat{\theta} = \widehat{\theta} + \eta \cdot \nabla\mathcal{L}(\widehat{\theta} \mid x) \tag{7}$$

By pulling the ∇ operator in (6) into the sum, the update step in (7) can be split into a series of updates for every individual observation x_i. This yields the update equation of SGD

$$\widehat{\theta}_i = \widehat{\theta}_{i-1} + \eta \cdot \nabla \mathcal{L}(\widehat{\theta} \mid x_i) \tag{8}$$

Sec. A derives the gradient equations required to solve (8) for the normal, exponential, and Gaussian mixture models. By keeping the learning rate η constant, Bonito implicitly reduces the weight of old observations relative to more recent observations. This way, devices dynamically learn changing properties of the charging time distribution locally, without information exchange with other devices.

Example. Fig. 8 illustrates how Bonito learns and tracks mean and standard deviation of a non-stationary normal distribution. To obtain ground truth, we sample charging times (i.e., observations) from a known normal distribution, whose mean and variance change dynamically over time. We extract these changes from one of the traces in the stairs dataset using a 2 min moving average filter. We can see in Fig. 8 that the parameter estimates of Bonito converge from their initial values (zero mean and unit standard deviation) to the true ground truth parameters within less than a minute. Then the estimates closely follow the changes of the underlying distribution.

3.4 Computing Inverse Joint CDF Efficiently

Having shared the type of model and the current estimates of the model parameters during an encounter, Bonito needs to compute the new connection interval T_C from the inverse joint cdf $F_{i,j}^{-1}$ for a user-defined probability p. This is difficult since there exists no closed-form solution for most bivariate distributions, let alone for joint cdfs of different distribution families (e.g., when a solar and a kinetic energy harvesting node in the jogging scenario want to communicate). Instead, we have to solve (3) numerically, while taking into account the energy and compute constraints of battery-free devices.

We are interested in the connection interval T_C where the joint cdf is equal to the user-defined target probability, that is, $F_{i,j}(T_C) = p$. This yields the following objective function

$$f(T_C) = F_{i,j}(T_C) - p = F_i(T_C) \cdot F_j(T_C) - p = 0 \tag{9}$$

Note that $f(T_C)$ has a single root—the sought solution—as $F_{i,j}$ is strictly monotonically increasing. Bonito solves this problem using the well-known bisection method, which iteratively finds the root of any continuous function that has its root inside a bracket (i.e., search interval). Indeed, we can derive such a bracket based on the inverse cdfs of our marginal distributions, which either have a closed form solution (exponential and normal) or are easy to approximate (GMM).

To derive a lower bracket, we first note that $F(x) < 1$ for any cdf F. It follows that $F_{i,j}(x=z, y=z) = F_i(x=z) \cdot F_j(y=z) < \min(F_i(x=z), F_j(y=z))$ and therefore the lower bracket

$$F_{i,j}^{-1}(p) > \max\left(F_i^{-1}(p), F_j^{-1}(p)\right) \tag{10}$$

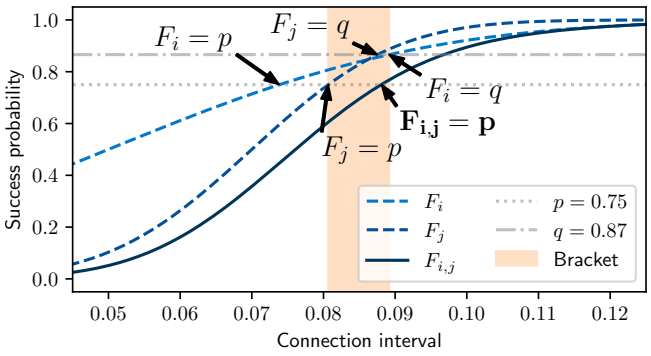


Figure 9: Bracketing the inverse joint cdf based on the inverse cdf of the marginal distributions enables efficient computation of the connection interval on resource-constrained battery-free devices.

To derive an upper bracket, we introduce $q = \sqrt{p}$ and $c = \max(F_i^{-1}(q), F_j^{-1}(q))$. Let F_m be the marginal cdf (i.e., either F_i or F_j) such that $F_m^{-1}(q) = c$, that is, the marginal cdf that reaches q *later*. Let F_n be the other marginal cdf that reaches q *sooner*. From $F_n(c) \geq F_m(c)$ follows $F_{i,j}(c) = F_m(c) \cdot F_n(c) \geq F_m(c) \cdot F_m(c) = q^2 = p$. Finally, because $F_{i,j}(c)$ is monotonically increasing, we obtain the upper bracket

$$F_{i,j}^{-1}(p) \leq max(F_i^{-1}(q), F_j^{-1}(q)) \tag{11}$$

Example. Fig. 9 shows an example, where (10) and (11) are used to determine an initial bracket for $F_{i,j}^{-1}(p = 0.75)$. The resulting bracket $[0.61, 0.77]$ is already relatively tight, and therefore we find the solution $F_{i,j}^{-1}(p = 0.75) = 0.88$ with a tolerance of 0.01 after only three bisection steps.

3.5 Impact of Target Probability

The target probability p is a key parameter of the Bonito protocol that must be set by the user. It specifies the probability that both devices have accumulated enough energy in their capacitors to become active after a connection interval T_C. A high target probability requires a long connection interval T_C, increasing communication delay and lowering throughput.

To illustrate how the choice of p impacts communication reliability and efficiency, we run trace-driven simulations as detailed in Sec. 2 on the traces from the datasets in Table 1. We use two metrics to quantify the performance of Bonito: As a proxy for communication reliability, we define the *success rate* as the ratio of successful encounters with Bonito to the total number of wake-ups. As a proxy for communication efficiency, we consider the *relative delay* as the median of all successful connection intervals with Bonito divided by the median of the optimal clairvoyant solutions according to (1).

Fig. 10 plots for each dataset success rate and relative delay averaged across all links. We can observe the following:

- A higher target probability p leads to a higher success rate, which demonstrates the plausibility of our approach.

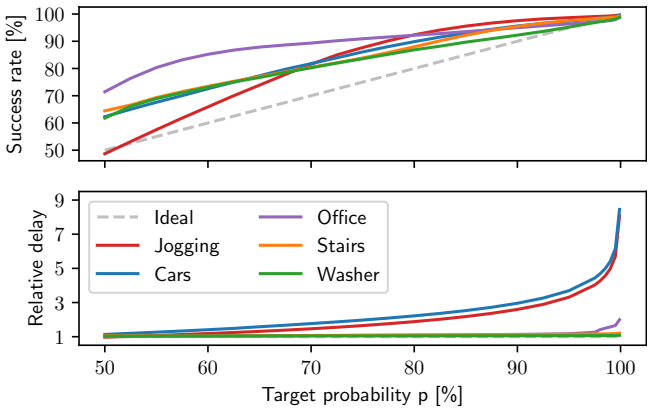


Figure 10: Trace-driven simulations reveal that the rate of successfully arranged encounters matches the user-defined target probability. The price to pay in terms of latency depends on the model of the underlying charging time distribution.

In most cases, the success rate is even slightly higher than requested, presumably due to small model errors.

- Since connection losses are costly, a higher target probability is preferable in practice. Fig. 10 shows that the price to pay in terms of a higher relative delay depends on the scenario. For the cars and jogging datasets, where most or all links include at least one node with approximately exponentially distributed charging times, the relative delay increases exponentially with p, due to the heavy tail of the distribution. For GMM (office), the increase is moderate, whereas it is hardly noticeable for the normal distribution (washer and stairs).

4 Implementation

In this section, we describe the hardware and software components of our prototype implementation.

4.1 Hardware

We design a ulta low-power battery-free node based on the popular Nordic Semiconductor nRF52805 microcontroller (MCU). This particular MCU features a 2.4 GHz BLE radio and a state-of-the-art 32-bit 64 MHz ARM Cortex-M4, which is powerful enough to complete also more demanding computations in a short time, benefitting overall system efficiency. To enable low-power timekeeping between wake-ups, the MCU is equipped with a 32 kHz crystal with ±20 ppm frequency tolerance. A TI BQ25504 DC-DC step-up converter charges a 2 mm × 1.25 mm × 1.25 mm 47 µF multilayer ceramic capacitor (MLCC) from a connected solar panel or a piezoelectric energy harvester. Once the capacitor voltage reaches a hardware-programmable threshold of 3.3 V, the BQ25504 sets one of its pins high. This pin is wired to a TI TS5A23166 analog switch that connects the MCU to the capacitor-buffered supply voltage.

Due to its DC bias characteristics, the capacitor has an effective capacitance of only 17 µF at 3.3 V. This allows for a maximum active time of around 1 ms per wake-up. A larger capacitance would increase the active time per wake-up and the charging time between wake-ups. To minimize the physical dimensions and the price of the node, we choose the minimum capacitance that allows the nodes to remain active for long enough to compensate for clock drift accumulated over a connection interval of 5 s (see Sec. 4.2).

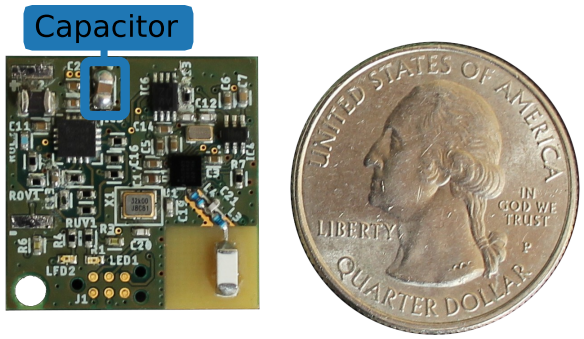


Figure 11: Prototype battery-free node based on the nRF52805 MCU. A sustainable 3.1 mm^3 ceramic capacitor is used as energy storage.

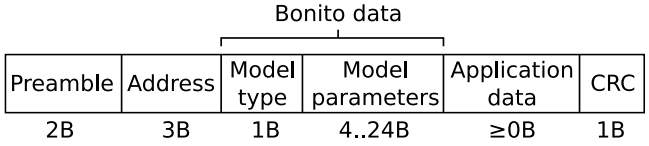


Figure 12: Packet format. Using Bonito, nodes exchange between 5 B and 25 B carrying model type and parameters during an encounter.

The node also integrates a circuit to measure the current flow from the harvester, which can be used as a sensing signal [39]. The two-layer printed circuit board (PCB) shown in Fig. 11 measures 20 mm × 20 mm. The total cost of all components is $8.73.

4.2 Software

We implement Bonito and the Find neighbor discovery protocol [19] on our battery-free nodes. Find is used to establish an initial encounter after a connection loss or a power failure.

Bonito protocol settings. We use the 2 Mbit/s BLE mode and the frame structure depicted in Fig. 12. Depending on the model type, encoded by one byte, a packet carries 1, 2, or 6 model parameters represented by 32-bit floating point values.

To jointly agree on the next connection interval, Bonito requires nodes to exchange messages bi-directionally during an encounter. The exact sequence of packet exchanges is subject to application requirements and can be flexibly configured. We implement the packet sequence shown in Fig. 13. When two nodes encounter each other using Find, one of the nodes receives the first beacon and replies with an acknowledgement. At all following encounters, the node that received the first beacon starts to listen at the time agreed on using Bonito.

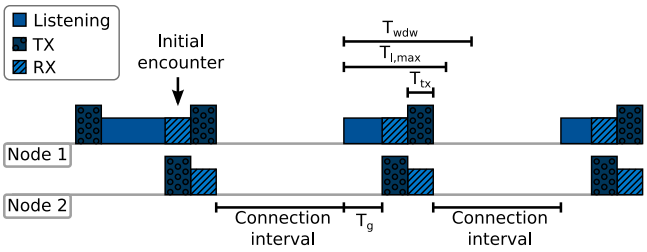


Figure 13: After the initial encounter, nodes use Bonito to agree on a connection interval. At the next encounter, one of the nodes starts to listen and the other node transmits its packet after a grace period to account for clock drift due to the long charging times. After receiving this packet, the listening nodes replies with its own packet.

Due to the small energy buffer, a node can keep the radio on for at most $T_{wdw} = 1\,\text{ms}$. Thus, the maximum listening time is $T_{l,max} = T_{wdw} - T_{tx} - T_{ta} = 820\,\mu\text{s}$, where $T_{ta} \approx 40\,\mu\text{s}$ is the time it takes to switch from receive to transmit mode and $T_{tx} = 140\,\mu\text{s}$ is the airtime of a packet with 6 model parameters and 4 B of application data. To increase the robustness to clock drift in the face of long charging times and hence long connection intervals, we let the node that sends first transmit its packet after a grace period of $T_g = 0.5 \cdot T_{l,max} - 1.5 \cdot T_{tx} = 200\,\mu\text{s}$. We can thus tolerate an offset of up to $\pm 200\,\mu\text{s}$ between the clocks of the two nodes, which corresponds to a maximum connection interval of 5 s when taking into account the frequency tolerance of the 32 kHz crystal oscillator. Upon receiving the packet, the other node switches to transmit mode and sends its own packet.

In our current implementation of Bonito, the devices consider a connection as lost whenever a planned packet exchange fails, for example, due to fading, external interference, or when one of the two devices does not reach the turn-on threshold by the end of the connection interval. In this case, they return to discovery mode and use Find to re-establish the connection.

Runtime support. In addition to Bonito and Find, we implement an efficient *soft intermittency* runtime, where the MCU is gracefully suspended to an ultra low-power mode before an impeding power failure [19]. This reduces the costs associated with a cold start after a hardware reset and allows to keep track of time between consecutive wake-up events using the built-in real-time clock (RTC). To this end, a node periodically samples the capacitor voltage during charging with the built-in analog-to-digital converter (ADC) until the capacitor voltage reaches a software-defined turn-on threshold. Then the node executes protocol and application code until it is interrupted by the power-fail comparator, upon which it immediately transitions back into low-power mode to replenish its energy buffer.

Although our runtime tries to prevent hardware resets, after multiple seconds without any energy input, the sleep current drains the remaining charge from the capacitor and the node eventually powers off. While powered off, the on-board static random access memory (SRAM) is subject to decay, that is, bits that were set to one may flip and become zero after some time. To still retain the trained model of a node's charging time distribution across short power failures, we store it in a dedicated section of the SRAM. After every model update, we compute a checksum over this section and store it next to the model parameters. If the recomputed checksum after a hardware reset does not match the checksum stored in memory, we conclude that the memory is corrupted and restart training the model with the initial parameters.

5 Evaluation

This section uses testbed experiments to evaluate Bonito on real battery-free nodes under realistic, repeatable conditions. We start by showing in Sec. 5.2 how Bonito dynamically adjusts the connection interval to changes in the nodes' charging times to maintain long-running connections. In Sec. 5.3, we compare Bonito against two baseline approaches. Finally, in Sec. 5.4, we quantify the runtime overhead of Bonito. Our experiments reveal the following key findings:

- Bonito establishes connections that outlast on average hundreds of consecutive encounters even between nodes that harvest from different types of energy sources.
- Bonito improves the throughput by 10–80× compared with the current state of the art. It achieves this by consciously keeping the connection interval as short as possible while maintaining a high success rate that agrees to within 1 % of the requested target probability.
- Depending on the distribution model, Bonito consumes between 4 % and 25 % of the energy available per wake-up on our nodes. The energy cost of losing a connection is 1000× higher than the energy overhead of Bonito.

5.1 Testbed and Settings

We connect two battery-free nodes (see Fig. 11) to two Shepherd observers [18]. In addition to recording spatio-temporal harvesting traces (see Sec. 2), Shepherd can also replay previously recorded traces and monitor the behavior of connected battery-free devices. The observers synchronously replay for all 60 links in our datasets (see Table 1) the two corresponding energy-harvesting traces. At the same time, the observers log the serial output and GPIO events of the attached nodes, which we use to compute performance metrics. In total, we collect measurements from 218 hours of testbed experiments.

For the stairs, office, and washer scenarios, we replay the recorded energy-harvesting traces as is. When using the original traces from the cars and jogging scenarios, however, we were not able to collect sufficient data points. The reason is that the piezoelectric harvesters were selected and tuned for the frequency and amplitude of the washer scenario, which led to a relatively low harvesting power in the cars and jogging scenarios, as evident from the small number of wake-ups in Table 1. Because it can take thousands of wake-ups until two nodes encounter each other, we had to scale the cars and jogging traces by a factor of five to allow for a meaningful

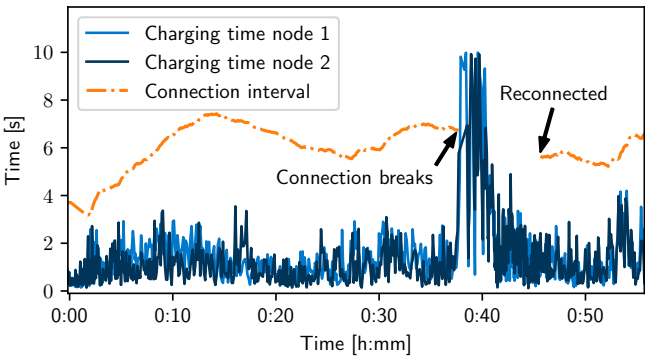


Figure 14: Real-world trace from testbed experiments showing the charging times of two nodes from the cars dataset. By dynamically adjusting the connection interval, Bonito maintains a connection for 37 min until the cars leave the highway and enter stop-and-go traffic; the charging times increase dramatically and the connection breaks.

evaluation. Note that this does not change the dynamics and shape of the charging time distributions, nor does it affect relative performance when comparing different approaches.

In all experiments, we configure Bonito with a target probability of $p = 0.99$. We use a learning rate of $\eta = 0.01$ for the normal and exponential models and $\eta = 0.001$ for GMM, which we found to perform well in a wide range of scenarios.

5.2 Maintaining Long-running Connections

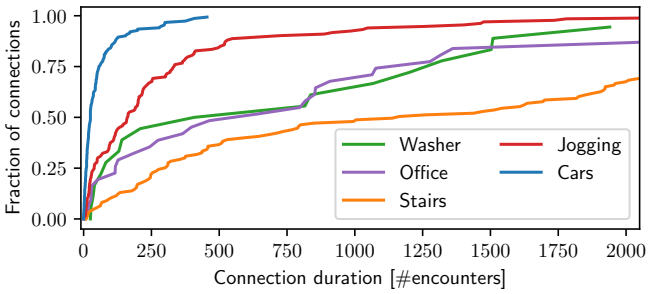
We begin by looking at how well Bonito can maintain a connection between battery-free devices. As an illustrative example, Fig. 14 shows the charging times of two nodes from the cars dataset and the connection interval determined by Bonito over the course of 55 min. Bonito successfully maintains the connection for more than half an hour by dynamically adjusting the connection interval based on the continuously updated models of the nodes' charging time distributions. Then, after around 37 min, the two cars driving in convoy exit the highway and enter stop-and-go traffic. As a result, the charging times increase suddenly and exceed the connection interval—the connection is lost. At this point, the nodes switch over to executing the Find neighbor discovery protocol and successfully reconnect after roughly 10 min. Afterward, Bonito takes over and again maintains a connection for several minutes.

Fig. 15a plots for all datasets the cdf of the connection duration in terms of the number of encounters, while Fig. 15b plots it in terms of time for the unscaled datasets (see Sec. 5.1). Overall, we find that in 90 % of the cases, the nodes stay connected for at least 30 consecutive encounters, and 40 % of the connections last for 800 encounters or more. This demonstrates that Bonito enables, for the first time, reliable and efficient communication between intermittently powered nodes.

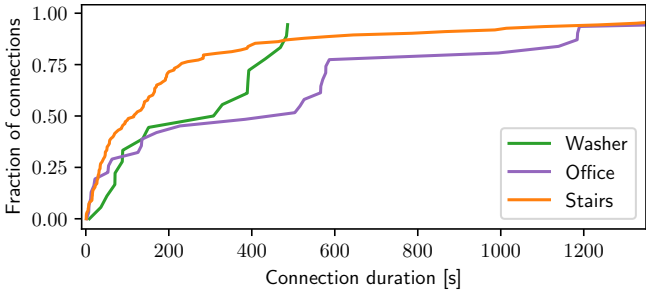
5.3 Bonito versus Baseline Approaches

We now compare Bonito against two baseline approaches:

- Greedy: This is the current state of the art. Using Greedy, nodes wake up and attempt to communicate as soon as they reach the minimum required energy level. Greedy is the prevalent execution model in the intermittent computing literature [8,33] as it maximizes the effective duty cycle of a battery-free device.
- Modest: As a complementary approach to Greedy, we design Modest. Using Modest, each node keeps track of the maximum observed charging time c_{max}. During an encounter, two nodes i and j share their current maximum charging times $c_{max,i}$ and $c_{max,j}$, and agree to meet again after a connection interval of $T_C = \max(c_{max,i}, c_{max,j})$.



(a) cdf of connection duration in terms of number of encounters.



(b) cdf of connection duration in terms of time.

Figure 15: Bonito maintains connections over hundreds of encounters even in challenging scenarios with different types of energy sources.

Our comparison uses two end-to-end metrics that also account for periods where Find runs to establish a first encounter after a connection loss or power failure. *Throughput* is the number of packets delivered from one node to another node per time unit. Note that traffic is always bi-directional, that is, the same number of packets is also delivered in the other direction (see Fig. 13). *Latency* is the time between two consecutive packet exchanges. We also consider *success rate*, which is the ratio of successfully arranged encounters to the total number of trials when using Greedy, Modest, or Bonito.

Fig. 16 plots for each dataset the throughput gains of Bonito and Modest over Greedy. We see that Bonito improves the throughput by 10–80×. For example, for the stairs dataset, Bonito achieves a throughput of 15.18 pkt/s versus 0.33 pkt/s with Greedy. Modest outperforms Greedy across the board, too, but often falls far short of Bonito's throughput.

To understand the reasons for the significant performance differences among the different approaches, we plot in Fig. 17 success rate and latency for the stairs dataset. As the charging

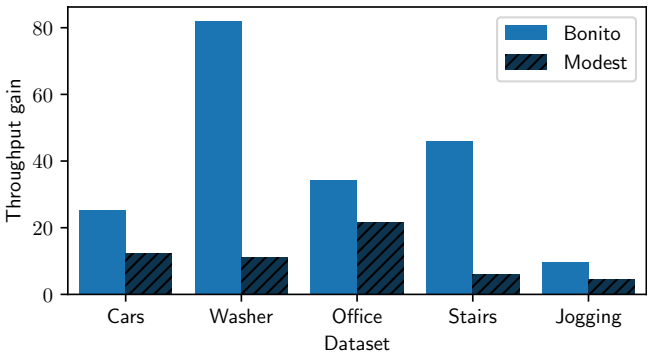
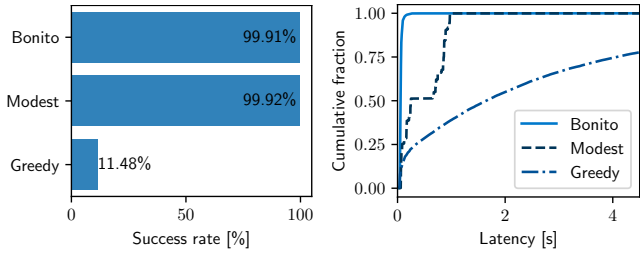


Figure 16: Throughput improvement over Greedy. By maintaining connections over many wake-ups, the average number of encounters with Bonito is at least an order of magnitude higher than with Greedy.



(a) Success rate (b) Latency

Figure 17: Detailed comparison of performance metrics for the stairs scenario. Bonito achieves a high success rate that is on a par with the Modest approach, while providing a significantly lower latency.

times vary across time and space, Greedy achieves a low success rate of only 11.48 % (see Fig. 17a). This means that in 9 out of 10 cases the nodes loses the connection right after the first encounter. Every time the connection is lost, the nodes cannot communicate until they reconnect, causing excessively long latencies as visible in Fig. 17b. Instead, Modest chooses the connection interval highly conservatively, which leads to a high success rate of 99.92 % but also long latencies. Bonito provides much shorter latencies at almost the same high success rate, which agrees to within 1 % of the requested target probability. By aiming to keep the connection interval short and to avoid the latency associated with reconnecting after a connection loss, Bonito significantly increases the end-to-end throughput compared with the two baseline approaches.

5.4 Bonito's Runtime Overhead

Next, we evaluate the runtime overhead of Bonito based on the logs from the testbed experiments. The overhead can be broken down into three components: (*i*) updating the model parameters using SGD, (*ii*) exchange of the model parameters over wireless during an encounter, and (*iii*) computing the inverse joint cdf to obtain the connection interval.

The time required to update the model is constant: 1.3 µs for exponential, 3.2 µs for normal, and 28.8 µs for GMM. This constitutes up to 2.8 % of the around 1 ms active time per wake-up. Similarly, the airtime to exchange 4, 8, or 24 bytes

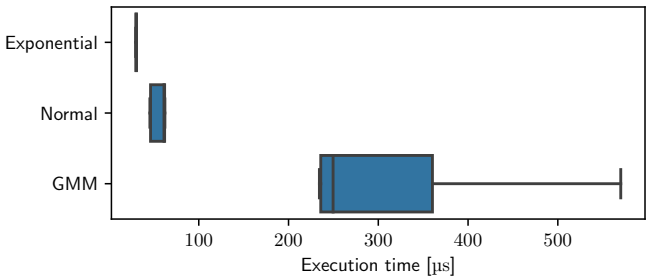


Figure 18: Distribution of execution times on our battery-free node when computing the inverse joint cdf. The execution time depends on the number of model parameters and varies with the number of bisection steps needed to satisfy the required tolerance.

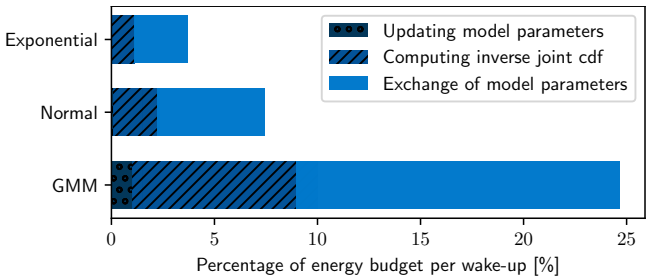


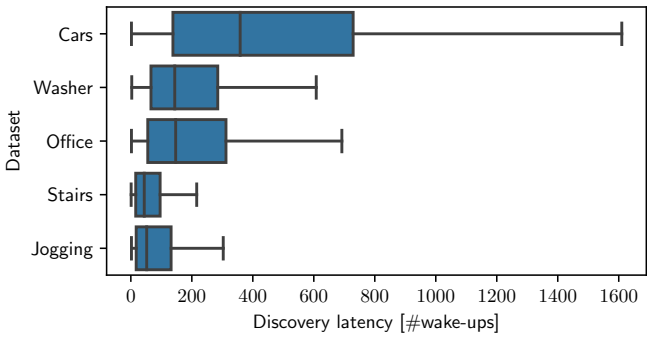
Figure 19: The energy overhead of Bonito ranges between 4 % and 25 % of the energy available per wake-up on our nodes. In absolute terms, the cost to recover from a lost connection is 1000× higher.

of model parameters is fixed and determined by the bitrate of the BLE radio. By contrast, Fig. 18 shows that the time to compute the inverse joint cdf varies depending on the number of bisection steps required to reach the desired tolerance.

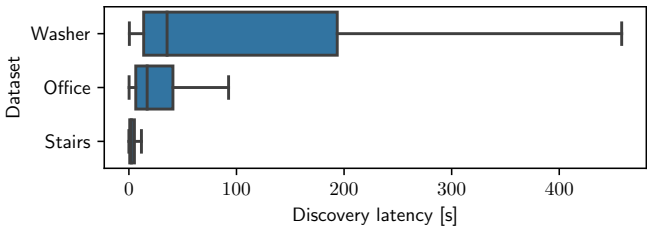
In terms of energy, our battery-free nodes have an energy budget of 27.5 µJ per wake-up. Fig. 19 shows for each model the median percentage of energy budget spent by Bonito. We can see that the required energy mainly depends on the number of model parameters and the computational complexity of evaluating the inverse joint cdf. In the worst case, for GMM, Bonito consumes 7.1 µJ, which amounts to about 25 % of the available energy per wake-up. To set this into perspective, Fig. 20a plots for all datasets the time it takes for two nodes to synchronize with the Find neighbor discovery protocol [19] in terms of the number of wake-ups, while Fig. 20b plots it in terms of time for the unscaled datasets (see Sec. 5.1). On average it takes 283 wake-ups, or 7782.5 µJ, to synchronize after a lost connection—1000× more than the energy required by Bonito to maintain a connection. This demonstrates that, overall, the absolute energy costs of Bonito are well spent.

6 Case Study: Occupancy Monitoring

Occupancy monitoring is essential to save energy in homes and commercial buildings [10, 16]. Recently, it has also become an important tool to manage the spread of infectious diseases, such as SARS-CoV2 [37]. To assess the potential of Bonito for real-world battery-free applications, we conduct an

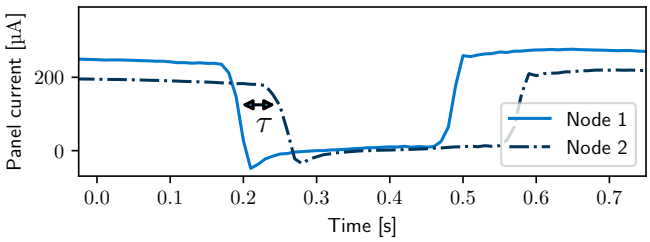


(a) Discovery latency in terms of number of wake-ups.

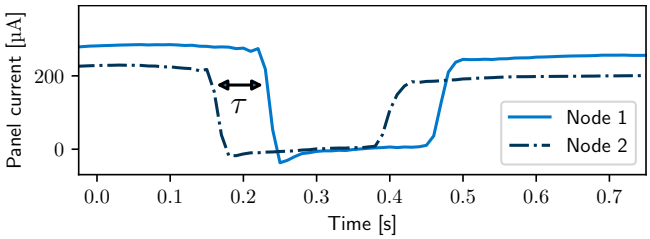


(b) Discovery latency in terms of time.

Figure 20: Synchronizing two devices with the Find neighbor discovery protocol takes a long time and consumes significant energy. Using Bonito, devices can establish long-running connections to periodically exchange data without the need to resynchronize.



(a) Person entering the room: $\tau > 0$



(b) Person leaving the room: $\tau < 0$

Figure 21: The shadow of a person passing ambient light harvesting devices on a doorframe causes a distinct temporal pattern in the solar panel current. By comparing the times of the onset of the shadowing on the two nodes, we can determine the direction of movement.

occupancy monitoring case study with our prototype nodes.

Occupancy sensor. To efficiently count the number of people in a room, we use the solar panel as a sensor [23, 39] to detect when a person enters or leaves the room. Fig. 21 shows the solar panel current of two nodes mounted next to each other on a doorframe (see Fig. 22), when a person enters the room in Fig. 21a and when a person leaves the room in Fig. 21b. To detect the direction of movement, the nodes record the time when they detect the onset of the shadowing by the person. Then the nodes exchange the recorded times and compute the time difference τ. The sign of τ indicates the direction.

Setup. We mount two battery-free nodes equipped with IXYS SM141K06L solar panels next to each other on the doorframe at the entrance of an office room, as shown in Fig. 22. The nodes sample the solar panel current with a sampling rate of 1 kHz, and record the time when the solar panel current falls below 87.5 % of its average value. The nodes run Bonito and insert the timestamp of detected events into the packets. Together with logging information (charging time, connection interval, etc.) every packet carries 26 B of application data.

Because the clocks of the two nodes are not synchronized, timestamps are transmitted relative to the start of the corresponding packet. To this end, nodes measure the time between the detected event and the start of the transmission and insert the result into the packet. The receiving node timestamps the reception of the packet and converts the contained relative timestamp to its local clock. Finally, by relating a received timestamp to the timestamp of the corresponding event that was recorded locally, the nodes compute the time difference τ.

The nodes transmit the result over wireless to an nRF52840 development board that serves as a base station. We configure the base station to timestamp the reception of packets containing a detected event and button presses of two on-board push buttons, one for each direction. Four participants randomly enter and leave the room one by one. Another person records ground truth by pressing the corresponding button on the nRF52840 board precisely when a person passes through the doorframe.

Results. The confusion matrix in Table 2 shows that the system correctly classified 60 out of 61 events, corresponding to an accuracy of 96 %. It missed just one in-event, and falsely reported an in-event and an out-event for a single in-event.

Fig. 23 plots the latency in terms of the time between a

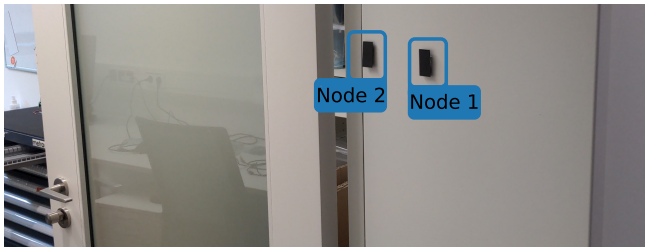


Figure 22: Two of our battery-free nodes are attached to the doorframe and harvest energy from ambient light. Thanks to Bonito, the nodes can communicate in a timely and reliable fashion, allowing them to count the number of people entering and leaving the room.

		Ground truth		
		In	Out	No event
Recorded	In	30	0	1
	Out	0	31	0
	No event	1	0	0

Table 2: By collaborating, the battery-free nodes classified people entering and leaving the room with an average accuracy of 96.83 %.

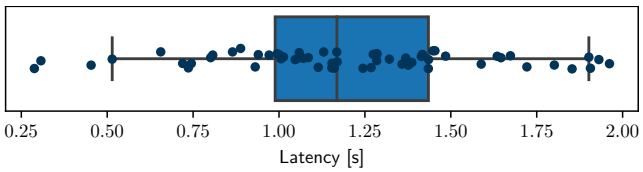


Figure 23: Due to the low communication latency provided by Bonito, the system reported detected events both timely and accurately.

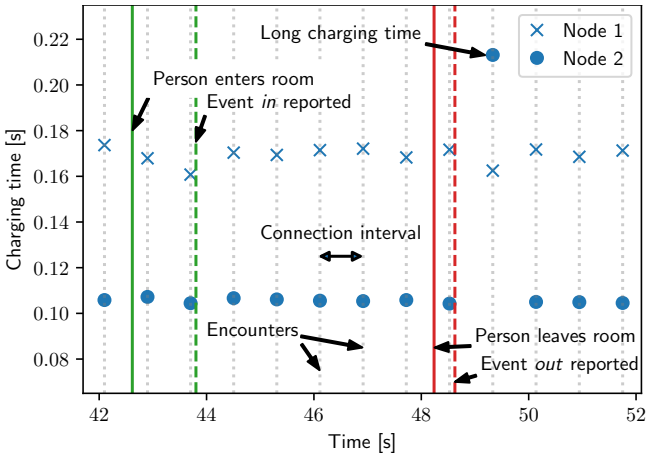


Figure 24: Example trace from the occupancy monitoring case study. The system correctly classifies and reports events to the base station. With Bonito, the connection interval is chosen large enough to sustain outliers of the charging time in response to transient shadowing.

button press and the reception of the detected event at the base station. The median latency was 1.2 s and all events were reported within less than 2 s. Over the course of the experiment, the two nodes successfully exchanged 10.56 kB of application data for an application-level throughput of 28.38 B/s.

Fig. 24 shows a ten-second excerpt from the experiment. The markers indicate the charging times of the nodes. Solid vertical lines indicate button presses (ground truth); dashed vertical lines indicate when a event was received at the base station. We can observe that, right after the received out-event, node 1 reports an exceptionally high charging time of 210 ms. This happens when the shadowing by a person occurs while a node charges its capacitor: The shadowing reduces the energy input for a short time, which prolongs the recharge. Nevertheless, by keeping the connection interval at around 700 ms, Bonito provides a stable connection despite such dynamics.

7 Discussion

Bonito is the first connection protocol for battery-free devices. It enables two devices to communicate efficiently and reliably by dynamically adapting the connection interval to changes in the devices' energy availability. In this section, we discuss limitations and opportunities for extending Bonito.

From connections to networks. The ability to efficiently and reliably exchange data between two devices is the fundamental building block required to form large wireless networks consisting of multiple battery-free devices. A number of trade-offs and challenges arise from each of the possible approaches to move from the two-node setting to larger networks, which could be explored by future work. For example, devices may sequentially connect with their neighbors or devices may try to establish Bonito connections with one common connection interval between multiple devices.

Communication with battery-powered devices. While we focus on communication between two battery-free devices, Bonito is also useful for effective communication from battery-free to battery-powered devices. For example, a battery-free tag may want to transmit data to a user's smartphone or to a battery-powered gateway in a wireless sensor network. Because battery-powered devices are in control of their wake-up times, any connection interval works for them. Thus, instead of computing the inverse joint cdf of the charging time distribution of both devices, it is sufficient to compute the inverse cdf of the charging time of the battery-free device in order to determine a connection interval that works for both devices.

Model accuracy. The goodness-of-fit of the learned charging time model critically affects the performance of Bonito. With perfect knowledge of the underlying distribution, Bonito would compute the minimum connection interval feasible for the requested target probability. Overestimating the real distribution leads to increased delay, while underestimation reduces reliability. If the distribution is so complex that a large number of model parameters or a non-parametric model (e.g., a deep neural network) would be required to accurately capture this complexity, then the limited resources on a battery-free device may not be sufficient to learn the model online.

Exploiting statistical dependence. In the current implementation, Bonito assumes statistical independence of the nodes' charging time distributions in order to compute a connection interval without prior knowledge of the statistical properties of the joint charging time distribution. After establishing a connection, the devices can record observations of the joint distribution and could attempt to exploit statistical dependence between their charging times, possibly improving communication efficiency and reliability.

8 Related Work

Intermittent computing. The thriving research area of intermittent computing has made great strides in recent years, including the first real deployments of battery-free sensors [1].

This achievement rests upon techniques that ensure forward progress [34], consistent peripheral state [6], and a reliable notion of time [11] despite frequent and random power failures. This line of research is highly relevant but completely orthogonal to our work as it deals exclusively with intermittency issues on *individual* devices and, if at all, considers communication with continuously powered base stations [42].

Battery-free device-to-device communication. Prior work on battery-free wireless device-to-device communication is mainly theoretical [24, 30, 50], studying the energy trade-offs for different scheduling, transmission, and decoding policies. Recent work discusses middleware and applications for networks of intermittently powered devices, yet explicitly leaves the question of how to communicate between the devices as an open problem [28, 48]. A simulative study also acknowledges the sheer difficulty of sychronizing the wake-up times of intermittently powered devices and proposes to communicate an energy state via an always-on backscatter radio, without demonstrating a real implementation or experiments [45]. Similar to Bonito, a recent theoretical work proposes to let nodes agree on a future point in time when they become active to increase communication throughput [46]. This time is computed based on a moving average of previous charging times, whereas Bonito lets the user explicitly trade reliability against delay by taking into account the charging time distributions.

In terms of practical work, tag-to-tag backscatter communication has mainly focused on physical-layer issues and considers intermittency an orthogonal problem [29, 35, 38, 49]. Instead, the Find neighbor discovery protocol explicitly addresses the intermittency problem and shows that by delaying wake-ups by a random time battery-free nodes can encounter each other faster [19]. We use Find to bootstrap efficient and reliable device-to-device communication with Bonito. Concurrently to our work, a protocol was proposed and implemented that lets devices "die early" when no packet is received to preserve energy and maximize the number of wake-ups [13].

Delay-tolerant networking (DTN). DTN studies networks that are only intermittently connected because of, for example, node failures, mobile users, and power outages [4, 17]. Both DTN and Bonito have the same high-level goal: effective communication in intermittently connected networks. However, DTN and Bonito address orthogonal problems toward the same end goal. While DTN is concerned with forwarding, routing, naming, in-network storage, and optimization of node trajectories to generate encounters in the spatial domain, Bonito aims to generate encounters in the time domain between nodes that are spatially close to each other. Whether concepts from the DTN literature could be applied on top of Bonito is an interesting question for future research.

Energy-aware MAC protocols. Numerous MAC protocols have been proposed for ad-hoc and sensor networks [15]. These protocols turn the radio off most of the time, and power it up only to send or receive a packet. The goal is to achieve a desired network lifetime by maintaining a certain average duty cycle. A fundamental assumption of these protocols is that the radio can be powered up *at any point in time*, which is exploited to reduce idle listening by flexibly scheduling communication among nodes. This is, however, not possible in a battery-free system, where devices are unavailable whenever the capacitor voltage is below a certain threshold, which renders existing energy-aware MAC protocols ineffective.

9 Conclusions

We have presented Bonito, a connection protocol for wireless battery-free devices. By adapting the connection interval to the different and time-varying charging times of intermittently powered nodes, Bonito maintains long-running connections that provide significantly better throughput, latency, and reliability than the state of the art. We have evaluated Bonito by implementing it on a battery-free prototype, conducting testbed experiments with real energy-harvesting traces from diverse scenarios, and demonstrating its utility in an occupancy monitoring case study. With Bonito, we contribute a prime communication primitive, device-to-device unicast, that brings the capabilities of battery-free systems one step closer to those known from today's battery-supported systems.

Availability

The data described in Sec. 2 and a Python implementation of the Bonito protocol from Sec. 3 are available under a permissive MIT license at `https://bonito.nes-lab.org/`.

Acknowledgments

We thank Sarah Nollau, Ingmar Splitt, Friedrich Schmidt, Justus Paulick, and Lebenshilfe Altenburg e. V. for supporting the data collection campaign, and all participants of the occupancy monitoring case study. Thanks also to the anonymous reviewers, and to our shepherd, Shyam Gollakota. This work was supported by the German Research Foundation (DFG) within the Emmy Noether project NextIoT (grant ZI 1635/2-1) and the Center for Advancing Electronics Dresden (cfaed).

A Appendix: Gradient Equations

Exponential distribution. The derivative of the log-likelihood function is given by:

$$\mathcal{L}(\lambda) = \log\left(\lambda \cdot \exp(-\lambda x)\right) = \log\lambda - \lambda x_i \tag{12}$$

$$\nabla\mathcal{L}(\lambda) = \frac{1}{\lambda} - x_i \tag{13}$$

Calculating the *natural gradient* by defining the step size in terms of the Kullback-Leibler divergence in the distribution space has been shown to speed up convergence in many cases [2]. We obtain the natural gradient by multiplying the regular gradient from (13) with the inverse of the Fisher Information Matrix of the exponential distribution M_{exp}:

$$M_{exp} = \lambda^{-2} \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = [M_{exp}]^{-1} \cdot \frac{1}{\lambda} - x_i = \lambda - \lambda^2 \cdot x_i \tag{15}$$

Gaussian mixture model. We adopt the gradient equations from [44]: Let $f(x_i, \mu, \sigma^2)$ be the probability density function of the standard normal distribution. The responsibility function $r(x_i, k)$ quantifies the contribution of the k-th component to the model:

$$r(x_i, k) = \frac{\rho_k \cdot f(x_i, \mu_k, \sigma_k^2)}{\sum_l^K \left(\rho_l \cdot f(x_i, \mu_l, \sigma_l^2)\right)} \tag{16}$$

The update equations for the model parameters for the k-th component are then:

$$\frac{\partial \mathcal{L}}{\partial \rho_k} = r(x_i, k) - \rho_k \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = \frac{1}{\rho_k} \cdot r(x_i, k) \cdot (x_i - \mu_k) \tag{18}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma_k^2} = \frac{1}{\rho_k} \cdot r(x_i, k) \cdot (x_i - \mu_k)^2 - \sigma_k^2 \tag{19}$$

Normal distribution. We consider the special case of a gaussian mixture model with a single component and also use the equations from [44]:

$$\frac{\partial \mathcal{L}}{\partial \mu} = (x_i - \mu) \tag{20}$$

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = (x_i - \mu)^2 - \sigma^2 \tag{21}$$

References

[1] Mikhail Afanasov, Naveed Anwar Bhatti, Dennis Campagna, Giacomo Caslini, Fabio Massimo Centonze, Koustabh Dolui, Andrea Maioli, Erica Barone, Muhammad Hamad Alizai, Junaid Haroon Siddiqui, and Luca Mottola. Battery-less zero-maintenance embedded sensing at the mithræum of circus maximus. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys)*, 2020.

[2] S. Amari and S.C. Douglas. Why natural gradient? In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[3] 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.

[4] Sanjit Biswas and Robert Morris. ExOR: opportunistic multi-hop routing for wireless networks. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, 2005.

[5] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade: Second Edition*. Springer Berlin Heidelberg, 2012.

[6] 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.

[7] 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.

[8] Alexei Colin, Emily Ruppel, and Brandon Lucia. A reconfigurable energy storage architecture for energy-harvesting devices. In *Proceedings of the 23rd ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2018.

[9] Pablo Corbalán and Gian Pietro Picco. Concurrent ranging in ultra-wideband radios: Experimental evidence, challenges, and opportunities. In *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2018.

[10] Stephen Dawson-Haggerty, Andrew Krioukov, Jay Taneja, Sagar Karandikar, Gabe Fierro, Nikita Kitaev, and David Culler. BOSS: Building operating system services. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2013.

[11] 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.

[12] Jasper de Winkel, Vito Kortbeek, Josiah Hester, and Przemysław Pawełczak. Battery-free game boy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3), 2020.

[13] Vishal Deep, Mathew L. Wymore, Alexis A. Aurandt, Vishak Narayanan, Shen Fu, Henry Duwe, and Daji Qiao. Experimental Study of Lifecycle Management Protocols for Batteryless Intermittent Communication. In *Proceedings of the 18th IEEE International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 2021.

[14] Farzan Dehbashi, Ali Abedi, Tim Brecht, and Omid Abari. Verification: can wifi backscatter replace RFID? In *Proceedings of the 27th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2021.

[15] I. Demirkol, C. Ersoy, and F. Alagoz. Mac protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4), 2006.

[16] Varick L. Erickson, Miguel Á. Carreira-Perpiñán, and Alberto E. Cerpa. Occupancy Modeling and Prediction for Building Energy Management. *ACM Transactions on Sensor Networks*, 10(3), 2014.

[17] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, 2003.

[18] 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.

[19] Kai Geissdoerfer and Marco Zimmerling. Bootstrapping battery-free wireless networks: Efficient neighbor discovery and synchronization in the face of intermittency. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2021.

[20] Graham Gobieski, Brandon Lucia, and Nathan Beckmann. Intelligence beyond the edge: Inference on intermittent embedded systems. In *Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2019.

[21] Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. Snaploc: An ultra-fast uwb-based indoor localization system for an unlimited number of tags. In *Proceedings of the 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2019.

[22] Josiah Hester and Jacob Sorber. The Future of Sensing is Batteryless, Intermittent, and Awesome. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys)*, 2017.

[23] Sara Khalifa, Mahbub Hassan, Aruna Seneviratne, and Sajal K. Das. Energy-Harvesting Wearables for Activity-Aware Services. *IEEE Internet Computing*, 19(5), 2015.

[24] 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.

[25] J.N. Laneman, D.N.C. Tse, and G.W. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information Theory*, 50(12), 2004.

[26] Christoph Lenzen, Philipp Sommer, and Roger Wattenhofer. Optimal clock synchronization in networks. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2009.

[27] 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.

[28] Gaosheng Liu and Lin Wang. Self-Sustainable Cyber-Physical Systems with Collaborative Intermittent Computing. In *Proceedings of the 12th ACM International Conference on Future Energy Systems*, 2021.

[29] 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 Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, 2013.

[30] Edward Longman, Oktay Cetinkaya, Mohammed El-Hajjar, and Geoff V. Merrett. Wake-up Radio-enabled Intermittently-powered Devices for Mesh Networking: A Power Analysis. In *Proceedings of the 18th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2021.

[31] Brandon Lucia, Vignesh Balaj, Alexei Colin, Kiwan Maeng, and Emily Ruppel. Intermittent computing: Challenges and opportunities. In *Proceedings of the 2nd Summit on Advances in Programming Languages (SNAPL)*, 2017.

[32] Brandon Lucia, Brad Denby, Zachary Manchester, Harsh Desai, Emily Ruppel, and Alexei Colin. Computational nanosatellite constellations: Opportunities and challenges. *ACM GetMobile: Mobile Computing and Communications*, 25(1), 2021.

[33] 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.

[34] Amjad Yousef Majid, Carlo Delle Donne, Kiwan Maeng, Alexei Colin, Kasim Sinan Yildirim, Brandon Lucia, and Przemysław Pawełczak. Dynamic Task-based Intermittent Execution for Energy-harvesting Devices. *ACM Transactions on Sensor Networks*, 16(1), 2020.

[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] Giulia Pazzaglia, Marco Mameli, Luca Rossi, Marina Paolanti, Adriano Mancini, Primo Zingaretti, and Emanuele Frontoni. People Counting on Low Cost Embedded Hardware During the SARS-CoV-2 Pandemic. In *Proceedings of Pattern Recognition. ICPR International Workshops and Challenges*, 2021.

[38] Jihoon Ryoo, Yasha Karimi, Akshay Athalye, Milutin Stanaćević, Samir R. Das, and Petar Djurić. Barnet: Towards activity recognition using passive backscattering tag-to-tag network. In *Proceedings of the 16th ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2018.

[39] 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.

[40] 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.

[41] Olga Saukh, David Hasenfratz, and Lothar Thiele. Reducing multi-hop calibration errors in large-scale mobile sensor networks. In *Proceedings of the 14th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2015.

[42] Lukas Sigrist, Rehan Ahmed, Andres Gomez, and Lothar Thiele. Harvesting-Aware Optimal Communication Scheme for Infrastructure-Less Sensing. *ACM Transactions on Internet of Things*, 1(4), 2020.

[43] Vamsi Talla, Joshua Smith, and Shyamnath Gollakota. Advances and Open Problems in Backscatter Networking. *ACM GetMobile: Mobile Computing and Communications*, 24(4), 2021.

[44] D. M. Titterington. Recursive Parameter Estimation Using Incomplete Data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2), 1984.

[45] Alessandro Torrisi, Kasim Sinan Yildirim, and Davide Brunelli. Enabling Transiently-Powered Communication via Backscattering Energy State Information. In *Applications in Electronics Pervading Industry, Environment and Society*. Springer International Publishing, 2021.

[46] Mathew L. Wymore, Vishal Deep, Vishak Narayanan, Henry Duwe, and Daji Qiao. Lifecycle Management Protocols for Batteryless, Intermittent Sensor Nodes. In *Proceedings of the 39th IEEE International Performance Computing and Communications Conference (IPCCC)*, 2020.

[47] Xun Xian, Xinran Wang, Jie Ding, and Reza Ghanadan. Assisted learning: A framework for multi-organization learning. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[48] Kasim Sinan Yildirim and Przemyslaw Pawelczak. On Distributed Sensor Fusion in Batteryless Intermittent Networks. In *Proceedings of the 15th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019.

[49] Jia Zhao, Wei Gong, and Jiangchuan Liu. X-tandem: Towards multi-hop backscatter communication with commodity wifi. In *Proceedings of the 24th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

[50] 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.