This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edi
content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2023.3266724

ELIXIR: An Expedient Connection Paradigm for Self-Powered IoT Devices

Chen Pan*, Wen Zhang*, Yanzhi Wang†, Mimi Xie‡
*Department of Computer Science, Texas A&M University–Corpus Christi
†Department of Electrical Computer Engineering, Northeastern Univesity
‡Department of Computer Science, The University of Texas at San Antonio

***Abstract*—IoT devices usually work under power-constrained scenarios like outdoor environmental monitoring. Considering the cost and sustainability, in the long run, energy-harvesting technology is preferable for powering IoT devices. Since harvesting power is intrinsically weak and transient, the connection between IoT devices not only cannot be maintained constantly but is also extremely difficult to be established. In order to communicate, those devices should have a synchronized timeline so that both transmitter and receiver can start connection simultaneously. Yet due to the transient nature of ambient energy, volatile time data can be easily tampered with by the unstable power supply or corrupted completely by frequent power outages. As a result, unsynchronized IoT devices require significant efforts in time and energy to reconnect and communicate, which further escalates the performance degradation of the edge network. To adapt to ubiquitous self-powered scenarios on the IoT edge, we propose ELIXIR, an expedient connection paradigm, to synchronize swiftly and autonomously in a joint effort for self-powered IoT devices. The lightweight and highly efficient natures enable ELIXIR to be integrated into low-power IoT devices easily and efficiently. The experimental results show that the proposed ELIXIR can avoid a connection loop and help speed up the connection $2.83X$ and $1.84X$ faster than baseline methods.**

***Index Terms*—Transient Power, Energy Harvesting, Communication, Embedded Systems, IoT**

I. Introduction

Self-powered IoT devices that operate without batteries are increasingly appealing due to their low maintenance cost, sustainability, and eco-friendliness. Powered by energy harvested from ambient sources such as light, vibrations, and radio frequency [1], those devices enable a new generation of applications where batteries are inconvenient for replacement or recharging. Although the benefits of implementing the energy harvesting system are promising for IoT devices, harvesting power is intrinsically weak and intermittent. A typical self-powered IoT device is usually connected with a small capacitor to buffer sufficient energy for progress preservation during power failure time [2]. When there is no power, the device is put to sleep mode or turned off which will be waken up when power comes back on again.

The intermittent nature of ambient energy supply can frequently interrupt the task execution resulting in progress loss and inconsistency errors. To address those problems, extensive solutions have been proposed to preserve the computation progress without errors [3], [4]. Besides the computation progress, volatile time data can also be easily tampered by the unstable power supply or corrupted completely by frequent power outages. As a result, the time of IoT devices will become asynchronized while a synchronized timeline is required so that both transmitter and receiver can turn on their radio simultaneously for communication. After losing time synchronization, two devices require huge efforts in time and energy to build a wireless connection for transmitting data.

Self-powered devices need to communicate with each other with the aim of passing messages, collaborative data processing, and the flexibility of the network architecture. However, wireless communications between self-powered devices have rarely been studied. Most existing work optimistically assumes that when a self-powered device needs to communicate with another one, the other one is active and ready for communication [5] which is unrealistic in energy harvesting scenarios. Unlike battery-powered devices, self-powered devices are inactive most of the time. For example, a self-powered device may need to stay inactive and harvest energy for 10 seconds for sustaining active radio for 1 second or even less when the harvesting power is ultra-low.

To establish a connection, two devices need to have an overlapping period of active radio. However, due to the unpredictability of the duty cycle caused by the unstable power supply, it takes a long time, or even forever for two devices to discover each other to establish the connection. What's even worse is that the attempt to establish the connection is not a one-time endeavor. After the two devices discover each other, they will try to synchronize their timeline for communication. However, their track of time will be gradually lost when there is no harvesting power even with the persistent time-keeping solutions [6]. As a result, two self-powered devices have to repeatedly reconnect while each attempt of connection costs enormous time and energy overhead.

The connection problem poses a grand challenge for future applications where direct radio communication is required between self-powered devices: how can two asynchronous self-powered devices that work intermittently establish a connection without prior knowledge of their timeline and external assistance? This challenge is fundamental to all self-powered networks regardless of the communication protocols. There is limited research on intermittent transmission for self-powered devices. One potential approach to address this challenge is based on neighbor discovery protocol. In [7], random waiting delays are introduced before turning on the radio to discover the neighbor node. Despite better performance than without introducing any random delay, all the energy and time are wasted for random waiting. Besides, there is no pure randomness. It might take forever for the devices to discover each other. [8] proposed to discover neighbour for asynchronous connection based on Chinese Reminder theory.

Neighbor discovery protocols have also been proposed for battery-powered wireless networks [8]–[10]. Those protocols let the devices wake up according to a schedule in order to maximize the lifetime of the battery. However, none of them are applicable to self-powered networks because they require the nodes to wake up at arbitrary points in time without taking the intermittent power supply into account [7].

To address this challenge, this paper proposes ELIXIR, an expedient connection paradigm that enables self-powered IoT devices to establish connections swiftly and autonomously even under frequent power failure conditions. In ELIXIR, each self-powered IoT device has two roles: transmitter and receiver. Before the deployment of the devices, two sets of different prime numbers that are co-prime to each other are locally stored on each device corresponding to the two roles. Specifically, ELIXIR works by letting the transmitter/receiver select the smallest prime number from the transmitter/receiver prime number set that meets the connection latency requirement with the current harvesting power. The two unsynchronized devices then wake up their radio after every prime number of time units. The duty cycle of each device is determined based on the harvesting power. To adapt to the varying availability of ambient energy, each device adaptively changes the prime numbers to ensure an expedient connection.

ELIXIR features a lightweight design that's greatly compatible and friendly to existing communication protocols. As far as we know, the proposed ELIXIR is the first-ever paradigm that can efficiently and expediently help establish connections for synchronization. It offers great flexibility for applications since each device can independently select its duty cycles while ensuring a successful connection. In a nutshell, the major contributions of this work are summarized as follows:

- Comprehensive system modeling and analysis are conducted, which serve as the theoretical foundations for the proposed techniques in ELIXIR.
- An expedient connection paradigm, known as ELIXIR, is proposed to repair connection anomalies swiftly and autonomously in a joint effort for self-powered IoT devices.
- Extensive experimental evaluations under various energy harvesting scenarios are conducted to demonstrate the energy-efficient, robust, and lightweight features of the proposed ELIXIR paradigm.

The remainder of this paper is organized as follows. First, Section II illustrates the motivation of this work with one example. Then, Section III provides system modeling and analysis, which provides a theoretic basis for the proposed ELIXIR. After that, Section IV proposes ELIXIR to repair connection anomaly expediently and autonomously. Following that, Section V conducts experiments to demonstrate the performance of ELIXIR. Finally, Section VI discusses related work and Section VIII concludes this work.

II. Motivation

This section uses an example to demonstrate the difficulty of establishing a connection after self-powered IoT devices lose track of time after a power outage.

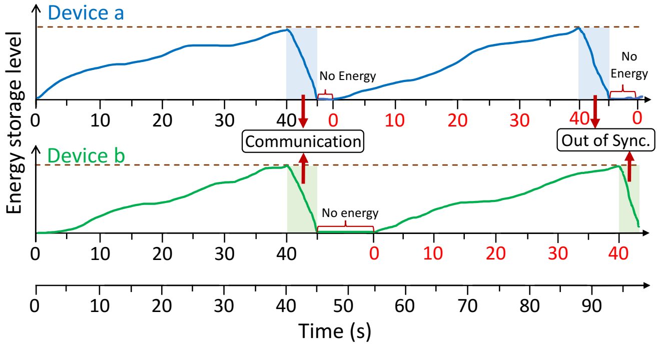


Fig. 1. IoT devices lose track of time after a power outage.

A. Connection Odyssey

The example in Figure 1 shows that the power interruption could disrupt the on-chip time module resulting in communication partners being unsynchronized with each other. Under such circumstances, establishing a connection between the unsynchronized IoT devices can be an odyssey, which requires huge efforts both in time and energy.

In this figure, the energy level of two devices a and b changes with time. Each device has a local timer for time tracking. At time 0, the two devices are synchronized and start harvesting energy. At time 40, both devices harvest enough energy and turn on their radio for communication, which is successfully completed at time 45. Nevertheless, a power outage happens after their communication, which resets the timer module of both a and b as highlighted in red numbers. Whenever the power recovers, a and b become out of synchronization. If a and b follow their previous timeline: turn on the radio to establish a connection after harvesting energy for the 40s, the attempt to connect will fail since devices a and b have no overlapping period of active radio. They will end up never being able to establish a successful connection if they follow this duty cycle. Such a situation will deteriorate the performance of self-powered IoT devices.

One way to address this problem is by "shifting" the time that the self-powered IoT device turns on its radio so that a and b could eventually end up having a time overlap with each other when their radios are both active. One existing solution [11] is to wait for a random delay before turning on the radio to connect with an appropriate duty cycle to save energy while making sure the connection can be established promptly. Take this figure for example, if device a waits for 5s at time 40s (red), device a will turn on the radio at approximately the same time as device b. As a result, the connection will be successful. However, in the energy harvesting system, such a procedure is not effective mainly for three reasons.

First, waiting for a random delay can not guarantee a successful connection. It might take a long time or even forever establish a connection. Second, it's a waste of time and energy to keep the device in the waiting state after harvesting sufficient energy. The capacitor has been charged to the maximum capacity, and thus the harvested energy during the waiting time is wasted. Third, as the communication power is far beyond

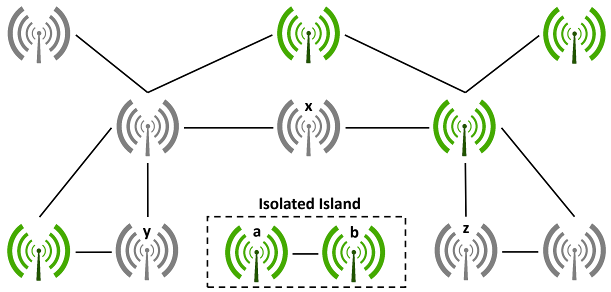


Fig. 2. Node a and b are Isolated from the Main Edge Network.

the energy harvesting power, the upper bound of the average duty cycle is determined by the ambient environment other than being deliberately manipulated. These reasons render the "randomized" connection procedure ineffective in the connection establishment of self-powered IoT devices. Therefore, in this paper, we propose a novel connection strategy by scheduling the self-powered IoT device to turn on its radio in a pattern without having to wait for a random time latency while considering the availability of ambient energy, which can guarantee the expedient establishment of the connection.

B. Isolated Islands

The second example, in Figure 2, explains a problem that even if the IoT device a and b have connected successfully, they may not likely synchronize to the rest of the network, which makes them isolated islands for external communication. Hence, the data they collected becomes meaningless as it cannot reach the upper layer of IoT.

As we can see from Figure 2, although a and b are connected, they are isolated from the rest of the edge network where x and y are one-hop neighbors of a, x and z are one-hop neighbors of b, and x, y, and z are unaware of the situation of a and b. In this situation, even if a and b have recovered their connection, They are isolated from the rest of the networks rendering their communication data useless. One way to help isolated pairs break out of such difficulty is for each member of the network to be aware that everyone including itself is subject to a power outage. So if an extra energy budget is available, it should be voluntarily contributed to help its neighbors, which may likely be out of sync with the rest of the network, for connection. Reciprocally, neighbors can help it get synchronized with its extra energy budget whenever it is possible. Such voluntary procedure (Section IV) can effectively solve the problem of isolated islands.

From the examples, we can see that both connection odyssey and isolated islands are major obstacles for unsynchronized self-powered IoT devices to recover communication. Regarding the problem, a comprehensive theoretic analysis will be given in Section III.

III. Modeling and Analysis

This section models the connection process of self-powered IoT devices. Based on the modeling, a series of comprehensive analyses is conducted which leads to the findings of how to guarantee a successful connection and how to further expedite such a connection. Those conclusions further shed light on the design of ELIXIR.

A. Connection Modeling

Since the connection needs to succeed as soon as possible, immediately after being charged up, devices will wake up and start to connect. Here, we define the duty cycles of node a and b as δ_a and δ_b respectively, and their time gaps between two adjacent wakeups as T_a and T_b. We define X as the very first connection event of node a at $t = x$. During X, node a will turn its radio on and resume its previous communication operation (listening or sending) continuously during the active period ($T_a\delta_a$). Respectively, we define Y as the very first connection event of node b at $t = y$. For both a or b, after the first connection events, their following connection events happen repetitively with time periods of T_a for a and T_b for b until the connection is established. This means, any event after the first connection event can be considered as definite, which can always be able to trace back to their first connection events. Therefore, we use $f_X(x)$ and $f_Y(y)$ to denote the probability density functions ($PDFs$) of event X and Y respectively. Although $f_X(x)$ and $f_Y(y)$ are environment-specific, they don't affect the following deduction. We further define the first joint connection event as XY. As X and Y are independent to each other, the corresponding PDF f_{XY} can be formulated as in Eq. (1)

$$\begin{aligned}
&f_{XY}(x,y) = f^x_{XY} + f^y_{XY} + f_X(x)f_Y(y)\\
&f^x_{XY} = \Sigma_{n=0}^{N_y-1} f_X(x)f_Y(y-nT_b)\\
&f^y_{XY} = \Sigma_{n=0}^{N_x-1} f_X(x-nT_a)f_Y(y)\\
&N_x = \lfloor x/T_a \rfloor, \quad N_y = \lfloor y/T_b \rfloor, \quad x \geq 0, y \geq 0
\end{aligned} \tag{1}$$

To evaluate whether the connection can be established, we only consider the joint first connection event XY where X only pairs with its adjacent Y. Specifically, we consider the scenario where both node a and b have started the connection attempt after accumulating enough energy. Given the periods of a and b, we can infer that during the first joint connection event XY, compared with the starting time of b, node a should start to connect no earlier than T_a and no later than T_b. Hence we can narrow down the joint sample space of XY as in Eq. (2)

$$\Omega = \{XY| - T_a < x - y < T_b, \quad x \geq 0, y \geq 0\} \tag{2}$$

Further, the first joint connection event XY with successful connection for a and b should belong to subspace S^0 as shown in Eq. (3)

$$S^0 = \{XY|\sigma - T_a\delta_a \leq x - y \leq T_b\delta_b - \sigma, \quad (x,y) \in \Omega\} \tag{3}$$

where σ is a small time overlap required for a and b to establish the connection when they are active simultaneously. The rationale for introducing σ is that, in order to connect, both nodes a and b should have time overlaps to each other to send preamble and sync packets including handshaking beacons such as RTS and CTS when they are active simultaneously. Therefore, σ should also be larger than the active cycles $T_a\delta_a$ and $T_b\delta_b$ of node a and b respectively. With that, for the event

XY bounded by Ω, the probability of a successful connection can be formulated as in eq. (4)

$$Pr^0 = Pr\{(x,y) \in S^0 | (x,y) \in \Omega\} = \frac{\iint_{S^0 \cap \Omega} f_{XY}(x,y)\,dx\,dy}{\iint_{\Omega} f_{XY}(x,y)\,dx\,dy} \tag{4}$$

For the n^{th} joint connection event bounded by Ω, event $X^{n_x}Y^{n_y}$ should belong to subspace S^n as shown in Eq. (5)

$$\begin{aligned} &S^n = \{X^{n_x}Y^{n_y} | \sigma - T_a\delta_a \le x^{n_x} - y^{n_y} \le T_b\delta_b - \sigma\} \\ &(x^{n_x}, y^{n_y}) \in \Omega \end{aligned} \tag{5}$$

As X^{n_x} and Y^{n_y} are determined by x and y, given time t for event observation, we have Eq. (6)

$$\begin{aligned} &X^{n_x} = x + n_x T_a \\ &Y^{n_y} = y + n_y T_b \\ &n_x = \lfloor (t-x)/T_a \rfloor, \quad n_y = \lfloor (t-y)/T_b \rfloor \end{aligned} \tag{6}$$

With Eq. (6), Eq. (5) can be further transformed into Eq. (7) which is represented by (x,y).

$$\begin{aligned} &S^n = \{XY | \sigma - T_a\delta_a - \Delta \le x - y \le T_b\delta_b - \sigma - \Delta\} \\ &n = n_x + n_y, \quad \Delta = n_x T_a - n_y T_b \\ &(x,y) \in \Omega \end{aligned} \tag{7}$$

Further, the probability that the connection is only succeeded during the n^{th} joint connection event can be formulated as in eq. (8)

$$\begin{aligned} Pr^n &= Pr\{(x,y) \in S^n | (x,y) \in \Psi^n\} \Pi_{k=0}^{n-1}(1 - Pr^k) \\ &= \frac{\iint_{S^n \cap \Psi^n} f_{XY}(x,y)\,dx\,dy,}{\iint_{\Psi^n} f_{XY}(x,y)\,dx\,dy,} \Pi_{k=0}^{n-1}(1 - Pr^k) \\ \Psi^n &= \Omega - \Sigma_{k=0}^{n-1} S^k \end{aligned} \tag{8}$$

Here, Ψ^n represents the subspace of Ω that is complementary to the union space of all previous $S^k|_{0<k<n}$. In other words, the Pr^n represents the probability that the connection only succeeded during the n^{th} connection events while all previous connections failed. With Pr^n, the expected time $Ex[t_a]$ and $Ex[t_b]$ for a and b to establish the connection after their first joint connection event can be formulated as in Eq. (9)

$$\begin{aligned} &Ex[t_a] = \Sigma_{n=0}^{\chi} n_x T_a Pr^n + H(1 - \Sigma_{n=0}^{\chi} Pr^n) \\ &Ex[t_b] = \Sigma_{n=0}^{\chi} n_y T_b Pr^n + H(1 - \Sigma_{n=0}^{\chi} Pr^n) \\ &H \to \infty, \quad \chi \to \infty \end{aligned} \tag{9}$$

Eq. (9) shows that if all S^n together cannot cover the entire sample space Ω, then $\Sigma_{n=0}^{\infty} Pr^n < 1$. This means that a and b may never be able to connect successfully, thus the expected connection time $Ex[t] \to \infty$ and the maximum connection time $Max[t] = H \to \infty$. Therefore, it is necessary to ensure $\Sigma_{n=0}^{\infty} Pr^n = 1$. In the following sub-sections, we will first conduct a comprehensive analysis of how to ensure a successful connection. Then, we further discuss the methods to improve the chance of a successful connection and minimize the maximum connection time ($Max[t]$).

B. Condition for a Guaranteed Connection

According to Eq. (8) and Eq. (9), $Ex[t]$, aside from f_{XY}, is also determined by the time values of (x,y)which is further determined by S^n. Thus, we first need to confine the boundary of $Ex[t]$, thus ensuring $Max[t] < \infty$, so that the connection can guarantee to be successful. Given T_a as the period for node a, T_b as the period for node b, $G = GCF(T_a, T_b)$ as the greatest common factor, and $M = LCM(T_a, T_b)$ as the least common multiple of T_a and T_b, Theorem 1 provides solution to address this concern.

Theorem 1. *Given the duty cycles of nodes a and b as δ_a and δ_b respectively, when $T_a\delta_a + T_b\delta_b - 2\sigma \ge G$ is satisfied, the connection of a and b are guaranteed to be successful where the maximum connection time $Max[t]$ is no greater than M ($Max[t] \le M$).*

Proof. First, let $\upsilon(t)$ represents the time distance between the events X^{n_x} and Y^{n_y} in the n^{th} joint connection event, which is formulated as in Eq. (10)

$$\upsilon(t) = X^{n_x} - Y^{n_y} \tag{10}$$

Based on Eq. (5), (6), and (7), the Eq. (10) can be further simplified as Eq. (11)

$$\upsilon(t) = mod(t - y, T_b) - mod(t - x, T_a) \tag{11}$$

Where the function $mod(J,k)$ represents the modulo operation to calculate the remainder of the Euclidean division of J by k. Given $M = LCM(T_a, T_b)$, we know that $\upsilon(t)$ should satisfy Eq. (12)

$$\upsilon(t) = \upsilon(t + kM) \quad k \in \mathbb{N}^0 \tag{12}$$

Further, for $t \in [\tau, \tau + M)$, the number of joint connection events can be calculated as in Eq. (13)

$$N_\Delta = \frac{M}{T_a} \times \frac{M}{T_b} \tag{13}$$

Therefore, for any (x,y) pair, there are only N_Δ corresponding $\upsilon(t)$. Therefore, we define a set Υ to store the value of $\upsilon(t)$, which is formulated in Eq. (14)

$$\Upsilon|_{x,y} = \{\upsilon_0, \upsilon_1, ..., \upsilon_{N_\Delta - 1}\} \tag{14}$$

Where each individual υ is sorted in ascending order with υ_0 as the minimum and $\upsilon_{N_\Delta-1}$ as the maximum. Given $G = GCF(T_a, T_b)$, we further know that the difference between any two adjacent elements is G. In this case, Υ can be further formulated in Eq. (15)

$$\Upsilon|_{x,y} = \{\upsilon_0, \upsilon_0 + G, ..., \upsilon_0 + (N_\Delta - 1)G\} \tag{15}$$

Since, $\forall t \in [\tau, \tau + M)$ with $\tau \ge min(x,y)$, $\exists t^+$ and t^- that satisfy Eq. (16)

$$\begin{cases} mod(t^+ - x, T_a) = 0, & mod(t^+ - y, T_b) \ge 0 \\ mod(t^- - y, T_b) = 0, & mod(t^- - x, T_a) \ge 0 \end{cases} \tag{16}$$

In this case, $\forall (x,y) \in \Omega$, we have

$$\begin{cases} \upsilon_0 \le 0 \\ \upsilon_0 + (N_\Delta - 1)G \ge 0 \end{cases} \tag{17}$$

With Eq. (11), the condition in Eq. (7) can be further transformed into Eq. (18)

$$\sigma - T_a\delta_a \leq \upsilon^* \leq T_b\delta_b - \sigma \tag{18}$$

Therefore, if $\exists \upsilon^* \in \Upsilon$ that satisfies the above condition, the connection can be established. Since $\sigma - T_a\delta_a < 0$ and $T_b\delta_b - \sigma > 0$, according to Eq. (17), either $\upsilon_0 = 0$ or $\upsilon_0 + (N_\Delta - 1)G = 0$ can surely satisfy Eq. (18). For the rest of the cases in Eq. (19)

$$\begin{aligned} &\upsilon_0 < 0 \\ &\upsilon_0 + (N_\Delta - 1)G > 0 \end{aligned} \tag{19}$$

We can always find $\upsilon^k \leq 0$ and $\upsilon^{k+1} \geq 0$. Since their distance is G, as long as Eq. (20) is satisfied, at least one of them (υ^k and υ^{k+1}) should satisfy the condition in Eq. (18)

$$\begin{aligned} T_b\delta_b - \sigma - (\sigma - T_a\delta_a) &\geq G \\ &\Downarrow \\ T_b\delta_b + T_a\delta_a - 2\sigma &\geq G \end{aligned} \tag{20}$$

Therefore, $\forall (x, y) \in \Omega$, as long as Eq. (20) is satisfied, within time M node a and b can surely connect to each other. This completes the proof of Theorem 1. □

C. Facilitation of the Condition

Notice that Theorem 1 provides the condition to guarantee a successful connection. Further, Corollary 1.1 provides insight on how to maximize the chance to satisfy the condition.

Corollary 1.1. *When $F = T_a\delta_a + T_b\delta_b - 2\sigma \geq G$, in order to maximize the chance for a successful connection, T_a and T_b should be co-prime to each other.*

Proof. According to Theorem 1, to maximize the chance for a successful connection, we can either maximize F or minimize G. However, by increasing F, the duration for connection will be further increased. Therefore, we focus on minimizing G. Since $G = GCF(T_a, T_b)$, we have minimum $G^{min}(T_a, T_b) = 1$ in which T_a and T_b are co-prime to each other. □

D. Expedition of the Connection

Corollary 1.2 further paves a path on how to reduce the connection time.

Corollary 1.2. *For all the combinations of (T_a, T_b) that are co-prime to each other and can guarantee a successful connection, a lower $M = LCM(T_a, T_b)$ results in a faster connection.*

Proof. From Theorem 1, we know that the maximum connection time $Max[t]$ is no greater than $M = LCM(T_a, T_b)$. However, the significance of $M = LCM(T_a, T_b)$ is beyond that. Specifically, since nodes a and b can start to reconnect at any moment, according to Eq. (13) and (15), the reconnection can be successful in any of the following timestamps shown in Eq. (21) after both nodes a and b attempt to connect.

$$T = 0, G, 2G, ..., (T_aT_b - 1)G \tag{21}$$

Therefore, on average the reconnection time is $(T_aT_b - 1)/2 = (M - 1)/2$. Hence, a lower M results in a faster connection. □

The experiments show that by minimizing $Max[t]$, the $Ex[t]$ can also be reduced significantly as well. Theorem 1 along with Corollary 1.1 and 1.2 provides a comprehensive theoretic basis for the proposed ELIXIR in Section IV to establish an expedient connection between self-powered IoT devices.

IV. ELIXIR

ELIXIR acts as a mutual transient communication recovery agreement between any two devices. Whenever their connection has been compromised due to the power outage, ELIXIR will act as a mutually agreed contingency plan, so that even though an IoT device has no knowledge of its communication partner, the connection can still be established expediently.

A. Design Flow

The design flow of ELIXIR is based on the theoretic analysis in Section III. First, according to Corollary 1.1, the periods of any communication pairs will be configured to be co-prime to each other. In this way, the condition required for a successful connection can be satisfied much easily. Then, according to Corollary 1.2, among all pairs of co-prime periods that satisfy the condition in Theorem 1, the smaller the least common multiple M is, the faster the connection will be. So ELIXIR will try its best to speed up the connection accordingly given the constraint of energy.

B. Paradigm Overview

The overview of the proposed ELIXIR is shown in Figure IV-B. The ELIXIR guides a self-powered IoT device to establish a connection. Initially, ELIXIR will check the runtime event, if the IoT device is Idle or conducting communication, TTC (Algorithm IV.1) will be triggered to initiate an attempt to connect. Specifically, if the IoT device fails to connect during the communication event or has extra energy during the Idle event, the core function $Gearbox$ (Algorithm IV.2)

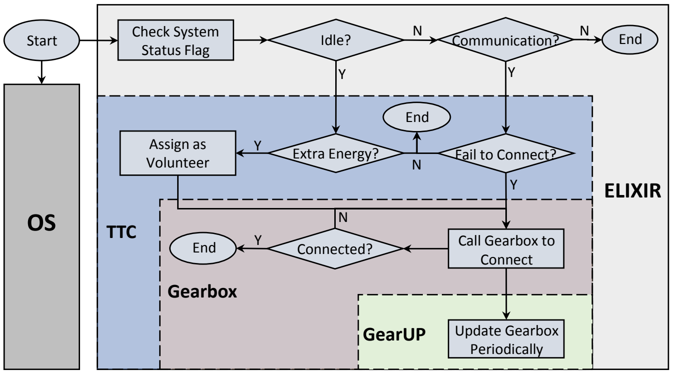


Fig. 3. Overview of ELIXIR with Submodules including TTC, Gearbox, and Gearup to Ensure Fast Wireless Reconnection after Losing Synchronizations.

TABLE I
LIST OF NOTATIONS

Symbol	Description
σ	time overlap required for node a and b to establish the connection
G^t, G^r, and G^v	The prime period pool for the transmitter, receiver, and volunteer respectively
R	The device's runtime role
Υ	The role-associated period pool
κ	The number of primes in Υ
Θ	The struct that holds all the prime periods in Υ and corresponding power thresholds
Ψ	The struct that holds candidate periods and corresponding votes for selection.
T	The duration of current period
Δ_t	Parameter updating period
$\overline{P_c}$	The average charging power within Δ_t
E_c	The residual energy in capacitor

will be triggered. For the IoT device, $Gearbox$ is responsible to help repair the connection anomaly of itself or acting as a volunteer to assist its possible neighbors with connection in a timely manner. To keep time and energy efficient for $Gearbox$ to operate, its parameters will be updated by $GearUP$ (Algorithm IV.3). In the following, TTC, $Gearbox$, and $GearUP$ will be explained in detail.

C. Tentative Transient Connection

The purpose of Tentative Transient Connection (TTC) is to help unsynchronized IoT devices detect and repair a connection anomaly in two parallel directions, known as "self-healing" (line 8-20) and "voluntary rescue" (line 21-25).

Whenever TTC is triggered, it first identifies the role of an IoT device. If the role is Transmitter/Receiver as shown from line 9-14, it initiates attempts to connect to its CP by sending/listening beacon signals during their arranged time window for communication. If attempts fail, the connection anomaly is detected. If the role is Volunteer, the device has an extra energy budget. In this case, from line 21-24, Volunteer further specifies its role as Transmitter or Receiver opportunistically, depending on how frequent its transmitting and receiving jobs are.

Once the connection anomaly has been identified or the Volunteer role has been further specified, TTC assigns the role-associated period pool (Υ) to the IoT device. Specifically, Transmitter is assigned with $\Upsilon = G^t$, the Receiver is assigned with $\Upsilon = G^r$, and Volunteer is assigned with $\Upsilon = G^v$. In line 20 and 25 Υ will be further used in Algorithm IV.2 ($Gearbox$) for the devices to repair connection anomaly with proper selection of periods. Here G^t, G^r, and G^v are mutual disjoint sets of prime numbers. This means whenever the devices with different roles try to connect with each other, their periods are co-prime to each other. Such a setting is followed by the conclusion of Corollary 1.1, which facilitates the condition required for a successful connection.

D. Co-prime Gearbox

The purpose of $Gearbox$ is to select an appropriate prime period according to the harvesting power for connection.

Algorithm IV.1: Tentative Transient Connection (**TTC**)

```
   Input: $\sigma$
1  $\Omega \leftarrow$ a set of $N$ prime numbers in ascending order;
2  $G^v \leftarrow$ a set of $k$ prime numbers from $\Omega$ in ascending order;
3  $G = \Omega - G^v$;
4  $G^t = \{G(2i) | i = 0, 1, .., \lfloor N/2 \rfloor\}$;
5  $G^r = \{G(2i+1) | i = 0, 1, .., \lfloor N/2 \rfloor\}$;
6  $P_w \leftarrow$ working power for communication;
7  switch $R$ do
8      case Transmitter or Receiver do
9          while within the arranged time window for communication do
10             $CXN = false$;
11             attempt to connect;
12             if connected successfully then
13                 $CXN = true$;
14                 break;
15         if CXN=false then
16             if R=Transmitter then
17                 $\Upsilon = G^t$;
18             else
19                 $\Upsilon = G^r$;
20             $Gearbox(\Upsilon, \sigma, P_w)$;      /* Algorithm IV.2 */
21     case Volunteer do
22         $\eta \leftarrow$ ratio between number of $TX$ and $RX$;
23         $R = Volunteer.Rand(TX, RX, \eta)$;
24         $\Upsilon = G^v$;
25         $Gearbox(\Upsilon, \sigma, P_w)$;      /* Algorithm IV.2 */
```

$Gearbox$ is the key module of ELIXIR which is called by TTC in Algorithm IV.1 when a connection is unsuccessful. Given the input parameters of a prime exclusive set of periods (Υ), the time required to establish the connection (σ), and the working power of the communication (P_w), Algorithm IV.2 presents the details of $Gearbox$.

The function of $Gearbox$ is self-explanatory, which acts as a gearbox for IoT devices to select appropriate gears (periods) so that the connection among IoT devices of different roles can be established. Since the periods for transmitters and receivers belong to mutually-disjoint and prime-exclusive sets Υ, whenever IoT devices with different roles select their own periods to connect each other, their periods are co-prime to each other. Based on Corollary 1.1, the communication pairs only need to satisfy $T_a\delta_a + T_b\delta_b - 2\sigma \geq 1$. This significantly facilitates the condition for a guaranteed connection.

At the beginning, $Gearbox$ initiates required parameters (line 1-3). Here, κ represents the number of primes in Υ; Θ is a struct that holds all the prime periods of Υ with corresponding power thresholds; Ψ is another struct that holds periods candidates with corresponding votes for election. After parameter initialization, as shown from line 4-6, $Gearbox$ first passes all elements (periods) from Υ into Θ. For each period $\Theta[i].T$, based on Theorem 1, $Gearbox$ calculates the minimum average harvesting power threshold $\Theta[i].p^{th}$ required for a guaranteed connection. The rationale behind the calculation is that, to satisfy Theorem 1, both pairs for connecting need to contribute ($T_a\delta_a + T_b\delta_b \geq 1 + 2\sigma$) and $Gearbox$ arbitrarily

Algorithm IV.2: Co-Prime **Gearbox**

```
   Input: $\Upsilon$, $\sigma$, and $P_w$
1  $\kappa \leftarrow$ the number of primes in $\Upsilon$;
2  $\Theta \leftarrow$ holds prime periods and corresponding power
     thresholds
3  $\Psi \leftarrow$ holds period candidates and corresponding votes
4  for $i$ from 0 to $\kappa - 1$ do
5      $\Theta[i].T = \Upsilon[\kappa - 1 - i]$;
6      $\Theta[i].P^{th} = \frac{P_w(1+2\sigma)}{2\Theta[i].T}$;
7  while connection hasn't been established do
8      $GearUP(\kappa, \Theta, \Psi)$;    /* Algorithm IV.3 */
9      $\psi_{id}^0 = \Psi[0].id$;
10     keep tracking of $t$ and $E_c$;
11     switch State do
12         case Sleep do
13             if $t\%T > T - \frac{1+2\sigma}{2}$ or $\gamma > \Theta[\psi_{id}^0 - 1].T$ then
14                 $\gamma = 0$;
15                 $T = \Theta[\psi_{id}^0].T$;
16             if $t\%T = T - E_c/P_w$ then
17                 $\gamma = \gamma + 1$;
18                 $State = Active$;
19                 wake up the system for connecting;
20         case Active do
21             if $t\%T = 0$ then
22                 $State = Sleep$;
23                 set the system into low-power mode;
```

requires them to contribute evenly (line 6).

While the connection hasn't been established, from line 8-10, $Gearbox$ calls $GearUP$ (Algorithm IV.3) to update the aforementioned parameters, providing the ID (Ψ_{id}^0) of the most preferable period ($\Theta[\Psi_{id}^0].T$) in struct Θ (line 9), and keeps tracking of time (t) and charged energy (E_c). When the device is in sleep mode, first, from line 14-15 depending on the quality of harvesting power, T needs to be updated for better performance and along with the update, γ will be reset to 0. Here, γ represents the number of continuous periods without updating on T. The rationale behind the update is that when the harvesting power is weaker than expected ($t > T - \frac{1+2\sigma}{2}$), T will be updated with a possible longer preferable period so that condition in Theorem 1 can be satisfied. When the harvesting power is better than expected ($\gamma > \Theta[\psi_{id}^0 + 1].T$), T will also be updated but for a possible shorter preferable period, which follows Corollary 1.2 to speed up the connection. Next, if charging period and working period (E_C/P_w) together equals to T, $Gearbox$ set $\gamma = \gamma + 1$ and wakes up the systems for connecting until the next charging period ($t\%T$), by then, $Gearbox$ will put the system into low-power mode for charging (line 23).

Algorithm IV.3: Gearbox Update (**GearUP**)

```
   Input: $\kappa, \Theta$, and $\Psi$
1  if $t\%\Delta_t = 0$ then
2      $\overline{P_c} \leftarrow$ the average charging power within $\Delta_t$;
3      if $\overline{P_c} \geq \Theta[n-1].P^{th}$ then
4          $id = n - 1$;
5      else if $\overline{P_c} < \Theta[0].P^{th}$ then
6          halt until $\overline{P_c} \geq \Theta[0].P^{th}$;
7      else
8          for $i$ from 1 to $\kappa - 1$ do
9              if $\Theta[i].P^{th} > \overline{P_c} \geq \Theta[i-1].P^{th}$ then
10                 $id = i - 1$;
11     $n \leftarrow$ the number of entries in set $\Psi$
12     for $i$ from $0$ to $n - 1$ do
13         $\Psi[i].votes = \Psi[i].votes - 1$;
14         if $\Psi[i].id = id$ then
15             $\Psi[i].votes = \Psi[i].votes + 2$;
16             remove all entries from $\Psi[i+1]$ to $\Psi[n-1]$;
17         if $i = n - 1$ and $\Psi[i].id \neq id$ then
18             $\psi_{id}^0 = \Psi[0].id$;
19             $\psi_{id}^{n-1} = \Psi[n-1].id$
20             if $\Theta[id].P^{th} < \Theta[\psi_{id}^0].P^{th}$ then
21                 remove all entries from $\Psi[0]$ to $\Psi[n-1]$;
22                 set $\Psi[0].id = id$ and $\Psi[0].votes = 1$;
23                 break;
24             else if $\Theta[id].P^{th} > \Theta[\psi_{id}^{n-1}].P^{th}$ then
25                 set $\Psi[n].id = id$ and $\Psi[n].votes = 1$;
26                 break;
27             else
28                 for $j$ from $1$ to $n - 1$ do
29                     $\psi_{id}^{j-1} = \Psi[j-1].id$;
30                     $\psi_{id}^{j} = \Psi[j].id$
31                     if
                         $\Theta[\psi_{id}^{j}].P^{th} > \Theta[id].P^{th} > \Theta[\psi_{id}^{j-1}].P^{th}$
                       then
32                         remove all entries from $\Psi[j]$ to
                             $\Psi[n-1]$;
33                         set $\Psi[j].id = id$ and $\Psi[j].votes = 1$;
34                         break;
35         if $\Psi[i].votes = 0$ then
36             remove the entry $\Psi[i]$;
```

E. Gearbox Update

The purpose of $GearUp$ is to update the selection of preferable periods based on the level of the harvesting power, so that $Gearbox$ in Algorithm IV.2 can select the appropriate periods participating in connection to address the synchronization problem. Given Υ, σ, and P_w as inputs, $GearUp$ is detailed in Algorithm IV.3.

Whenever $GearUP$ is called, it first checks whether it is time for an update ($t\%\Delta_t = 0$). Here Δ_t is a customized time interval, which can be adjusted to balance between performance and energy consumption. During the interval Δ_t,

$GearUP$ measures the average charging power ($\overline{P_c}$) which will be further compared with every entries of Θ to find out the ID (id) of the shortest period that can satisfy Theorem 1, which is shown from line 3-10. Then, $GearUP$ start to search all entries in Ψ from very first to the last to find out whether id has already been saved. Here each $\Psi^i_{id} = \Psi[i].id$ is used to indicate a entry $\Theta[\Psi^i_{id}]$ in Θ.

In the first case, if id has already been stored in $\Psi[i].id$, form line 14-16, the selection of $\Psi[i]$ will be enhanced with $\Psi[i].votes + 1$ and all rest of the entries ($\{\Psi[j] | i < j < n\}$) that hasn't been searched will be removed. The reason for such removal is that compared with $\Psi[i].id$, $\Psi[j].id$ leads to a smaller period which cannot satisfy the condition in Theorem 1 given measured $\overline{P_c}$. In the second case, if id is new to Ψ or has been removed before, from line 17-34, id will be reinserted which allows Psi to be sorted in ascending order of the corresponding $\Theta[\Psi^i_{id}].p^{th}$. Once $\Psi[j].id = id$ has been inserted, it needs to be enhanced as $\Psi[j].votes = 1$. In the end, whenever $\Psi[i].votes = 0$, $\Psi[i]$ will be removed.

From the above operation, Ψ is sorted automatically in ascending order of the corresponding $\Theta[\Psi^i_{id}].p^{th}$. Since we know that from Algorithm IV.2, $Gearbox$ always selects the ID (Ψ^0_{id}) from the very first entry $\Psi[0].id$ to indicate the the most preferable period ($\Theta[\Psi^0_{id}].T$), the $\Theta[\Psi^0_{id}].T$ will be the longest among all the other candidates. On the one hand, when $\overline{P_c}$ becomes stronger, $\Theta[\Psi^0_{id}].T$ will be gradually replaced by a short period to create a safe and smooth transition to help IoT devices to achieve a faster connection. On the other hand, when $\overline{P_c}$ becomes weaker, $\Theta[\Psi^0_{id}].T$ will be quickly replaced by a longer period to satisfy the condition in Theorem 1 for a guaranteed connection.

V. Experimental Evaluation

A. Experiment Setup

The proposed ELIXIR is a prototype design to facilitate existing communication protocol working under intermittent-powered conditions. In this paper, we take the very first step to set up a hardware platform for implementing and evaluating the performance of ELIXIR. The detailed hardware description and parameter configuration are shown as follows:

*1) **Hardware Platform**:* We implemented ELIXIR on TI's ultra-low-power evaluation board, MSP430FR6989 which consists of a 16-bit MCU, a 12-bit ADC, a 2KB volatile SRAM, a 128KB nonvolatile FRAM memory. Though SPI, MSP430FR6989 can control the external low-power transceiver CC1101 for communication. Figure 4 shows the detailed connection between CC1101 and MSP430FR6989 for a pair of communication nodes. The evaluation board is connected with a two-stage voltage regulator Bq25570 + LTC3459EDC which can supply a constant voltage of 3.3V to power the edge device and a maximum voltage of 4.2V to charge up a $470\mu F$ capacitor. During the experiments, $1MHz$ clock is configured internally to keep track of local time. We used a multimeter to measure the currents of both continuous transmission and receiving on the motherboard MSP430FR6989 with 3.3v input voltage. The measured currents are 10.2 mA and 8.0 mA respectively. Hence the transmission power is 33.7 mW and receiving power is 26.4 mW. In order to synchronize successfully, during the overlapping active time σ, the preamble and sync word should be picked up successfully by the receiver. Therefore, by configuring with 250K baud rate, the $\sigma = 0.025ms$ is employed for ELIXIR to conduct proper calculation and selection of periods.

*2) **Time Tracking Mechanism**:* To keep track of time during reconnection cycles, we use the real-time clock (RTC) module to generate time. Based on the level of the harvesting power, the ELIXIR will guide the MCU to choose the appropriate duty cycle switching between working and low-power mode (LPM). Under the LPM, the RTC with such a low frequency consumes negligible power of about $1.25\mu W$. The RTC module will keep functioning as long as the input voltage of the MSP430FR6989 is above the Brown Out Reset (BOR) threshold which is $1.8V$. When the harvesting power is recovered, BOR is less likely to happen as the duty cycle for reconnection is determined by the intensity of the harvesting power based on ELIXIR. However, in case the harvesting power is unable to support LPM such as when the solar panels are in the evening and vibration stop on piezoelectric generators (PEG), the voltage on the capacitor will gradually deplete. Once the voltage on the capacitor becomes too low for the power regulator to maintain a stable supply voltage for the MSP430FR6989, the supply voltage will quickly drop down below the BOR threshold resulting in a loss of time.

*3) **Energy Tracking Mechanism**:* We use RTC to trigger periodical ADC sampling in both LPM and working modes. To minimize the energy consumption, we activate the ADC interrupt so that once the ADC produces a result to the register, the corresponding interrupt will be triggered for further simple yet critical decision-making. To balance between accuracy and energy conservation, we set the sampling frequency of $2Sa/s$. After deployment, the monitoring of voltage on the capacitor can help track the residual energy E_c and calculate the average charging power $\overline{P_c}$ of MSP430FR6989 to update the appropriate prime periods by ELIXIR. Regarding the parametric update of charging periods in Algorithm IV.3, the updating periods Δ_t is a customized value to balance between the performance and energy consumption. This experiment selects $5s$ as an updating interval. Although we cannot track the detailed power consumption of ELIXIR during runtime, based on the connection speed, we can conclude that the energy overhead of ELIXIR has been well paid off by the improved performance for reconnection by comparison.

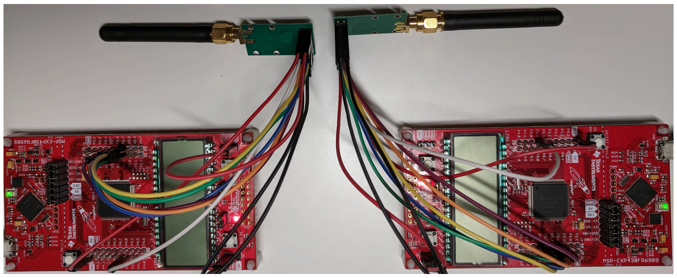


Fig. 4. MSP430FR6989 (MCU) + CC1101(Transceiver).

*4) **Power Trace:*** In total, eight power traces are generated using a signal generator. Without loss of generality, eight traces are categorized into two types. In Figure 5, the first four traces (1-4) are with stable voltages and the other four traces (5-8) are with unstable voltage, which are shown in Figure 6. The eight power traces with different power magnitudes are recorded from oscilloscope at the sample rate of $240KSa/s$. The maximum power that the regulator can provide under each power trace is also given in both figures. As we can see, the harvesting power is far less than the working power of edge devices.

*5) **Co-Prime Pools:*** Co-prime pools are essential to ELIXIR. It is necessary to ensure that the periods picked by the transmitter, receiver, and volunteers will always be co-prime to each other while maintaining a similar distribution for the sake of time fairness. To this end, for the first 100 prime numbers sorted in ascending order, we first select 20 prime numbers evenly across these 100 numbers for the volunteer pool; then the rest of the prime numbers are distributed into two co-prime pools where the odd positions are assigned to the transmitter and even positions are assigned to the receiver.

*6) **Baseline Methods:*** In the following set of experiments, Two baselines are used for comparison. The FIND [7] is a stochastic asynchronous connection method and similar methods have been adapted by industrial low-power solution such as CC3220SF from TI. The DISCO [8] is a research method that has been used to save energy for asynchronous connection based on Chinese Reminder theory.

B. Experimental Evaluation

In the following set of experiments, we target on a pair of asynchronous nodes (transmitter and receiver). First we show the statistic of connection time under different power traces. Then we analyze the time and energy overhead for them to establish the connection autonomously with proposed ELIXIR and baseline methods. Finally, we discuss the influence of the time unit to the overall performance.

*1) **Connection Time:*** For each power trace given in subsection V-A4, more than 500 sets of connection time measurements are logged for each comparison method through serial port. First, we evaluate the performance of ELIXIR when the harvesting power is stable which also can prove the effectiveness of the proposed Theorem 1. Then, we further evaluate the robustness of ELIXIR under the dynamic harvesting power scenarios. The following histograms in Figure 7 shows the comparison statistical results of connection time under different power traces.

In Figure 7, the results of the baseline methods FIND and DISCO are represented with red and yellow colors, and the proposed ELIXIR is represented with blue color. We observe that, when the harvesting power is stable, ELIXIR reveals its superiority to deliver a faster connection over the baselines. What's even more appealing is that with a limited amount of prime values of periods for selection, when the power is stable, ELIXIR presents a salient upper boundary of connection time which is in accordance with theorem 1. Notice that DISCO also demonstrates a similar upper boundary as ELIXIR but with a far longer connection time. This is because DISCO only wakes up the device for a single cycle with one connection attempt in each prime period. Since ELIXIR fully utilizes the harvested energy to remain active for requesting and listening during the entire active period of $T\delta$, it increases the chance of overlapping wake-up events. For FIND, due to the innate random behavior, it requires a much longer connection time.

*2) **Influence of Static Power:*** We can observe under with static power (trace 1-4) that when the harvesting power is extremely weak such as shown in Trace 1, 2, and 3 (Figure 5), the ELIXIR tends to select a larger period for connection to satisfy the condition for a guaranteed connection which is given in Theorem 1. When the harvesting power is good such as the power trace 4, ELIXIR will pick smaller periods to speed up the connection based on Corollary 1.2. Clearly, ELIXIR demonstrates a significant performance improvement over baseline methods. Specifically, within 10s, when harvesting power is weak (trace 1, 2, and 3), on average, ELIXIR has a 55.77% chance to connect successfully, while the chances for baseline FIND and DISCO to connect successfully are 8.33% and 0.11% respective. When harvesting power is good (trace 4), on average, ELIXIR has a 96.73% chance to connect successfully, while the chances for baseline FIND and DISCO to connect successfully are 34.29% and 1.42% respectively.

*3) **Influence of Dynamic Power:*** After evaluating ELIXIR with static power. We further evaluate the robustness of ELIXIR with dynamic power. From Figure 6, we can observe that even when the harvesting power is unstable, ELIXIR still delivers a much faster connection time compared with the baselines. The rationale behind such superiority is that even if power fluctuation happens very often, especially in a more realistic scenario, such fluctuation will not affect the

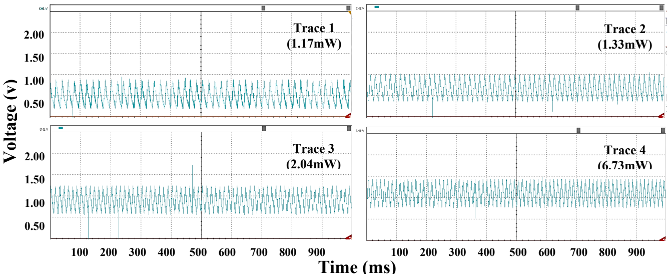


Fig. 5. Static Energy Harvesting Power Traces.

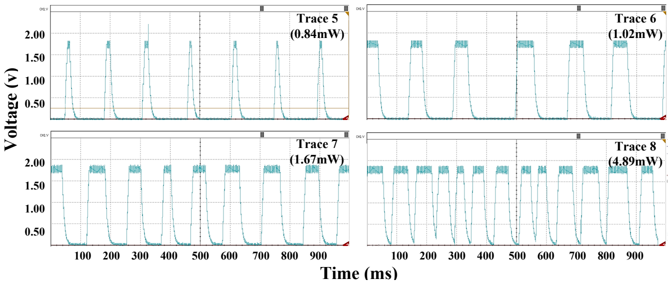


Fig. 6. Dynamic Energy Harvesting Power Traces.

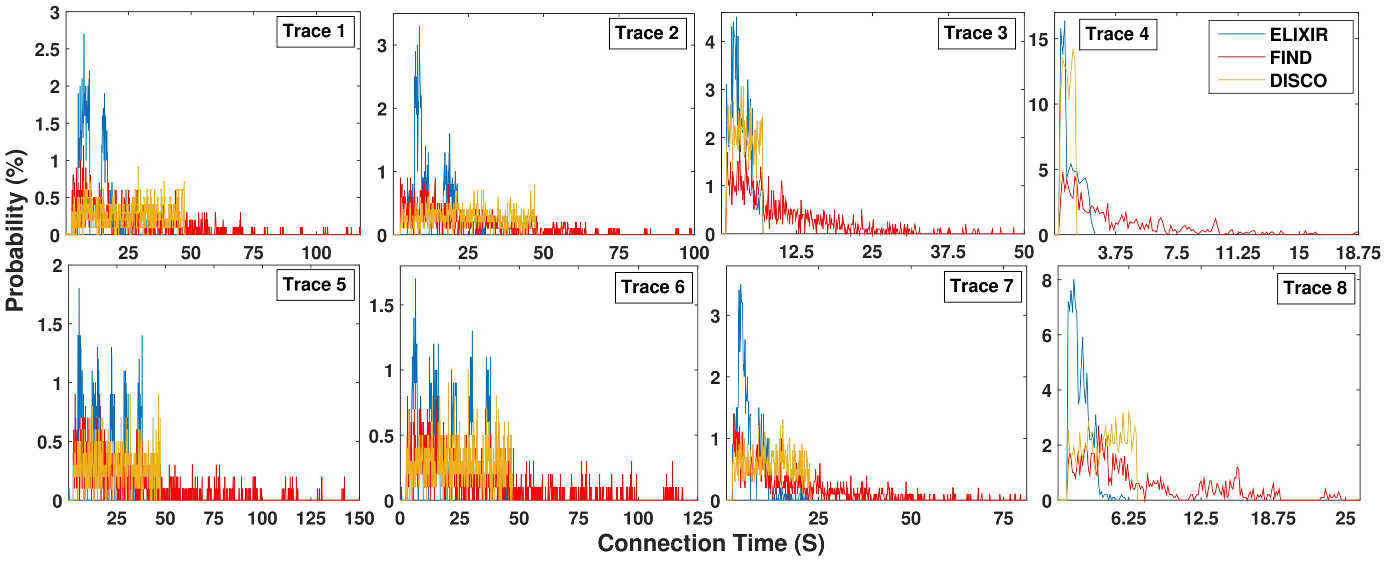


Fig. 7. Connection Probability.

Theorem 1, Corollary 1.1, and 1.2 to take effect. The reasons are bi-fold, first, the edge device never uses the harvesting power directly when harvesting power is weak and unstable. It always charges the energy reservoir such as a supercapacitor first. And then the energy from the reservoir will be further harnessed by a power regulator to power edge devices with stable voltage. Second, in the long run, the harvesting power is relatively stable, and such fluctuation will be cushioned by the energy reservoir. These two facts create a friendly environment for ELIXIR to take effect.

*4) **Average Connection Time&Energy**:* From Figure 8, we can observe that compared with baselines, ELIXIR demonstrates a significant advantage to connect each other in an expedited way. Overall, ELIXIR shows magnitudes of improvements in speeding up the connection compared with baselines. Specifically, ELIXIR help establish the connection $2.83X$ and $1.84X$ faster than FIND and DISCO respectively. Since under each comparison, ELIXIR, FIND, and DISCO are with the same power traces, the corresponding improvement in energy efficiency is the same as the improvement of connection time.

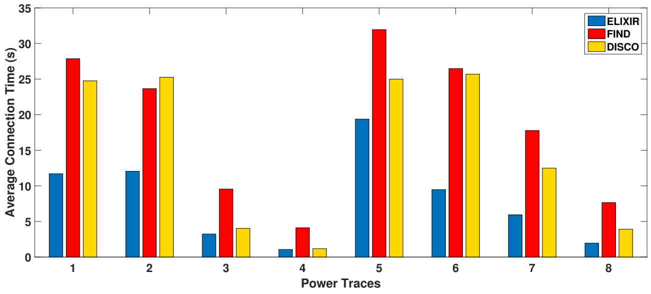


Fig. 8. Average Connection Time.

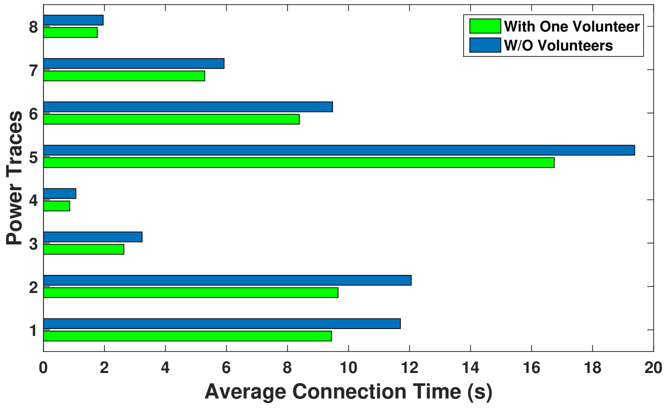


Fig. 9. Influence of Volunteer to the Average Connection Time.

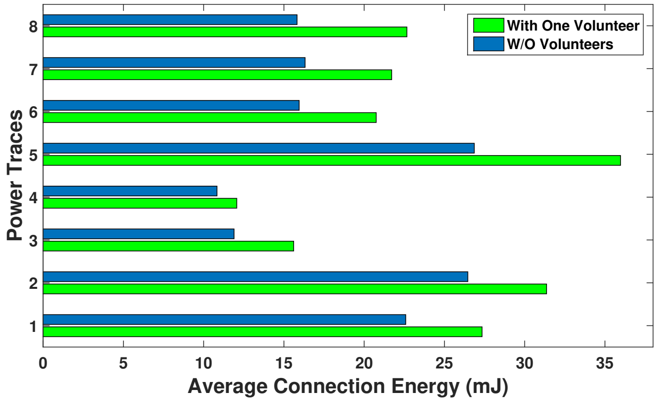


Fig. 10. Influence of Volunteer to the Average Connection Energy.

*5) **Neighbor's Help**:* Previous methods consider the performance of ELIXIR with only communication partners involved. In the following, the performance of ELIXIR is further evaluated with the participation of neighbors for help. Here we add the third MSP430FR6989 with CC1101 acting as a volunteer, which creates a scenario where each communication pair has one neighbor who can spare 50% of its time volunteering randomly as transmitters and receivers. Under such settings, we evaluate the performance of ELIXIR in terms of time and energy consumption for connection, and results are demonstrated in Figure 9 and 10 respectively. Figure 9 compares the expected connection time. We can observe that with the extra effort of all neighbors, the communication pairs' connection and synchronization to the main edge network are slightly faster. Specifically, the connection can further speed up 18.4% on average when the harvesting power is static (trace 1-4). While in a more realistic scenario with dynamic harvesting power (trace 5-8), the connection can further speed up by 12.1% on average. Figure 10 shows the comparison of the expected energy overhead. As we can observe that compared with the improvement of time efficiency, the participation of neighbors requires more effort in energy. Specifically, the connection consumes an extra 20.6% overall energy when the harvesting power is static (trace 1-4). While in a more realistic scenario with dynamic harvesting power (trace 5-8), the connection consumes an extra 35.1% overall energy on average. Considering the large extra energy cost, there is only a slight improvement in synchronization time. However, the neighbors' participation is very significant to prevent the victim CPs from isolating the rest of the IoT edge network when their synchronized timeline has deviated from the others. Such sacrifice of time and energy is totally worthwhile to improve the stability and efficiency of the entire self-powered IoT edge network as a whole.

VI. Related works

This section discusses related works on guaranteeing forward progress of computation and I/O tasks in standalone self-powered devices, and how to continue communication and re-connection of those devices when interrupted by power failures.

A. Self-powered Devices

Although energy harvesting is very promising for next-generation IoT devices, the harvested energy [12], [13] is weak and transient that impedes its wide adoption in IoT devices.

1) Checkpointing Computation Tasks: To guarantee forward progress of computation tasks, researchers have proposed to save intermediate results into NVM upon power failures [14]–[17] by leveraging the non-volatility of non-volatile memory (NVM). In addition, several studies [18]–[25] reported a novel non-volatile processor design with ferroelectric non-volatile registers that have the access efficiency close to that of SRAM [26] with superior endurance. Such non-volatile processors can more efficiently store the intermediate run-time states [27].

2) Checkpoinging I/O Tasks: To address the issues of frequent state loss of volatile I/O modules and repeated identical operations under power failures, [28] presents the concept of autonomous I/O which can accumulatively and transparently complete I/O operations regardless of intermittent power supply. Besides, [29] presents a checkpointing system that safely supports peripheral operations. This system correctly runs user-annotated peripheral functions by selectively disabling checkpoints and undo-logging.

Besides guaranteeing forward progress, researchers have addressed data inconsistency issues between volatile and non-volatile memory across power failures [30]–[32]. Specifically, the proposed checkpointing system in [29] can avoid inconsistency between peripheral and program state. There is little research on intermittent networks of multiple nodes. [33] leveraged existing data copies inherently created in the network to mitigate data unavailability and inconsistency due to intermittency.

B. Communication between Self-powered Devices.

Communication modules of self-powered devices are the most energy-consuming on self-powered devices. On-going communication between two devices may be interrupted which will cost tremendous effort to rebuild the connection once there is a power outage on one device. While communication between self-powered devices is significant in developing self-powered IoT networks, there is limited research on intermittent transmission with energy harvesting power supply. The research can be grouped into three categories including timekeeper schemes, back-scatter, and neighbour discovery.

1) Timekeeper: One category of solutions tries to synchronize the transmission time of two devices by designing timekeeper schemes. [6], [34] proposed a timekeeper that tried to keep track of time even when the device is depleted of energy. the time elapsed between two reboots was estimated by discharging a small capacitor over a large resistor during execution and off-time and measuring the voltage. However, the resolution decreases as measured time increases and the timekeeper is hardly to have a longevity of multiple seconds. CHRT [35] cascaded those RC circuits together to improve resolution. Those methods require extra hardware to address the synchronization problem which will be expensive for industrial implementation.

2) Backscattering: The second category of intermittent communication research works integrates backscattering tags into the self-powered devices to learn the energy state of other devices. In this way, transmitters can prevent packet losses to unavailable receivers due to power failures. [36] proposed an intermittent communication protocol using a backscatter radio circuit to propagate each node's energy state, where transmitters can always be updated about the receiving nodes' availability. [37] proposed to equip each self-powered device with multiple radios including a backscattering tag and devised a smart mechanism to dynamically select the matched radio for communication-based on the energy state of devices. Despite the appealing features, backscatter is limited by the requirement of a centralized high radio frequency (RF) power source and short communication distance. RF power attenuation is extremely high if devices are far away from the power source.

3) Neighbour Discovery: Popular industrial low-power IoT solutions such as CC3220SF [11] from Texas Instrument provides alternative software way for connection. To save energy for low-power operation, CC3220SF randomly wakes up and attempts to connect instead of keeping the radio on until connected. Yet, such a random connection approach, although can end the "never-ending connecting" problem as mentioned in Section II, has unpredictable performance, which is uncontrollable to developers and always has a chance to have a very long connection time. Similarly, [7] proposed a neighbor discovery protocol for the self-powered wireless network. In this protocol, random waiting delays are introduced before turning on the radio. Despite better performance than without introducing any random delay, all the energy and time are wasted during the random waiting time. [8] proposed a neighbor discovery protocol to save energy for asynchronous connection based on Chinese Reminder theory. [38] used greedy communication policies to guarantee bounded worst-case communication latency for two self-powered devices by encoding types of energy-harvesting traces.

The proposed ELIXIR in this paper is designed to be flexible with maneuverable upper bounds of connection time to address the disrupted communication problem. ELIXIR is built upon a comprehensive theoretic basis to repair connection anomalies by conducting synchronization jointly and efficiently in an expedient way, which can be simply integrated into the existing communication framework to support transient communication under energy harvesting scenarios. This work also lays a foundation for building future self-powered IoT networks.

VII. Further Discussion

ELIXIR can enable fast self-healing of communication for the next-generation IoT network. This section discusses several considerations before properly integrating ELIXIR into existing self-powered IoT systems including the influence of ELIXIR on the original runtime operation, the necessity of checkpointing, and how to address potential data package collision when the network is scaled up.

A. Influence to Runtime Operation

During the reconnection stage when the power recovers, runtime tasks such as data forwarding, routine sensing, and in-situ data analysis need to be conducted. For implementation purposes, the harvested energy E_c should be partially budgeted for other runtime task execution and the remaining can be used for reconnection. In particular, we need to consider the communication task where a healthy node can also be a transmitter/receiver for multiple other nodes. Aside from exerting energy for ELIXIR after discovering its neighbor that loses synchronization, it also needs to fulfill the communication duty for other devices. To this end, we can properly adjust the algorithm parameter P_w by taking the energy consumption and the routine schedules of other runtime tasks into consideration. Besides, to address the time conflict, proper scheduling needs to be conducted among all tasks but prioritize the ELIXIR for fast reconnection. This is because, without a successful connection, the information is useless regardless of how accurate or critical it is.

B. Necessity of Checkpointing

As one of the standard modules of intermittent systems, checkpointing techniques such as compiler-level checkpoint insertion, periodical checkpointing, and voltage-triggered on-demand checkpointing have been widely used to maintain the runtime volatile state to accumulate the forwarding progress of computation and I/O tasks against power failures. Different from those tasks, ELIXIR program itself does not needs checkpointing. Specifically, the temporary variables of ELIXIR are the current selection of prime periods which, according to ELIXIR, are correlated to the level of the harvesting power. So even if we checkpoint the period, after power comes back, the current power intensity will definitely be much better than before the BOR happens. Thus, checkpointing the previous selection of period is unnecessary. In other words, if the execution of ELIXIR algorithm is interrupted by a power failure, nothing needs to the checkpointed. ELIXIR just needs to be restarted when power comes back.

C. Network Scalability

ELIXIR is designed with a good scalability potential for implementation into a large network enabling multiple self-powered IoT devices to synchronize automatically in a joint effort. Under the context of ELIXIR, any healthy node with good synchronization to the network (gateway) can become a volunteer, which is crucial for solving the isolated islanders' problem described in Section II. In particular, if nodes are within communication range and share a channel, ELIXIR can consequently maximize its effectiveness as, probabilistically speaking, more synchronized nodes will be available during their idle time in assisting with potential isolated islander problems. Single-channel architecture (SCA) remains useful today due to its low cost and simplicity of implementation, especially in environmental monitoring systems that utilize energy-harvesting sensors. These systems typically have limited data transfer requirements, making SCA-based communication modules a suitable choice as communications are less prone to heavy data collisions, ensuring efficient data transfer.

Yet, multi-channel architecture (MCA) is more widely used in scenarios where the potential frame collision problems of SCA can be easily addressed. This can be done by assigning nodes to different pass-band channels using frequency division multiplexing (FDM) or by employing state-of-the-art techniques such as orthogonal frequency division modulation (OFDM) or chirp spread spectrum modulation. These modulation techniques are widely used in various modern MAC protocols. Therefore, any device after network initialization will only operate on a dedicated channel. In this case, when volunteering or reconnecting, the node will also operate on the assigned channel. Nodes sharing the same channel can still fulfill the volunteer's role. However, it should be noted that compared to SCA, MCA-based applications generally require higher data rates and consume more power, such as home Wi-Fi and cellular networks. Besides, nodes within the radio range are highly unlikely to be assigned to the same channel. Therefore, the volunteering mechanism of ELIXIR may not be effective in MCA scenarios. Regarding collision avoidance, nodes that are under the same channel, regardless of SCA and MCA, typically use the CSMA/CA mechanism, which

This article has been accepted for publication in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. This is the author's version which has not been fully edi
content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2023.3266724

is embedded in most MAC protocol stacks, to automatically avoid data collisions.

VIII. Conclusion

This paper proposes ELIXIR, an expedient connection paradigm, to conduct synchronization swiftly and autonomously in a joint effort for self-powered IoT devices. ELIXIR is built with a strict and comprehensive theoretic basis to solve the synchronization problem expediently in a software approach. The significance of this work is that the lightweight design fashion and highly efficient feature makes ELIXIR especially appealing for existing communication frameworks to adapt to ubiquitous energy harvesting scenarios. Extensive experiments under diversified power traces show that ELIXIR outperforms baseline methods $2.83X$ and $1.84X$ in terms of connection time.

References

[1] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *IEEE SoutheastCon 2008*. IEEE, 2008, pp. 442–447.

[2] G. Gobieski, B. Lucia, and N. Beckmann, "Intelligence beyond the edge: Inference on intermittent embedded systems," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019, pp. 199–213.

[3] H. Jayakumar, A. Raha, and V. Raghunathan, "Quickrecall: A low overhead hw/sw approach for enabling computations across power cycles in transiently powered computers," in *2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems*. IEEE, 2014, pp. 330–335.

[4] K. Maeng, A. Colin, and B. Lucia, "Alpaca: Intermittent execution without checkpoints," *Proceedings of the ACM on Programming Languages*, vol. 1, no. OOPSLA, pp. 1–30, 2017.

[5] K. Ma, X. Li, M. T. Kandemir, J. Sampson, V. Narayanan, J. Li, T. Wu, Z. Wang, Y. Liu, and Y. Xie, "Neofog: Nonvolatility-exploiting optimizations for fog computing," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, 2018, pp. 782–796.

[6] J. Hester, N. Tobias, A. Rahmati, L. Sitanayah, D. Holcomb, K. Fu, W. P. Burleson, and J. Sorber, "Persistent clocks for batteryless sensing devices," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 15, no. 4, pp. 1–28, 2016.

[7] K. Geissdoerfer and M. 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 21)*, 2021, pp. 439–455.

[8] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 71–84.

[9] Y. Zhu, B. Yang, M. Liu, and Z. Li, "Pharos: A rapid neighbor discovery algorithm for power-restricted wireless sensor networks," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2019, pp. 1–9.

[10] B. Han, J. Li, and A. Srinivasan, "On the energy efficiency of device discovery in mobile opportunistic networks: A systematic approach," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 786–799, 2014.

[11] "Cc3220sf technique documentation," https://www.ti.com/product/CC3220SF#tech-docs, accessed: 2020-08-08.

[12] J. Taneja, J. Jeong, and D. Culler, "Design, modeling, and capacity planning for micro-solar power sensor networks," in *Proceedings of the 7th international conference on Information processing in sensor networks*. IEEE Computer Society, 2008, pp. 407–418.

[13] Q. Khan and S. Bang, "Energy Harvesting for Self Powered Wearable Health Monitoring System," *Health, San Francisco*, pp. 1–5, 2009. [Online]. Available: https://www.techonline.com/tech-papers/energy-harvesting-for-self-powered-wearable-health-monitoring-system/

[14] S. Kawai, A. Hosogane, S. Kuge, T. Abe, K. Hashimoto, T. Oishi, N. Tsuji, K. Sakakibara, and K. Noguchi, "An 8kb eeprom-emulation dataflash module for automotive mcu," in *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*. IEEE, 2008, pp. 508–632.

[15] A. Mirhoseini, E. Songhori, and F. Koushanfar, "Automated checkpointing for enabling intensive applications on energy harvesting devices," in *2013 IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2013, pp. 27–32.

[16] B. Ransford, S. S. Clark, M. Salajegheh, and K. Fu, "Getting things done on computational rfids with energy-aware checkpointing and voltage-aware scheduling." in *HotPower*, 2008.

[17] B. Ransford, J. Sorber, and K. Fu, "Mementos: system support for long-running computation on rfid-scale devices," *ACM SIGPLAN Notices*, vol. 47, no. 4, pp. 159–170, 2012.

[18] S. Ducharme, T. J. Reece, C. Othon, and R. K. Rannow, "Ferroelectric polymer langmuir-blodgett films for nonvolatile memory applications," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 4, pp. 720–735, 2005.

[19] M. Zwerg, A. Baumann *et al.*, "An 82ua/mhz microcontroller with embedded feram for energy-harvesting applications," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 2011, pp. 334–336.

[20] Y. Horii, Y. Hikosaka *et al.*, "4 mbit embedded fram for high performance system on chip (soc) with large switching charge, reliable retention and high imprint resistance," in *International Electron Devices Meeting*, 2002, pp. 539–542.

[21] H. Nakamoto, D. Yamazaki *et al.*, "A passive uhf rf identification cmos tag ic using ferroelectric ram in 0.35-um technology," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 101–110, 2007.

[22] Y. Wang, Y. Liu *et al.*, "A 3us wake-up time nonvolatile processor based on ferroelectric flip-flops," in *2012 Proceedings of the ESSCIRC,*, 2012, pp. 149–152.

[23] Y. Wang, Y. Liu, Y. Liu, D. Zhang, S. Li, B. Sai, M.-F. Chiang, and H. Yang, "A compression-based area-efficient recovery architecture for nonvolatile processors," in *Design, Automation Test in Europe Conference Exhibition*, 2012, pp. 1519–1524.

[24] Y. Wang, Y. Liu *et al.*, "Pacc: A parallel compare and compress codec for area reduction in nonvolatile processors," pp. 1–1, 2013.

[25] X. Sheng, Y. Wang *et al.*, "Spac: A segment-based parallel compression for backup acceleration in nonvolatile processors," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, 2013, pp. 865–868.

[26] H. Shiga, D. Takashima *et al.*, "A 1.6 gb/s ddr2 128 mb chain feram with scalable octal bitline and sensing schemes," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 142–152, 2010.

[27] J. Wang, Y. Liu, H. Yang, and H. Wang, "A compare-and-write ferroelectric nonvolatile flip-flop for energy-harvesting applications," in *2010 International Conference on Green Circuits and Systems (ICGCS)*, June 2010, pp. 646–650.

[28] Y.-C. Lin, P.-C. Hsiu, and T.-W. Kuo, "Autonomous i/o for intermittent iot systems," in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2019, pp. 1–6.

[29] K. Maeng and B. Lucia, "Supporting peripherals in intermittent systems with just-in-time checkpoints," in *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2019, pp. 1101–1116.

[30] M. Hicks, "Clank: Architectural support for intermittent computation," *ACM SIGARCH Computer Architecture News*, vol. 45, no. 2, pp. 228–240, 2017.

[31] K. Maeng, A. Colin, and B. Lucia, "Alpaca: Intermittent execution without checkpoints," *arXiv preprint arXiv:1909.06951*, 2019.

[32] A. Colin and B. Lucia, "Chain: tasks and channels for reliable intermittent programs," in *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, 2016, pp. 514–530.

[33] W.-C. Tsai, W.-M. Chen, T.-W. Kuo, and P.-C. Hsiu, "Intermittent-aware distributed concurrency control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[34] A. Rahmati, M. Salajegheh, D. Holcomb, J. Sorber, W. P. Burleson, and K. Fu, "{TARDIS}: Time and remanence decay in {SRAM} to implement secure protocols on embedded devices without clocks," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, 2012, pp. 221–236.

[35] J. de Winkel, C. Delle Donne, K. S. Yildirim, P. Pawełczak, and J. Hester, "Reliable timekeeping for intermittent computing," in *Proceedings*

of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, 2020, pp. 53–67.

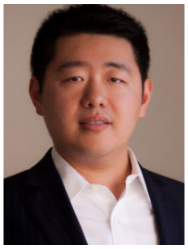
[36] A. Torrisi, D. Brunelli, and K. S. Yildirim, "Zero power energy-aware communication for transiently-powered sensing systems," in *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 43–49.

[37] Y. Wu, Z. Jia, F. Fang, and J. Hu, "Cooperative communication between two transiently powered sensor nodes by reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 1, pp. 76–90, 2021.

[38] K. Wardega, W. Li, H. Kim, Y. Wu, Z. Jia, and J. Hu, "Opportunistic communication with latency guarantees for intermittently-powered devices," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1093–1098.

Chen Pan (S'13-M'20) received M.S. degree in Electrical Engineering from Oklahoma State University in 2017 and the PhD degree in Electrical and Computer Engineering from University of Pittsburgh in 2019. He is currently an assistant professor with the Department of Computing Sciences at Texas A&M University-Corpus Christi. His current research interests include Sustainable and Intelligent IoT Systems, Edge-driven AI, low power embedded systems, and emerging non-volatile memories.

Wen Zhang received the B.E. degree from Chang' an University, China, in 2017. She is currently a Ph.D student at Texas A&M University-at Corpus Christi, USA. Her research interests are in Self-sustaining IoT system, Deep Reinforcement Learning for energy efficient IoT system, and Intelligent Sparsity-Aware Sensing for Low-power IoT devices.



Yanzhi Wang is currently an Assistant Professor at the Department of ECE at Northeastern University, Boston, MA. His research focuses on model compression and platform-specific acceleration of deep learning architectures, maintaining the highest model compression rates on representative DNNs since 09/2018. His work on AQFP superconducting based DNN acceleration is by far the highest energy efficiency among all hardware devices. His recent research achievement, CoCoPIE, can achieve real-time performance on almost all deep learning applications using off-the-shelf mobile devices, outperforming competing frameworks by up to 180X acceleration. He received the U.S. Army Young Investigator Program Award (YIP), Massachusetts Acorn Innovation Award, Ming Hsieh Scholar Award, and other research awards from Google, MathWorks. etc.

Mimi Xie (S'13-M'19) received the BE and MS degrees from the College of Computer Science, Chongqing University, Chongqing, China, in 2010 and 2013, respectively and the PhD degree in Electrical and Computer Engineering from University of Pittsburgh in 2019. She is currently an assistant professor with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX. Her current research interests include energy harvesting embedded Systems and AI on Edge.