This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3154355, IEEE Transactions on Network Science and Engineering

Energy Stimulated Time Synchronization for Energy Harvesting Wireless Networks

Yu Luo, *Member, IEEE,* Lina Pu, *Member, IEEE,* and Zheng Peng, *Member, IEEE*

Abstract— **In this work, we develop a new energy-preserving time synchronization method, called the energy stimulated time sync (ESTS), for ultra-low-power wireless networks with energy harvesting capabilities. Compared with existing methods, ESTS does not rely on timestamp exchange but uses short energy tones to synchronize the nodes within the receiving range. ESTS is a centralized synchronization method based on a fully-passive wake-up radio (WuR) circuit, which can provide a variety of synchronization accuracy to meet different application requirements while consuming much lower energy than timestamp-based synchronization methods. We have implemented ESTS on Microchip ATmega256RFR2 system-on-chip (SoC). According to experiment results, ESTS can achieve an average of 5 ms and 70 μs time precision for rough synchronization and fine synchronization, respectively. Moreover, the fully-passive WuR circuit consumes zero energy in the idle status. In the entire synchronization process, the rough synchronization only needs to be performed once, and the node consumes as low as 4 μJ of energy for each fine synchronization, making ESTS a promising solution for ultra-low-power wireless networks.**

Index Terms—**Time synchronization, wake-up radio receiver (WuR), energy harvesting, ultra-low-power wireless networks**

I. Introduction

In recent years, the dramatic growth of wireless devices in the emerging applications of smart city, precision agriculture, body area network (BAN), and Internet of Things (IoT) raises critical challenges on network sustainability and scalability [1], [2]. The energy harvesting is envisioned as a promising alternative to traditional batteries for low-power wireless networks [3], [4]. Given the limited power supply from the renewable energy, an energy harvesting node (EHN) may stay asleep most of the time and be briefly activated when communication or sensing is needed. The communicating parties thus must be synchronized, otherwise asleep receivers will miss messages from the sender.

The conventional time synchronization protocols rely on the timestamp exchange for clock skew and offset estimation [5]–[7], which may not work well in ultra-low-power energy harvesting wireless networks. In the initial synchronization stage of these protocols, nodes need to keep idle listening until the initial synchronization is finished. Therefore, nodes generally have heavy energy consumption in the initial phase of timestamp-based time sync methods. Compared with battery-powered wireless nodes, EHNs usually have very limited energy storage (e.g., tens of millijoules or less) and are therefore very sensitive to bursty energy consumption [8]–[10].

In order to mitigate the high energy consumption on idle listening, wake-up radio receivers (WuRs) have been widely used in low-power wireless networks [11], [12], where each wireless node is coupled with a fully-passive wake-up circuit. The wake-up circuit can generate an interrupt after receiving the tone signal to activate the sleeping node. With the assistance of the WuR, EHNs can stay in sleep mode and wait for the initial synchronization messages from other EHNs.

Although the WuR can prevent idle listening in time synchronization, its potential has not been fully exploited. Specifically, in the existing wireless network, WuR does not directly participate in the time synchronization process, but only serves as an auxiliary circuit to activate nodes for message reception. Therefore, EHNs still need to exchange timestamp messages to correct their clock skews and clock offsets, which greatly wastes the potential of WuR.

In this work, we develop an energy stimulated time sync (ESTS) method that directly uses the WuR circuit for time synchronization. ESTS is a centralized synchronization method. In ESTS, wireless nodes do not exchange timestamps, but only use single-frequency pulse signals called the radio frequency tones (RFTs) sent from a central node to synchronize nearby devices that receive RFTs. The RFT is responsible for two major tasks: (a) waking up sleeping nodes, and (b) serving as a synchronization signal. The second task of RFT has not been studied in conventional time synchronization mechanisms.

There are several challenges in applying ESTS in practice. First, due to the high propagation loss of radio waves in free space, the strength of the RFT received by an EHN is low. Therefore, how to reliably and effectively synchronize sleeping nodes through a weak RFT requires careful consideration. Second, due to the hardware characteristic of the WuR, there is a response delay between receiving the RFT and generating the wake-up interrupt. This delay is proportional to the incident power of RFT: the closer EHNs will be activated earlier than the distant nodes. The variations in response delay need to be compensated in ESTS. Third, the response delay is not constant on a given node, but is susceptible to the random variations in the wireless channel. As a consequence, the time it takes for the WuR circuit to activate the node becomes unpredictable. How to accurately estimate the response delay with low energy and time costs is critical, because it determines the synchronization precision of ESTS.

To handle the above challenging issues, we study the output characteristics of the wake-up circuit and model the wake-up signal through theoretical analysis and experiment

Y. Luo is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, 39762. e-mail: yu.luo@ece.msstate.edu

L. Pu is with Department of Computer Science, University of Alabama, Tuscaloosa, AL 35487. e-mail: lina.pu@ua.edu

Z. Peng is with Department of Computer Science, City College of New York, New York, NY 10031. e-mail: zpeng@ccny.cuny.edu

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

TABLE I
RESOURCE CONSUMPTION TO PREPARE FOR TIME SYNC INITIALIZATION

Source	Condition	Power density	Efficiency	Harvester size	Harvest rate (μW)	T_h
Radio (peak) [10]	DTV 470 - 610 MHz	0.46 μW/cm^2	30%	6 dBi	134.7	21.5 min
	GSM900 (BT) 920 - 960 MHz	1.93 μW/cm^2	30%	6 dBi	186.4	15.5 min
	GSM1800 (BT) 1805 - 1880 MHz	6.39 μW/cm^2	30%	6 dBi	161.0	17.9 min
Thermal [9]	Tmp Diff = 10 °C	3.2 nW/cm^2		400 cm^2	1.28	38 hr
	Tmp Diff = 22.5 °C	12.44 nW/cm^2		400 cm^2	4.98	9.8 hr
Solar [13], [14]	Indoor	1 mW/cm^2	15%	1 cm^2	150	19.3 min
	Outdoor morning (shaded by tree)	3 mW/cm^2	15%	1 cm^2	450	6.3 min
	Outdoor morning (unshaded)	20 mW/cm^2	15%	1 cm^2	3000	48.3 s
	Outdoor noon (shaded by tree)	10 mW/cm^2	15%	1 cm^2	1500	1.8 min
	Outdoor noon (unshaded)	60 mW/cm^2	15%	1 cm^2	9000	9.4 s
Piezo [15]	Human biomechanics	7.34 μW/cm^3		8 cm^3	58.7	49.5 min

measurements. Two strategies are proposed to accurately estimate the response delay of real EHNs. As will be discussed in the article, by selecting wake-up circuits with different sensitivities and response delays, ESTS can flexibly provide various degrees of time precision.

We have implemented ESTS on the off-the-shelf Microchip ATmega256RFR2 wireless board and Powercast P1110B radio energy harvester. According to the experiment results, ESTS can provide an average of 5 ms and 70 μs time precision with the rough and fine synchronizations, respectively. The long sampling time (40 μs) of the analog-to-digital (ADC) is the main reason that limits the performance of ESTS. By using a buffer op-amp and increasing the clock rate of ADC, we can expect 0.5 μs or even higher time accuracy.

The main contributions of our work are threefold:

a) A new time synchronization method called ESTS is developed. To the best of our knowledge, ESTS is the first work that uses WuR to perform time synchronization.
b) We propose two rough time synchronization strategies based on the output characteristics of WuR. In addition, a fine synchronization strategy is developed to further improve the precision of ESTS.
c) We have implemented ESTS on a commercial SoC, and comprehensively evaluated its performance through both theoretical analysis and extensive experiments.

The rest of the article is organized as follows: Section II introduces the related work and motivates ESTS. We introduce the RFT detection in Section III. Two rough synchronization strategies are presented in Section IV and Section V, respectively. In Section VI, the fine synchronization method is discussed. The experimentation platform built for ESTS is introduced in Section VII. We evaluate the performance of ESTS in Section VIII and conclude our work in Section IX.

II. Related Work

This section introduces related work. We first review the design of several WuRs. After that, we investigate time synchronization methods developed for wireless networks and discuss the motivation for WuR-based time synchronization.

A. WuR Design

In the past few decades, the use of WuR to reduce the power consumption of wireless nodes has been extensively studied. Generally speaking, WuR can be divided into three categories: fully-passive WuRs, fully-active WuRs, and semi-passive WuRs.

Fully-passive WuRs contain only passive components such as Schottky diodes, complementary metal-oxide-semiconductor (CMOS), and capacitors [16], [17]. Due to the low sensitivity of the fully-passive WuR, the strength of the received wake-up signal must be high (usually ≥ -15 dBm) in order to general a sufficient voltage (i.e., 1.5 V) that can successfully interrupt the SoC [18]–[20]. Therefore, the fully-passive WuR is commonly used in short-range applications like radio frequency identification (RFID) and remote control.

Fully-active WuRs contain many active components like amplifiers, oscillators, and active inductors. The sensitivity of fully-active WuR is much higher than that of fully-passive WuR, but at the cost of higher power consumption and longer interrupt delay. [21] introduces the design of a low-cost fully-active WuR, which operates in the 868 MHz frequency band. The sensitivity of the proposed WuR can reach -51 dBm, and the power consumption is about 50 μW. In order to further improve the sensitivity and energy efficiency, a CMOS-based WuR is presented in [22], which includes an envelope detector, a low-noise baseband amplifier, and a mixed signal correlation unit. The receiver can reach a sensitivity of -71 dBm at 868 MHz, while the power consumption is only 2.4 μW. However, the interrupt delay is as high as 7 ms, and a long wake-up signal is required to activate the node.

The only active component in semi-passive WuRs is the comparator with low offset voltage. Due to the small number of active components, the power consumption of semi-passive WuRs is much lower than that of fully-active WuRs. The authors in [23] designed a semi-passive WuR with -51 dBm sensitivity and 270 nW power consumption for wireless body area networks. The receiver works in the 434 MHz industrial, scientific and medical (ISM) band. It can successfully activate the Texas Instrument MSP430 microcontroller through the Gaussian on-off keying (OOK) pulse width modulated wake-up signal. A semi-passive WuR introduced in [24], called SμA-WuRx, is used to detect the wake-up signal of -45 dBm at

868 MHz. SμA-WuRx employs a multi-stage Dickson rectifier to boost the input voltage, and then utilizes a comparator to detect the amplified voltage for interrupt generation. The current consumption of SμA-WuRx is as low as 1 μA.

B. Time Synchronization in Wireless Networks

Time synchronization plays a critical role in the distributed wireless networks. To date, many popular time synchronization methods, such as the flooding time synchronization protocol (FTSP), the timing-sync protocol for sensor networks (TPSN), and the reference broadcast synchronization (RBS), have been developed for wireless sensor networks, the details of which can refer to [25]. In this section, we briefly introduce several wireless synchronization solutions related to our work that are proposed in recent years.

The authors in [26] proposed an energy-efficient coefficient exchange synchronization protocol (CESP) for wireless sensor network. Similar to RBS, CESP is also a receiver-receiver synchronization protocol. However, unlike the nodes in RBS exchanging timestamps to estimate the clock skew, nodes in CESP directly exchange coefficients of the clock model, thereby reducing communication overhead. In [27], a beaconless asymmetric energy-efficient time synchronization scheme (BATS) for multi-hop wireless sensor networks is proposed. In BATS, the clock skew correction and clock offset estimation are not performed by sensor nodes but offloaded to a head node that has a powerful processor and unlimited energy supply. This can significantly reduce the energy consumption on resource-constrained sensor nodes while maintaining high synchronization accuracy.

In [28], a slow-flooding-based time synchronization is developed to reduce the negative impact of waiting time on the synchronization accuracy when nodes propagate the timing information of the reference node hop-by-hop. This is achieved by using a clock speed agreement algorithm that forces all the nodes to run at the same speed. In order to reduce the hardware requirement on flooding-based time synchronization methods in a multihop network, an adaptive proportional-integral clock synchronization (PISync) is proposed [29]. In PISync, the clock offsets and the clock skew are compensated by using a proportional-integral (PI) controller, which does not store distinct time information. As a result, the required memory space and computational complexity of PISync is significantly reduced. In [30], a rapid-flooding time synchronization protocol called PulseSync is designed for large-scale wireless networks. PulseSync can provide better synchronization accuracy and shorter latency than FTSP. This is achieved by calibrating the message delay and the propagation delay, and using linear regression to estimate the root node's clock.

C. Motivation for WuR-based Time Synchronization

When a wireless node is sleeping, all major digital modules with no data retention requirements (e.g., ADC, amplifier, and transceiver) are disconnected from the main power supply to save energy. Therefore, the node cannot receive any packets while asleep. However, existing time synchronization methods [25]–[30] rely on timestamp exchange between the communication parties. This causes a paradox: on the one hand, the communication parties need to be synchronized first so that they can wake up at the same time to exchange timestamps; but on the other hand, a time synchronization process requires the communication parties to exchange timestamps first so that the clock skew and the clock offset can be estimated.

To solve the above problem, wireless node running conventional time synchronization protocol need to keep idle listening until they successfully receive the required messages for initial synchronization, which consumes a lot of energy. For battery-powered wireless networks, this is not a big problem because the initial synchronization is not frequent, and the energy consumption is negligible in the long run. However, the high energy consumption in the initial time synchronization becomes unaffordable for ultra-low-power energy harvesting wireless networks.

Let E_n be the energy needed for the sync initialization. According to the power consumption rate of the node at idle listening (i.e., 7 mA×2.5 V) [31], E_n will be around 175 mJ with 10 seconds sync initialization time. Considering the limited capacity of supercapacitor and the low energy harvesting rate as listed in Table I, it may take tens of minutes for EHNs to cumulate sufficient energy for synchronization initialization. We list the required preparation time, T_h, under different energy harvesting rate in Table I.

Assume that the valid power supply voltage of a wireless node ranges from V_I to V_H, then the supercapacitor should be able to provide at least E_n joules of energy when its voltage decreases from V_H to V_I. Let C_m be the minimum capacitance of the supercapacitor to support sync initialization. According to the energy storage equation of the capacitor, we have that

$$C_m = \frac{2E_n}{V_H^2 - V_L^2}. \tag{1}$$

The required capacitance is around 35 mF [32].

In order to reduce the requirement on supercapacitor capacity and the preparation time for time synchronization, it is necessary to simplify or remove the initialization process. This prompted us to propose a new WuR-based time synchronization method, called ESTS, for ultra-low-power energy harvesting wireless networks. In the new method, EHNs do not need to exchange any timestamp but use a short analog signal to synchronize nearby nodes.

Although WuR has been widely used in wireless devices, it does not directly participate in the time synchronization process. To the best of our knowledge, our ESTS [33] is the first work that uses the WuR instead of timestamp exchanges for time synchronization. ESTS is a fully-passive WuR based approach that achieves 5 ms accuracy in the rough synchronization mode [33]. In this article, we will design a new synchronization scheme called fine synchronization for ESTS. The fine synchronization uses a low-stage voltage multiplier to achieve higher synchronization precision. As will be introduced in Section VIII, the accuracy of ESTS is improved to

70 μs with fine synchronization. In [34], we developed a semi-passive WuR-based synchronization method, called WUR-TS. Different from fully-passive method, an active component (i.e., a low-offset voltage comparator) is added to WUR-TS for wake-up signal detection. The involvement of active component improves the sensitivity (i.e., 30 dBm higher than ESTS) at the cost of higher energy consumption. The idle power consumption of WUR-TS is about 3.2 μW, three times higher than ESTS.

III. Challenges of WuR-based Time Sync

Due to the high propagation attenuation, the received RFT is usually too weak to directly wake up a sleeping node. Therefore, we need to use a voltage multiplier to boost the output voltage of the wake-up circuit. The output of a well-designed circuit can achieve 1.7 V when the intensity of the incident radio waves is as low as -18.86 dBm [35].

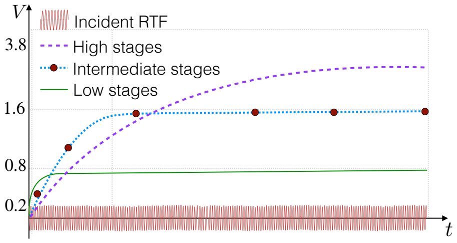


Fig. 1. Output characteristics of fully-passive wake-up circuits.

In order to reliably and quickly wake up an EHN, it is important to choose an appropriate number of stages in the voltage multiplier, as there is a trade-off between the rise time[1] and the output voltage [36]. As illustrated in Fig. 1, a higher stage voltage multiplier can produce a higher output voltage at the cost of longer rise time. Therefore, increasing the stages of the voltage multiplier can greatly improve the synchronization reliability and extend the distance, but it will take a longer time for the wake-up circuit to generate a high voltage interrupt to activate a sleeping node. In practice, this response delay is not a constant, but varies with the strength of the received RFT. As a consequence, with distance heterogeneity, different RFT receivers will be activated at different time.

We use Powercast P1110B radio energy harvester to implement the wake-up circuit and show how its output voltage changes with the intensity of the incident RFT in Fig. 2. We can observe that if the received RFT is weak, the output voltage of P1110B is low and the rise time of the wake-up signal is long. When the incident RFT becomes strong, the P1110B can generate a high voltage in a relatively short time. To successfully wake up an EHN, the interrupt voltage must be higher than a certain threshold (i.e., 1.3 V for ATmega256RFR2). As shown in the figure, when the incident power of the RFT is -7.23 dBm, it takes 60 ms for P1110B to produce a valid interrupt. However, if the incident power increases to -1.21 dBm, the circuit only needs 10 ms to generate the interrupt.

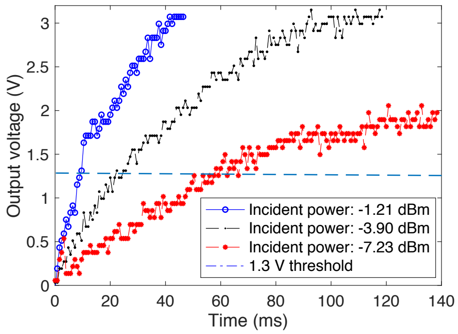


Fig. 2. Impact of RFT's strength on output of P1110B.

From the above analysis, it can be realized that the RFT may wake-up EHNs at completely different time. For example, suppose two nodes are 50 m and 100 m away from the central node. According to the free space path loss (FSPL) of radio waves in the open area and the measurement results shown in Fig. 2, a closer EHN will be activated 100 ms earlier than the farther node. This varying response delay will add a large uncertainty on the response delay, which poses a grand challenge in the use of the WuR circuit for synchronization. In the following sections, we will introduce two synchronization solutions that utilize the output characteristics of the fully-passive wake-up circuit to compensate for the uncertainties in the response delay.

Although we use Powercast P1110B as an example in the synchronization implementation, ESTS can be generally applied to any wireless nodes that use fully-passive WuRs. If the hardware is changed, some parameters in Section IV and V may need to be re-calibrated before the network deployment.

IV. NLS-Based Rough Sync

In this section, we first introduce a non-linear least squares (NLS) based method for ESTS. Then we discuss the limitation of the the NLS-based time synchronization approach.

A. Mechanism of NLS-Based Synchronization

Although the wake-up signals stimulated by an RFT can have different shapes, their starting points are the same. This is because in a fully-passive wake-up circuit, the switching time of diodes is short (nanoseconds) and the capacitance of capacitors is small (ten picofarads). This allows the wake-up circuit to quickly respond to a weak input at the beginning. Based on this observation, we realize that if we know the exact start time of the wake-up process, we can eliminate the uncertainties in the response delay.

To achieve the above objective, a model is needed to describe the wake-up process. We know that the fully-passive wake-up circuit can be considered as a resistor-capacitor (RC) circuit, where the resistance (R) and the capacitance (C) are

[1] Rise time is the time taken for the output voltage of a wake-up circuit to rise from 10% to 90% of the final value.

not constants but are variables related to the strength of the incident RFT. Accordingly, we can use the following capacitor's charging model to describe the wake-up process [37]:

$$V_o(t) = V_m\left(1 - e^{-\frac{(t-t_s)}{\tau}}\right), \tag{2}$$

where $V_o(t)$ is the instant voltage at time t; V_m and $\tau = RC$ are the open-circuit voltage and the time constant of the wake-up circuit, respectively; t_s represents the time offset, which is the time when $V_o(t)$ is zero. In (2), both τ and V_m are affected by the strength of the received RFT.

To verify the accuracy of the above model, we conducted experiment using ATmega256RFR2 wireless board and P1110B energy harvester. After the ATmega256RFR2 being activated by the wake-up signal, it samples the output of P1110B and uses Levenberg-Maquardt iterative algorithm, one of the most popular NLS algorithms, to fit the sampling voltages with (2). As shown in Fig. 3, the fitted curves match the wake-up charging process very well, especially at a high RFT intensity (i.e., $-2.5\,\text{dBm}$ in Fig. 3). When the strength of the incident RFT is reduced to $-6.2\,\text{dBm}$, there is about 20 ms error between the estimated starting point from the fitted curve and actual starting point.

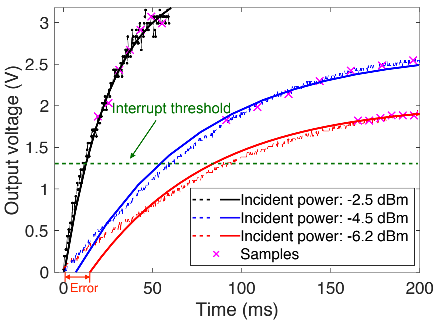


Fig. 3. Comparisons between the real wake-up processes and the NLS curve fitting results, where dot lines and solid lines are experimental data and fitted curves, respectively.

Based on the above analysis, we propose an NLS-based rough synchronization method for ESTS with the following steps:

1) The central node transmits an RFT of 150 ms length at local time, t_c.
2) Upon the reception of RFT, the output voltage of the wake-up circuit gradually rises. The EHN will be activated when the voltage reaches 1.3 V.
3) After activated, the EHN will periodically sample the charging voltage with time interval t_{in}. Let v_i be the voltage of the sample i and t'_i is the EHN's local time of the sample i, where $t'_i = t'_1 + (i-1)\,t_{in}$.
4) After getting N samples ($N \geq 3$), run the Levenberg-Maquardt algorithm with $\{v_1, v_2, \ldots, v_N\}$ and $\{t'_1, t'_2, \ldots, t'_N\}$ to calculate V_m, t_s, and τ in (2).
5) According to the definition of t_s in (2), the clock offset between the central node and RFT receiver is calculated as $t_s - t_c$. If we use the local time of the central node as the reference time of the network, then the EHN can estimate the reference time, which is denoted by $\hat{t}_c$, through $\hat{t}_c = t'_c - t_s$, where t'_c is the RFT receiver's local time.

B. Limitations of NLS-Based Rough Sync

When using NLS method to estimate the clock offset, the time and energy consumption of the EHN are significant. In Fig. 4, we show the time consumption of the NLS-based rough synchronization with clock rate of ATmega256RFR2 set at 1 MHz to save energy.

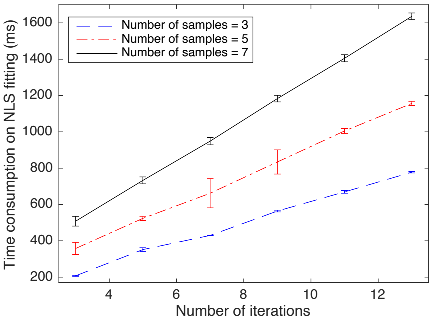


Fig. 4. Time consumption of the NLS-based rough synchronization. Each point is the average of 15 independent experiments.

As observed in Fig. 4, the time consumption of the NLS method is high due to the low computing speed of ATmega256RFR2. For example, when 3 samples of the charging voltage are used in Levenberg-Maquardt algorithm, it takes the microcontroller at least 200 ms for solving V_m, t_s, and τ in (2). In addition, the time consumption of the NLS method increases linearly with the number of iterations performed by the microcontroller. Specifically, for every two more iterations, the time consumption increases by around 60 ms, 80 ms, and 110 ms when $N = 3$, 5, and 7 samples are used for each clock offset estimation, respectively. In addition, given the number of iterations, the time consumption of the algorithm is doubled for every four more samples used for synchronization. For example, if $N = 3$ and the number of iterations is 5, the microcontroller only needs 352 ms to calculate the clock offset. However, if N is increased from 3 to 7, the algorithm's time consumption will increase to 732 ms.

Due to the high computational complexity of the Levenberg-Maquardt algorithm, the energy consumption of the NLS-based rough synchronization become unaffordable for ultra-low-power wireless devices. Specifically, when running the Levenberg-Maquardt algorithm, the average current of ATmega256RFR2 at 1 MHz clock rate is about 1 mA. Assume the supply voltage of the EHN is 2 V. Let both the number of iterations and samples be 3 for each synchronization, then the energy consumption for the clock offset estimation is over 0.41 mJ (i.e., $1\,\text{mA} \times 2\,\text{V} \times 207\,\text{ms}$). Referring to the energy harvesting rate listed in Table I, the EHN powered by radio, thermal, or piezoelectric energy will need a long time to collect enough energy for rough synchronization, making it unacceptable for some energy-constraint applications.

In order to improve the time and energy efficiency of the rough synchronization, we need a new method to estimate the clock offset, which will be introduced in the next section.

V. Mapping Table based Rough Sync

In order to prevent the high energy consumption on NLS, we develop a mapping table based rough synchronization method. We rewrite (2) as follows:

$$\Delta t = -\tau \ln\left(1 - \frac{V_0(t_1')}{V_m}\right), \tag{3}$$

where $\Delta t = t_1' - t_s$. Denote the intensity of the incident RFT by I_r. In (3), V_m and τ are functions[2] of I_r. Therefore, Δt can also be written as a function of I_r, i.e.,

$$\Delta t = \mathcal{F}(I_r). \tag{4}$$

Assume that the EHN gets the second voltage sample at the local time t_2'. Let $t_\delta \triangleq t_2' - t_1'$ and $\Delta V \triangleq V_o(t_2') - V_o(t_1')$, based on (2) we have that

$$\Delta V = [V_m - V_0(t_1')]\left(1 - e^{-\frac{t_\delta}{\tau}}\right). \tag{5}$$

Similar to (4), if we set t_δ as a constant, ΔV can be represented as a function of I_r , i.e.,

$$\Delta V = \mathcal{G}(I_r). \tag{6}$$

Due to the high propagation attenuation of RFT, I_r is usually very low. In this case, diodes in the wake-up circuit neither produce strong high-order harmonics nor conduct in the reverse direction since the voltage swings at the diode is low. Therefore, the power conversion efficiency of the voltage multiplier increases when I_r becomes large. As a consequence, V_m and τ changes monotonically as I_r increases. Accordingly, ΔV will be a monotonically increasing function of I_r; its inverse function, $\mathcal{G}(\cdot)$, will exist. Hence, we have that

$$I_r = \mathcal{G}^{-1}(\Delta V). \tag{7}$$

Substituting (7) into (6), it can be realized that Δt is a function of ΔV, i.e.,

$$\Delta t = \mathcal{H}(\Delta V). \tag{8}$$

Assume the resolution and the reference voltage of an ADC are m bits and V_{ref} volts, respectively. Denote the ADC code of $V_o(t_1')$ and $V_o(t_2')$ by A_1 and A_2, respectively. Let $\Delta A = A_2 - A_1$, then it can be obtained that

$$\Delta V = \frac{V_{ref}\,\Delta A}{2^m}. \tag{9}$$

Substituting (9) into (8), Δt can be represented as a function of ΔA:

$$\Delta t = \mathcal{Z}(\Delta A), \quad \Delta A = 1, 2, \ldots, 2^m - 1. \tag{10}$$

According to $\Delta t{=}t_1'{-}t_s$, the clock offset can be expressed as

$$t_s = t_1' - \mathcal{Z}(\Delta A). \tag{11}$$

Compared with (2), which requires at least three voltage samples to get the NLS solutions, (11) only needs to know ΔA to obtain the clock offset, t_s, thereby significantly reducing the computation overhead.

The one-to-one correspondence between ΔA and Δt allows us to describe their relationship through a mapping table. To create such a table, we set the ADC resolution to 8 bits and let the ATmega256RFR2 immediately sample the wake-up signal after being activated by the WuR. After t_δ microseconds, the ATmega256RFR2 samples the wake-up signal for the second time and then calculates ΔA. To obtain the corresponding Δt, we use a Tektronix TDS 7104 oscilloscope to measure the interval between the starting point of the wake-up signal and the time that ATmega256RFR2 takes the first sample, which will be detailed in Fig. 8 of Secion VII-B.

In (10), the expression of the function $\mathcal{Z}(\cdot)$ is unknown, therefore we gradually increase I_r in the experiment until getting all possible ΔA and corresponding Δt to create a complete mapping table. However, this approach is not convenient especially when the ADC resolution becomes high. To further simplify the mapping process, we only need to measure three pairs of Δt and ΔA, and then run the NLS algorithm with (12) to calculate parameters a_1, a_2, and a_3:

$$\Delta t = a_1 e^{-a_2 * \Delta A} + a_3. \tag{12}$$

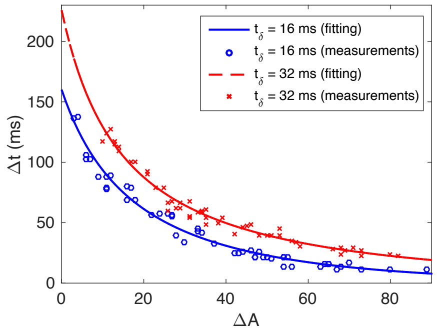


Fig. 5. Δt with respect to ΔA.

In Fig. 5, we compare the experimental data of Δt and ΔA with the NLS results. As shown in the figure, the results from NLS solution matches the measurements very well. After getting a_1, a_2, and a_3, we submit $\Delta A = 1, 2, \ldots, 2^m - 1$ into (12) to calculate the corresponding Δt. Now, the complete mapping table for the WuR circuit at a given t_δ is available, as listed in Table II, where $t_\delta = 32\,\text{ms}$. By looking up the mapping table, the EHN can obtain Δt immediately based on ΔA without any arithmetic operation.

The mapping table is determined by the hardware structure of the WuR circuit. Therefore for the same WuR, α_1, α_2, and α_3 are fixed and only need to be calculated once. This is quite different from the NLS-based rough time synchronization that we introduced in Section IV, where Levenberg-Marquardt

[2] The resistance and capacitance of a fully-passive WuR is not a constant, but changes with I_r, especially when I_r is small, as will be verified in Fig. 12. Therefore, the time constant τ becomes a variable related to I_r.

TABLE II
MAPPING TABLE FOR ROUGH SYNC

ΔA	1	...	21	22	23	24	25	...	255
Δt (ms)	210.9	...	84.7	81.2	79.3	76.8	74.5	...	2.3

algorithm runs in each rough synchronization since V_m, t_s and τ are not only determined by the WuR's hardware but also affected by the strength of the incident power I_r, which may vary in each rough synchronization.

Next, we summarize the mapping table based rough synchronization method for ESTS.

Preparations before network deployment:

1) At the initial network deployment, we first feed an EHN several RFTs with different strengths to obtain at least three pairs of Δt and ΔA.
2) Substitute the set of Δt and ΔA into (12), and then perform the NLS algorithm to obtain a_1, a_2, and a_3. Afterward, let $\Delta A = 1, 2, \ldots, 2^m - 1$, and then we can calculate Δt via (12) to create the mapping table for the EHN.

Rough synchronization process:

1) The central node broadcasts a 150 ms of RFT.
2) After receiving RTF, the output voltage of the wake-up circuit gradually rises. The EHN is activated when the output of WuR reaches 1.3 V.
3) The activated EHN immediately samples the output of WuR. The ADC code of the first sample is represented by A_1, and the local time when EHN gets the first sample is denoted by t'_1.
4) After t_δ microseconds, the node samples the output of WuR for the second time, and the ADC's sampling code is represented by A_2. Now, ΔA can be calculated via $\Delta A = A_2 - A_1$.
5) Afterward, the EHNs reads the mapping table and gets the Δt corresponding to ΔA.
6) Eventually, the clock offset between the EHN and the central node can be calculated through $t_s = t'_1 - \Delta t$.

The time and energy consumption of the mapping table based time synchronization is very small, because the EHN can directly get the current clock offset by reading the mapping table. The whole process can be done in several instruction cycles. Moreover, as will be evaluated in Section VI, the time accuracy of the mapping table based rough synchronization is higher than the NLS-based method, especially when the strength of the received RFT is weak.

VI. FINE SYNCHRONIZATION

In ESTS, the performance of rough synchronization is susceptible to the circuit noise and variation of wireless channels. If the channel changes significantly or there is a strong interference during RFT transmission, the EHN will not be able to accurately estimate the clock offset. As will be evaluated in Section VIII, the synchronization error of the mapping table based method is 5 ms on average, which may not be sufficient for high-precision applications. For this reason, we develop an efficient fine synchronization method for ESTS to improve the time accuracy.

In this section, we introduce the fine synchronization method first and explain how to use a low-stage voltage multiplier to improve the synchronization precision in ESTS. Afterward, we analyze the performance of the fine synchronization.

A. Details of Fine Synchronization

As depicted in Fig. 1 and Fig. 3, the long rise time with the high-stage voltage multiplier is the main reason for low accuracy of rough synchronization. In contrast, a low-stage voltage multiplier has a short rise time, thereby can produce an output voltage with a steep rising edge, which has a good potential of improving the synchronization precision. In addition, benefitting from the short rise time of the low-stage voltage multiplier, the central node can shorten the length of the RFT to tens of microseconds or less. The short RFT not only reduces the energy consumption of the central node, but also improves the robustness of the synchronization in highly dynamic radio environments.

Although the fine synchronization can provide a high time accuracy, it cannot replace the rough synchronization since the output voltage of the low-stage voltage multiplier is not high enough (e.g., 1.3 V for ATmega256RFR2) to reliably activate the sleeping device. Therefore, we need to use a long RFT to wake up the EHN for rough synchronization, and then use a short RFT for fine synchronization. The details are described as follows:

1) In the rough synchronization phase, the central node transmits an RFT of 150 ms length at local time, t_c. After t_x microseconds, the central node transmits a short RFT of 100 μs for the fine synchronization. Here, t_x is a predetermined value known by the whole network.
2) The EHN activated by the long RFT samples the output of WuR and then performs the rough synchronization to estimate Δt, i.e., the starting point of rough synchronization.
3) Once obtained Δt, the EHN goes to sleep and wakes up after $t_x - \Delta t$ seconds.
4) After the EHN is awakened by the timer timeout interrupt, it samples the fine synchronization signal, which is the output of a low-stage voltage multiplier, and then re-estimates the clock offset from the central node using the mapping table method or NLS algorithm.

If the rise time of the low-stage voltage multiplier is shorter than the minimum sampling interval of ADC, we simplify step (4) as follows:

4)′ After the EHN is awakened by the timeout interrupt, ADC works in free running mode to continuously check the output of the low-stage voltage multiplier. Once the output voltage exceeds a certain threshold, which is denoted by V_{th}, the node records its current local time as t_y. Due to the short rise time of the voltage multiplier, t_y approximates to the arrival time of the short RFT. Hence, the clock

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3154355, IEEE Transactions on Network Science and Engineering

offset between the central node and RFT receivers can be estimated via $t_s = t_y - t_x$.

The pseudocode of rough and fine synchronization processes used to correct the clock skew and clock offset of the EHN can be found in Appendix.

B. Performance Analysis of Fine Sync

In ESTS, neither the central node nor RFT receivers encode/decode or modulate/demodulate any messages, because there is no information exchange during synchronization. Therefore, ESTS does not have the sending or receiving delays like conventional timestamp based synchronization methods. However, it involves several hardware related delays. Next, we use the simplified fine sync method, i.e., step (4)′, as an example to analyze the performance of fine synchronization.

The error of the fine synchronization mainly comes from three parts: (a) the RFT generation delay on the central node, (b) the response delay of WuR circuit, and (c) the ADC's sampling time in step (4)′.

- RFT generation delay (τ_r) is the time it takes to generate a steep RFT signal. This delay is determined by the rise time of the RFT from zero to the specific threshold and the sensitivity of voltage multiplier of RFT receiver.
- Response delay (τ_d) is the time it takes for the voltage multiplier to generate a signal that can be reliably detected by the EHN[3]. The response delay is affected by the strength of the received RFT, the sensitivity and time constant of the WuR, the noise level at the ADC input, and the ADC resolution.
- Sampling time of ADC (τ_c) is the minimum time it takes for ADC to correctly sample the output of WuR. The sampling time is affected by the ADC's sampling rate, the output impedance of the WuR, and the ADC resolution.

Let t_r be the rise time in μs, which can be calculated through a well-known equation: $t_r = 0.35/B$, where B is the bandwidth of the central node in MHz. Denote the sensitivity of the voltage multiplier by V_s. Let V_I be the final voltage of the received RFT. Then the RFT generation delay, τ_r, can be estimated as follows:

$$\tau_r \approx \frac{1.25 * t_r V_s}{V_I} \approx \frac{0.44 * V_s}{B V_I}. \tag{13}$$

According to (13), τ_r is less than 500 ns if the bandwidth of the central node is higher than 1 MHz.

Assume that the noise level at the ADC input pin is V_n, and the resolution voltage of ADC is V_{res}. Then the lowest output voltage of the voltage multiplier that can be recognized as fine sync signal will be $V_n + 0.5V_{res}$. Let K be the ADC's output when the fine sync signal is first detected. We have k equal to the minimum integer making $KV_{res} \geq (V_n + 0.5V_{res})$. According to the definition of the response delay, τ_d will be the time that the output of the WuR rises from 0 to KV_{res}, which is inversely proportional to the intensity of the incident RFT. The accurate response delay may not be obtained in a real system, because it is not only determined by V_I and t_r, but also affected by the WuR hardware. According to our measurements, τ_d is usually less than 500 ns for a low-stage voltage multiplier.

In our implementation, the long sampling time of the ADC is the main cause of the fine sync error. To make the system work reliably under low power supply voltage, the oscillator frequency of ATmega256RFR2 is set to 1 MHz. In this case, the maximum sampling rate of ADC is only 25 kHz, resulting in a long sampling time. In addition, the high output impedance of the low-stage voltage multiplier also increase the sampling time. To be specific, the integrated ADC in ATmega256RFR2 is optimized for the signal source with output impedance of approximately 3 kΩ or less [31]. A high-impedance source like the voltage multiplier on P1110B will cause ADC to have a longer acquisition time to charge its holding capacitor. As a result, we need to reduce the sampling rate to ensure the accuracy of signal conversion. Depending on the structure of the ADC, the required acquisition time may vary. Taking the integrated ADC in Atmega256RFR2 as an example, the minimum acquisition time can be estimated through [31]:

$$\tau_c = (Z_O + 2000) * 0.097\,\text{ns}, \tag{14}$$

where Z_O is the output impedance of the signal source. For the low-stage voltage multiplier on P1110B, $Z_O = 16\,\text{k}\Omega$ and at least $1.57\,\mu$s of acquisition time is required for accurate voltage measurements. In the worst case when the output of the voltage multiplier just exceeds the threshold $(V_n + 0.5V_{res})$ after taking a sample, the fine synchronization error, which is denoted by e_f, will be $e_f \approx \tau_r + \tau_d + \tau_c$. Apparently, we always have positive fine sync errors because τ_r, τ_d, and τ_c are always larger than zero.

VII. Platform and Waveform Analysis

In this section, we first introduces the experimentation platform built for ESTS testing. Afterward, we show the output voltage and current waveforms of the EHN as well as the received RFT, and demonstrate how to measure the synchronization errors.

A. Experimentation Platform

Fig. 6 and Fig. 7 show the diagram of the platform and the experiment setup we built for ESTS testing, respectively. At the central node (i.e., RFT sender), a Tektronix AWG510 arbitrary waveform generator is programmed to produce a square wave with a specific pulse width (i.e., length of RFT signal). Its output is connected to the modulation input of an HP 8648C signal generator, which produces a 912 MHz of continuous wave (CW) as the carrier signal. After signal modulation, HP 8648C outputs a single-frequency pulse wave as the RFT, which is then amplified by a TekBox TBMDA3

[3] The response delay in rough sync is much longer than that in fine sync, as the output voltage of voltage multiplier needs to reach 1.3 V for ATmega256RFR2 to wake-up the sleeping node. In fine sync, by contrast, it only needs to reach a low voltage that can be recognized at ADC.

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

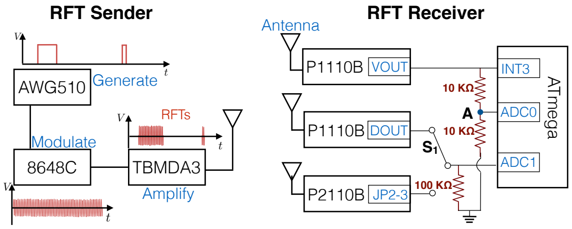


Fig. 6. Diagram of the platform for ESTS.

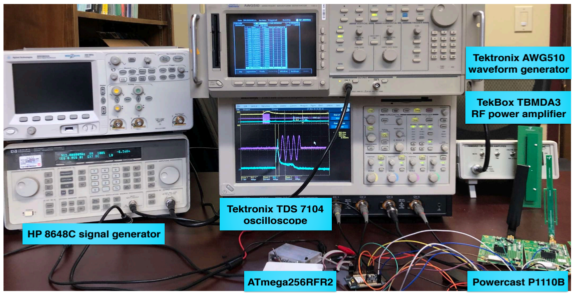

Tektronix AWG510 waveform generator
TekBox TBMDA3 RF power amplifier
Tektronix TDS 7104 oscilloscope
HP 8648C signal generator
ATmega256RFR2
Powercast P1110B

Fig. 7. Experiment setup for ESTS.

RF power amplifier and emitted by a 6 dBi directional (120-degree) patch antenna.

The RFT receiver consists of one ATmega256RFR2 SoC as the controller and two Powercast P1110B boards, serving as either a fully-passive WuR for rough sync or a low-stage voltage multiplier for fine sync. Specifically, P1110B has two outputs: VOUT and DOUT, where DOUT is the output of an RF-DC converter that can be considered as a low-stage voltage multiplier; VOUT is the output of a booster converter, which amplifies the voltage of DOUT, so it can serve as the a high-stage voltage multiplier. For comparison purpose, we also use the JP2-3 pin of the Powercast P2110B as an output of an intermediate-stage voltage multiplier. Compared with the DOUT of P1110B, the output voltage at the JP2-3 pin of P2110B is higher but the rise time is relatively long (several milliseconds). In Fig. 6, the switch S_1 is used to select between P2110B and P1110B for fine synchronization. The Powercast boards are equipped with 1 dBi omnidirectional dipole antennas to receive the RFT signal.

In our experiment setup, the output voltage must be above 1.3 V in order to reliably activate a sleeping ATmega256RFR2 SoC. This wake-up voltage is very close to the internal reference voltage of ADC in ATmega256RFR2. Therefore, when a strong RFT is received, the maximum sampling value of the wake-up signal is likely to be clamped at 1.6 V, resulting in a large synchronization error. To avoid this problem, we connect VOUT of P1110B to two 10 kΩ dividing resistors, and then let ADC sample the voltage at the point A, which is only half of VOUT, as shown in Fig.6. In this way, ADC will not be clamped until VOUT is above 3.2 V. We can adjust the ratio of the two dividing resistors to accommodate RFTs with different strengths.

In ESTS, the outputs of P1110B and P2110B are tied to the ground through one or more pull-down resistors. The resistance of these resistors needs to be appropriately chosen. If the resistance is too small, the circuit cannot generate enough voltage to wake up the node or assist the EHN in time synchronization. If the resistance is too large, the rise time of the wake-up signal will be long and the duration of the RFT needs to be extended, resulting in high energy consumption at the central node and low synchronization accuracy at the EHN. In our experiment, the resistance of the pull-down resistor is 20 kΩ (two 10 kΩ dividing resistors) for VOUT of P1110B and 100 kΩ for DOUT of P1110B and JP2-3 of P2110B.

B. Voltage and Current Waves Analysis

To measure the synchronization error, we use the Tektronix TDS 7104 oscilloscope to monitor the VOUT and DOUT of P1110B, the voltage of the received RFT, and the power supply current of ATmega256RFR2 through a 10 Ω series resistance. Moreover, one channel of the oscilloscope is directly connected to a reference antenna, which is placed right next to the EHN's antenna to estimate the strength of the received RFT.

In Fig. 8, we set the time interval between the long RFT (for rough sync) and the short RFT (for fine sync) as $t_x = 400$ ms, and then show the current and voltage waveforms that running the mapping table based rough sync method and the step $(4)'$ based fine sync approach. The voltage of DOUT is multiplied by 20 to clearly show the waveform in the figure.

We let the central node transmit a 150 ms of long RFT (not shown in Fig. 8) at time $t = 0$. As shown in Fig. 8, when VOUT, i.e., the wake-up signal, reaches 1.3 V at $t = 37$ ms, the RFT receiver is awakened to immediately sample VOUT. Thereafter, the EHN sets a timer of 20 ms and then goes to sleep. At around $t = 57$ ms, the timeout interrupt activates the sleeping node to sample VOUT again. After getting two samples from VOUT, the node runs the mapping table method to estimate the response delay in the rough synchronization. Every time when the ADC is turned on, it consumes a significant amount of energy, hence we can observe the current pulses at $t = 37$ ms and 57 ms when the first and second sampling take place. The RFT receiver returns to sleep mode after rough synchronization.

At $t = 400$ ms, the central node transmits a 100 μs of short RFT for fine synchronization. Based on the estimated response delay and the predetermined interval between rough and fine synchronizations, $t_x = 400$ ms, the RFT receiver can calculate the time to wake up in order to receive the upcoming short RFT. As shown in Fig. 8 (b), the node estimates that the short RFT will arrive at $t = 394.7$ ms, thereby turning on ADC to sample DOUT, which creates the third current pulse. However, due to the error of the rough synchronization, which is denoted by e_r, it waked up $e_r = 400 - 394.7 = 5.3$ ms earlier than needed. In order avoid missing the short RFT, the node monitors DOUT until the voltage exceeds the threshold, V_{th}.

In order to measure the error of fine synchronization, we let the central node transmits a second short RFT at $t = 409$ ms,

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3154355
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3154355, IEEE Transactions on Network Science and Engineering

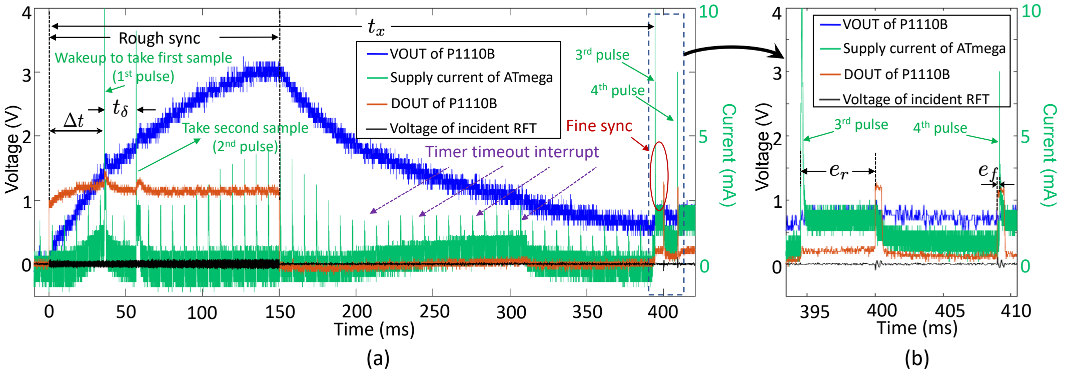


Fig. 8. Voltage and current waveforms on RFT receiver. The voltage waveform of DOUT is multiplied by 20 to make it readable. (a) Panorama. (b) Details.

which is 9 ms behind the first short RFT. When the EHN is activated by the timer, it immediately turns on ADC, thereby creating the fourth current pulse. As marked in Fig. 8 (b), the fine synchronization error, e_f, can be obtained by measuring the time difference between the rising edge of DOUT and the fourth current pulse, which is about 60 μs in Fig. 8. Note that the second short RFT is not required to perform ESTS; it is merely used for performance evaluation purposes.

VIII. Performance Evaluation

In this section, we evaluate the performance of ESTS through experiments. We first test the synchronization accuracy under different settings. Then, the clock skew correction and the energy consumption of ESTS are analyzed.

A. Performance of Rough Synchronization

We run the NLS-based rough synchronization method and show how the synchronization error, e_r, varies with the intensity of the incident RFT signal, I_r, in Fig. 9. As can be observed from the figure, when I_r is large, the rough synchronization can achieve a relatively good accuracy. Taking Fig. 9 (a) as an example, when $N = 7$ and $I_r = -7.6$ dBm, the average of e_r is as high as -30.1 ms, meanwhile e_r has significant variation, ranging from -22.8 ms to -35.4 ms. When I_r increases to -1 dBm, the variance of e_r remarkably decreases, while the average of e_r reduces to 4.7 ms.

The above results are consistent to the observations in Fig. 3. In Fig. 3, we notice that for the NLS-based synchronization method, there is a smaller estimation error of the clock offset with higher incident power of RFT. For a weak RFT signal, the rise time of VOUT will be long while the open-circuit voltage of P1110B will be low. As a consequence, all samples packed in the end of the charging process where the voltages have little changes. This will cause a large error in the NLS fitting. By contrast, when I_r is large, the open-circuit voltage of P110B is high and the rise time of VOUT is short. In this case, the samples scattered over a relatively wide range of the charging curve (i.e., voltages of seven samples range from 1.872 V to 3.072 V in Fig. 3), which helps reduce the fitting error thereby improving the rough synchronization accuracy.

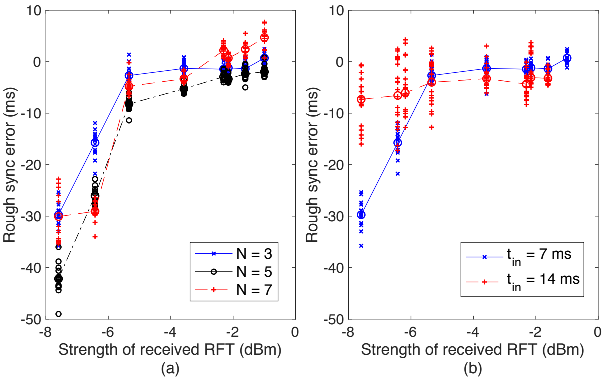


Fig. 9. The error of the NLS-based rough synchronization. Each marker represents the error in a single experiment. Lines are the average of errors measured with the same I_r. (a): Number of iterations is 5, $t_{in} = 7$ ms. (b): Number of iterations is 5, number of samples $N = 3$.

A non-uniform sampling scheme can improve the performance of the NLS-based rough synchronization method. Specifically, after being awakened by the long RFT, the ADC can sample VOUT more frequently at the beginning and then gradually increases the sampling interval at the end. Given a number of samples, this allows a node to get more details of the fast-rising part of the charging process, which is useful to reduce the rough synchronization error.

In Fig. 9 (a), we also evaluate how the samples size, N, affects the performance of the NLS-based rough synchronization method. The results show that when the sampling intervals are the same, increasing the sampling size can hardly reduce the synchronization error. However, as shown in Fig. 9 (b), increasing the sampling interval, t_{in} can greatly improve the synchronization accuracy at low I_r. For instance, when $I_r = -7.6$ dBm and $t_{in} = 7$ ms, the synchronization error is as high as -29.7 ms on average. When increasing t_{in} to 14 ms, the average of e_r can be reduced to -7.6 ms, which is only a quarter of the previous situation.

In Fig. 10, we evaluate the performance of the mapping table based rough synchronization method. As shown in the figure, its performance is comparable with the NLS-based method when the received RFT is strong. To be specific,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3154355, IEEE Transactions on Network Science and Engineering

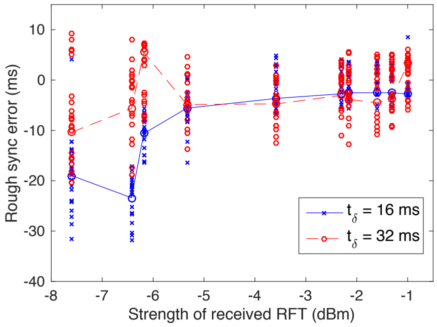
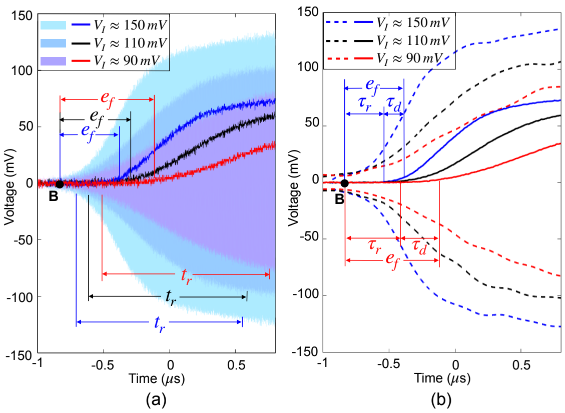


Fig. 10. The error of the mapping table based rough synchronization with respect to the strength of the incident RFT.

when $t_\delta = 32\,\text{ms}$ and I_r is higher than $-2.3\,\text{dBm}$, the synchronization error is less than $-3.1\,\text{ms}$. However, if I_r is low, the synchronization accuracy of the mapping table based method will be better than the NLS-based one. In addition, increasing the time interval between the two samples, i.e., t_δ, can reduce the synchronization error, especially in the case of low I_r. For example, when $I_r = -7.6\,\text{dBm}$ and $t_\delta = 16\,\text{ms}$, e_r is $-18.6\,\text{ms}$; when t_δ is doubled, e_r is reduced to $-10.4\,\text{ms}$.

B. Performance of Fine Synchronization

In Fig. 11, we investigate the possible reasons, such as the RFT generation delay (τ_r) and the response delay (τ_d), that lead to the fine synchronization error. From Fig. 11 (a), it can be observed that when the final voltage of the incident RFT, V_I, varies from $90\,\text{mV}$ to $150\,\text{mV}$, the rise time, t_r, does not change since the bandwidth of the central node is a constant. In addition, the fine synchronization error, e_f, is inversely proportional to V_I. This is because both τ_r and τ_d decrease when the intensity of received RFT grows, as has been analyzed in Section VI-B. By comparing the scenario with $V_I \approx 150\,\text{mV}$ and the case with $V_I \approx 90\,\text{mV}$ in Fig. 11 (b), we observe that τ_r increases from $0.32\,\mu\text{s}$ to $0.43\,\mu\text{s}$ and τ_d grows from $0.17\,\mu\text{s}$ to $0.31\,\mu\text{s}$.



(a)

(b)

Fig. 11. Voltage waves of short RFTs and DOUT. (a) Original measurements. (b) Envelopes of RFTs and smoothed DOUT.

Next, we evaluate the impact of the ADC sampling time, τ_c, on the fine synchronization accuracy. As introduced in Section VI-B, τ_c is related to the output impedance (Z_O) of the voltage multiplier. Therefore, we first measure Z_O of DOUT with respect to I_R, where Z_O is calculated via the open-circuit voltage divided by the shortcut current. From Fig. 12, we know that Z_O is inversely proportional to I_R. If I_R is lower than $-26\,\text{dBm}$, Z_O can be higher than $71\,\text{k}\Omega$. When I_R reaches $-7\,\text{dBm}$, Z_O is reduced to $7\,\text{k}\Omega$.

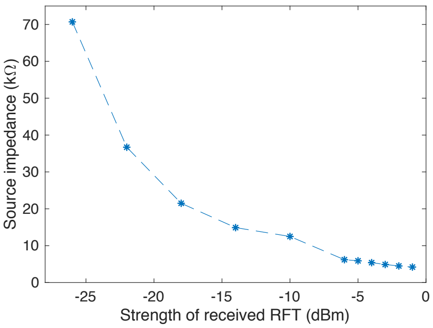


Fig. 12. Variation of Z_O with the intensity of incident RFT.

The integrated ADC in ATmega256RFR2 is optimized for the signal source with an output impedance of approximately $3\,\text{k}\Omega$ or less [31]. However, from Fig. 12 we know that the output impedance on DOUT pin of P1110B is higher than $3\,\text{k}\Omega$, especially when I_R is small. There are two consequences: (a) it introduces strong noises at the ADC input during data conversion, and (b) ADC needs a long acquisition time to charge the holding capacitor for accurate sampling. Both problems will contribute to a longer ADC sampling time.

As demonstrated in Fig. 8 (b), the EHN is in the sleep mode between 393.4 ms and 394.7 ms. During this period, the noise on the DOUT pin (orange line in the right figure) is about 2.91 mV. From 394.7 ms to 400.2 ms, the node starts measuring DOUT for fine synchronization, thus injecting a strong ADC noise. According to our measurements, the average noise on the P1110B DOUT pin increases to 9.9 mV during voltage sampling, which is 3.4 times higher than that in the sleep mode. As analyzed in Section VI-B, strong noises at the ADC input increases the response delay. This is because for a given I_r, it will take longer time to detect the arrival of short RFT waiting for the voltage to reach $V_n + 0.5V_{res}$. In addition to a longer response delay, the high output impedance also increases the time constant of ADC circuit. As a consequence, the ADC will spend more time charging the holding capacitor for each sampling, resulting in a long acquisition time. According to (14), when $Z_O = 7\,\text{k}\Omega$, the shortest acquisition time of ADC will be greater than $0.87\,\mu\text{s}$.

In Fig. 13, we show the performance of fine synchronization, where the CPU frequency, the sampling rate of ADC and the threshold for RFT detection (V_{th}) are set to 1 MHz, 25 kHz and 30 mV, respectively. For comparison purpose, we also test the fine synchronization method with the Powercast P2110B working as the low-stage voltage multiplier. As can be observed from the figure, the average of the fine synchronization error with P1110B is less than -70 μs. Moreover, the

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

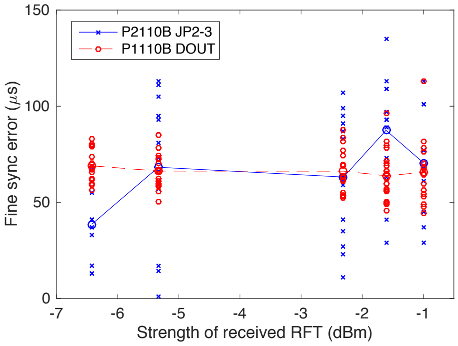


Fig. 13. The error of fine synchronization with respect to the strength of the received RFT.

average performance of the fine synchronization with P2110B is comparable to that with P1110B. However, the former has a much larger variance than P1110B. Therefore, we use P1110B as the default voltage multiplier in ESTS.

From Fig. 13, we realize that the performance of fine synchronization is not sensitive to the strength of the received RFT. This is because I_r only affects the RFT generation delay and the response delay. For a low-stage voltage multiplier, both of them are less than 0.5 μs even with a weak RFT, as illustrated in Fig. 11. In our implementation, the error of fine synchronization is mainly caused by the ADC sampling time. The long sampling time caused by the low sampling rate of ADC does not change with I_r. Therefore, ESTS is able to maintain a stable synchronization accuracy regardless of the intensity of the incident RFT.

Increase the ADC sampling rate can significantly improve the fine synchronization accuracy. With 25 kHz sampling rate, we have 40 μs sampling time contributing to up to 95% of fine sync error shown in Fig. 11. The ADC sampling rate is proportional to the CPU frequency. If we increase the CPU frequency to 8 MHz, the ADC sampling time will also be eight times less and we can dramatically reduce the synchronization error. However, the energy consumption will double. Therefore, we choose 1 MHz CPU frequency for low energy consumption and acceptable synchronization accuracy.

To further improve the performance of fine synchronization, we can increase the bandwidth of the central node to minimize the RFT generation delay. At the same time, we can reduce the output impedance of the low-stage voltage multiplier to further shorten the ADC's sampling time. This can be achieved by inserting a buffer op-amp between the output of the voltage multiplier and the ADC input. If the sampling time and the RFT generation delay can be minimized, the fine synchronization error will be mainly determined by the response delay of the voltage multiplier, which can be less than 0.5 μs, as shown in Fig. 11. However, using the buffer op-amp and increasing the clock speed of ADC consumes extra energy. Therefore, there is a trade-off between the synchronization accuracy and energy consumption.

C. Skew Correction

To correct the clock skew of the EHN, the central node broadcasts a short RFT every T_α seconds, where T_α is a predetermined value known to all nodes in the network. Let T_i be the local time when the central node sends the i^{th} RFT. Once the EHN receives the i^{th} RFT , it performs the fine synchronization to estimate the arrival time of the RFT, which is represented by EHN's local time as t_y^i.

Denote the clock skew and the offset of the EHN relative to the central node by C_s and C_o, respectively. It can be obtained that

$$T_i = C_s \times t_y^i + C_o \tag{15}$$

and

$$T_{i+1} = C_s \times t_y^{i+1} + C_o \tag{16}$$

Subtracting (15) from (16), we have that

$$C_s = \frac{T_{i+1} - T_i}{t_y^{i+1} - t_y^i} = \frac{T_\alpha}{t_y^{i+1} - t_y^i}. \tag{17}$$

Based on (17), the EHN can calculate its clock skew relative to the central node. When the EHN receives a new RFT, it will update C_s based on the results calculated by (17). Moreover, to improve the estimation accuracy, we use 6 RFTs as a set for the clock skew calculation. The highest and lowest values are discarded to eliminate the glitches and the average of the middle three values is then used as the estimated clock skew.

In order to track the local time during sleep, the SoC is configured to use a crystal oscillator with 32.768 kHz frequency as the clock of the timer. In Fig. 14 (a), we set T_α to 2 seconds and then show how ΔT_i changes in a 2-hour experiment, where $\Delta T_i = t_y^{i+1} - t_y^i - T_\alpha$ is the time error accumulated in T_α seconds caused by the clock skew between the central node and RFT receivers. During the test, ambient temperature gradually increased from 24°C to 26.8°C.

As illustrated in Fig. 14 (a), in most cases, the node accumulates about -7.7 ms of error in every 2 seconds if there is no clock skew correction. Furthermore, over 90% of cumulative errors distribute between -7.3 ms and -7.9 ms, which means the EHN's clock frequency is relatively stable over time, as shown in Fig. 14 (b).

In Fig. 14 (c), we let the EHN periodically update the clock skew to correct its local time. Compared with Fig. 14 (a), the cumulative time error is significantly reduced with skew correction. As shown in Fig. 14 (c), if the node only corrects the clock skew at the beginning the test, ΔT gradually increases from 0 to -40.71 μs. If the node can estimate and update the clock skew every 10 mins, we can limit most of ΔT between -10 μs and 10 μs. If we further increase the frequency of the clock skew correction to once per minute, then ΔT is between -3 μs and 3 μs in most cases.

D. Energy Consumption

The energy cost of ESTS is very low. As demonstrated in Fig. 8, to complete one mapping table based rough synchronization and one fine synchronization, the node just needs to briefly wake up three times (i.e., two for rough and one for fine

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2022.3154355

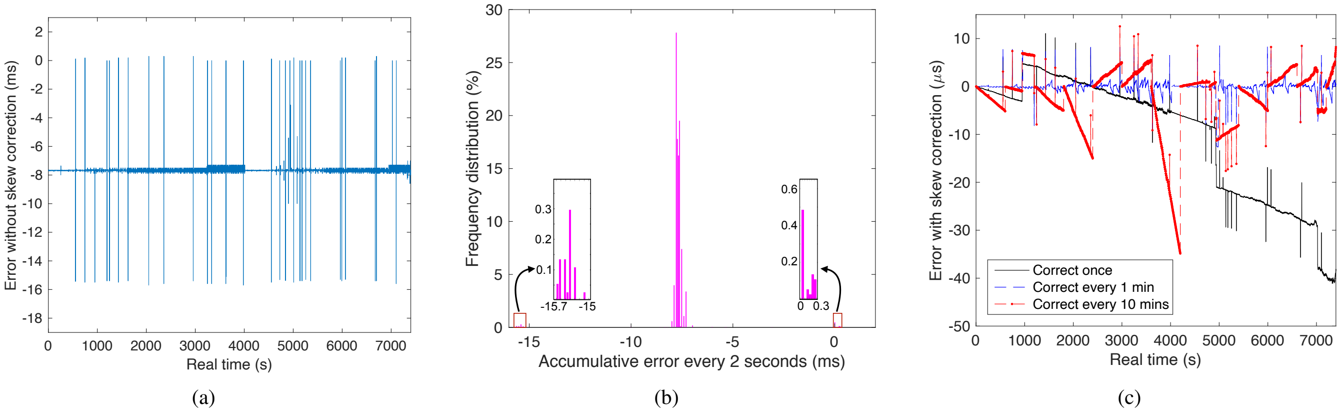


Fig. 14. Clock skew correction. (a) The time error accumulated in 2 seconds without skew correction. (b) The distribution of the time error in (a). (c) The time error accumulated in 2 seconds with skew correction.

sync), which creates three current pulses, and then returns to sleep for the rest of the time. In the sleep mode, since the fully-passive WuR consumes zero energy, the power consumption of the ESTS is only 1.2 μW with 2 V power supply voltage.

In our experiments, the energy consumption in the first and second active periods are only 8.92 μJ and 9.79 μJ, respectively. The energy consumption of the third active period includes two parts: a) turning on ADC, and b) continuously sampling the output of the voltage multiplier to detect the short RFT. Part a) consumes 3.65 μJ of energy, which is a constant that depends only on supply voltage. The power consumption of Part b) is a fixed value, which is 3.49 mW, but the energy cost depends on the rough synchronization accuracy. According to Fig. 9 (b), if the strength of the received RFT is higher than -5.3 dBm, the rough synchronization error of the mapping table based method is less than -5 ms. Therefore, the EHN will continuously sample the output of the voltage multiplier for 5 ms until the real RFT arrives, which consumes 17.5 μJ (3.49 mW× 5 ms) of energy.

It is worth noting that ESTS needs to perform the rough sync only once, and the remaining process, including the clock offset estimation and clock skew correction, can be completed by fine sync. According to measurements, an EHN consumes 3.65μJ of energy during each fine sync process, which is negligible compared with the energy consumption of wireless communications. Next, we analyze the average power consumption of the central node.

Assume that the central node and RFT receivers are equipped with 6 dBi 912 MHz antennas. In the long run, the energy consumption for transmitting the long RFT is negligible since the rough synchronization is performed only once during the entire synchronization process. Assuming we run the clock screw correction every minute, a set of 6 short RFTs will be sent out every minute, with each short RFT 100 μs. If the coverage of the central node is 15 m, the free-space path loss of radio waves is as high as 43.16 dB. Accordingly, the transmission power of each RFT should be at least 4.13 W to ensure that the RFT received by all EHNs is higher than -7 dBm. After a simple calculation, it can be obtained that the central node consumes 413 μJ of energy to transmit each short RFT, that is, 2.48 mJ of energy for a set of six RFTs per minute. In this situation, the average power consumption of the central node is only 41 μW. The long-term power consumption of the central node can be even less if we correct the clock skew less frequently, for example, every 10 minutes.

The low energy consumption of the RFT making it possible to develop mobile center to improve the scalability of ESTS. In the mobile center, a DJI Matrice 600 pro industrial drone [38], which can carry up to 6 kg payload, is equipped with a GPS module and a software-defined radio (SRD) to periodically generate short plus signal at 915 MHz frequency. Afterward, the signal passes through an RF amplifier with 40 dBm gain to generate the RFT having enough strength. The battery life of the DJI Matrice 600 is about 30 mins at 40 mph speed, which means that the coverage of the drone can reach 20 miles for a one-way cruise or 10 miles for a two-way cruise.

E. Performance Comparison

In Table III, we compare the performance of ESTS with several conventional time synchronization methods and the semi-passive WUR-TS [34]. We list the number of synchronization messages that each synchronization protocol needs to send and receive as an indicator of energy consumption. In RBS and CESP, a reference node broadcasts a beacon in each synchronization period. After receiving the beacon, the master node in RBS sends the timestamp to the slave node, whereas the master and slave nodes exchanges coefficients in CESP. Both ESTS and WUR-TS are WuR-based time synchronization methods. All nodes get synchronized after receiving the wake-up beacon without timestamp exchanges.

From Table III, it can be realized that the energy consumption of ESTS is much lower than conventional methods. It is worth noting that when testing the performance of conventional time synchronization methods, nodes never enter the sleep mode. Therefore, the actual energy consumption for RBS, CESP, and TPSN will be much higher than that estimated in Table III. If wake-up circuit is applied to allow nodes to enter sleeping mode, extra delay uncertainty caused by the

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

WuR may further reduce the accuracy of RBS, CESP, and TPSN in real applications.

ESTS is superior to WUR-TS in terms of lower power consumption. Specifically, ESTS is a fully-passive time synchronization method. In the idle mode when no wake-up signal is received, only the microcontroller consumes energy, which is about 1.2 μW. By contrast, WUR-TS is a semi-passive time synchronization method. In addition to the microcontroller, the comparator in the WuR receiver also consumes energy during the idle phase. The total power consumption is about 3.2 μW, three times higher than ESTS. At the cost of higher energy consumption, the synchronization accuracy of WUR-TS is higher than ESTS; however, the former is much more sensitive to the changes in the intensity of the wake-up signal than the latter. According to our experimental results, if the strength of the RFT is low, the synchronization error of WUR-TS increases to 80 μs; ESTS can maintain a stable synchronization precision.

To summarize, all synchronization methods listed in Table III can achieve sufficient synchronization accuracy for wireless networks. Compared with the conventional methods, ESTS and WUR-TS can achieve much higher energy efficiency and become more suitable for ultra-low-power wireless devices. The fully-passive ESTS has lower sensitivity and thus has shorter synchronization range than the semi-passive WUR-TS, but has three times lower energy consumption. Therefore, ESTS is recommended if the synchronization range is not concerned (e.g., with mobile center node) or the energy harvesting rate of EHNs is a very limited. WUR-TS is advocated for longer synchronization range when the energy resources are relatively sufficient on EHNs.

TABLE III
SYNCHRONIZATION PERFORMANCE COMPARISON

	Accu.	Operation and Energy consumption
ESTS	70 μs @≥-7.6 dBm	Reference: Tx 1 wake-up beacon Receiver: Rx 1 wake-up beacon (1.2 μW idle power consumption; 3.65 μJ on one round of fine sync)
WUR-TS [34]	3 μs @≥-33 dBm 80 μs @-40 dBm	Reference: Tx 1 wake-up beacon Receiver: Rx 1 wake-up beacon (3.2 μW idle power consumption; 3.6 μJ on one round of synchronization)
RBS [5]	3.7 μs	Reference: Tx 1 beacon Master node: Rx 1 beacon, Tx 1 packet Slave node: Rx 1 beacon and 1 packet (17.5 mW during sync initialization)
CESP [26]	11 μs	Reference: 1 beacon Mater: Rx 1 beacon and 1 packet, Tx 1 packet Slave: Rx 1 beacon and 1 packet, Tx 1 packet (17.5 mW during sync initialization)
TPSN [5]	22.7 μs	Reference: Tx 1 packet, Rx 1 packet Receiver: Rx 1 packet, Tx 1 packet (17.5 mW during sync initialization)

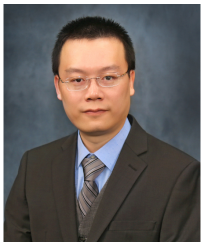
IX. CONCLUSIONS

In this article, we developed a new time synchronization method, called energy stimulated time sync (ESTS), for ultra-low-power energy harvesting wireless networks. ESTS does not rely on timestamp exchanges, but uses short radio frequency tones to synchronize EHNs with a central node.

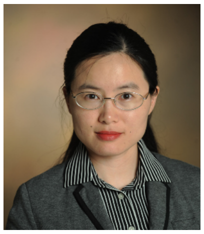
We implemented ESTS on a real platform and evaluated its performance through theoretical analysis and extensive experiments. The results show that compared with timestamp-based synchronization methods, ESTS can significantly reduce the EHN's energy consumption, while providing an average accuracy of 5 ms and 70 μs with rough and fine synchronizations, respectively. Based on experiment results, we discussed how to further improve the performance of ESTS. By increasing the bandwidth of the central node, using a buffer op-amp to reduce the output impedance of the voltage multiplier or increasing the clock rate of ADC, we can expect 0.5 μs or even higher accuracy for ESTS with fine synchronization. We hope this article can provide a new perspective for synchronizing the time of ultra-low-power wireless networks using simple analog signals.

ACKNOWLEGEMENT

This work is supported in part by the US National Science Foundation under Grant No. 2051356, No. 2122167 and No. 2122159.



Dr. Yu Luo received the B.S. degree and the M.S. degree in electrical engineering from the Northwestern Polytechnical University, China, in 2009 and 2012, respectively. In 2015, he received the Ph.D. degree in computer science and engineering from University of Connecticut, Storrs. Dr. Luo is currently an Assistant Professor at Mississippi State University. His major research focus on the sustainable wireless networks for emerging IoT, RF energy harvesting hardware, security in RF energy harvesting wireless networks, and underwater wireless networks. He is a Co-recipient of the Best Paper Award in IFIP Networking 2013 and Chinacom 2016.



Dr. Lina Pu received the B.S. degree in electrical engineering from the Northwestern Polytechnical University, Xi'an, China in 2009 and the Ph.D. degree in Computer Science and Engineering from University of Connecticut, Storrs. Dr. Pu is currently an Assistant Professor at University of Alabama. Her research interests lie in the area of edge computing, RF energy harvesting wireless networks, security in the sustainable IoT, and underwater acoustic/VL networks. She owned IFIP Networking 2013 best paper award.

Zheng Peng received a Bachelor's degree in Control Theory and a Bachelor's degree in Computer Science from Zhejiang University, China in 2002. He obtained a Master's degree in Computer Science from University of Electrical Science and Technology of China in 2005, and later his Ph.D. from University of Connecticut, USA. Zheng joined The City College of New York as an Assistant Professor in 2016. Prior to the current position, Zheng was an Assistant Research Professor at University of Connecticut. His research interests include the design, modeling, development, and performance evaluation of the wireless sensor networks, embedded/distributed systems and their applications in challenging environments.

References

[1] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[2] M. Ghamari, B. Janko, R. S. Sherratt, W. Harwin, R. Piechockic, and C. Soltanpur, "A survey on wireless body area networks for ehealthcare systems in residential environments," *Sensors*, vol. 16, no. 6, p. 831, 2016.

[3] F. Akhtar and M. H. Rehmani, "Energy harvesting for self-sustainable wireless body area networks," *IT Professional*, vol. 19, no. 2, pp. 32–40, 2017.

[4] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, vol. 11, no. 2, pp. 983–994, 2017.

[5] M. A. Sarvghadi and T.-C. Wan, "Message passing based time synchronization in wireless sensor networks: a survey," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, p. 1280904, 2016.

[6] A. R. Swain and R. Hansdah, "A model for the classification and survey of clock synchronization protocols in WSNs," *Ad Hoc Networks*, vol. 27, pp. 219–241, 2015.

[7] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE network*, vol. 18, no. 4, pp. 45–50, 2004.

[8] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, and et al., "Energy harvesting wireless communications: a review of recent advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015.

[9] A. R. M. Siddique, R. Rabari, S. Mahmud, and B. Van Heyst, "Thermal energy harvesting from the human body using flexible thermoelectric generator (FTEG) fabricated by a dispenser printing technique," *Energy*, vol. 115, pp. 1081–1091, 2016.

[10] M. Piñuela, P. D. Mitcheson, and S. Lucyszyn, "Ambient RF energy harvesting in urban and semi-urban environments," *IEEE Transactions on microwave theory and techniques*, vol. 61, no. 7, pp. 2715–2726, 2013.

[11] H. Bello, Z. Xiaoping, R. Nordin, and J. Xin, "Advances and opportunities in passive wake-up radios with wireless energy harvesting for the internet of things applications," *Sensors*, vol. 19, no. 14, p. 3078, 2019.

[12] A. Froytlog, T. Foss, O. Bakker, G. Jevne, A. Haglund, Y. Li, J. Oller, and Y. Li, "Ultra-low power wake-up radio for 5G IoT," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 111–117, 2019.

[13] J. Randall, *Designing indoor solar products: photovoltaic technologies for AES*. John Wiley & Sons, 2006.

[14] G. Papadakis, P. Tsamis, and S. Kyritsis, "An experimental investigation of the effect of shading with plants for solar control of buildings," *Energy and Buildings*, vol. 33, no. 8, pp. 831–836, 2001.

[15] S. Niu, X. Wang, F. Yi, Y. S. Zhou, and Z. L. Wang, "A universal self-charging system driven by random biomechanical energy for sustainable operation of mobile electronics," *Nature communications*, vol. 6, p. 8975, 2015.

[16] R. Jurdak, A. G. Ruzzelli, and G. M. O'Hare, "Multi-hop RFID wake-up radio: design, evaluation and energy tradeoffs," in *2008 Proceedings of 17th International Conference on Computer Communications and Networks*. IEEE, 2008, pp. 1–8.

[17] H. Ba, J. Parvin, L. Soto, I. Demirkol, and W. Heinzelman, "Passive RFID-based wake-up radios for wireless sensor networks," in *Wirelessly Powered Sensor Networks and Computational RFID*. Springer, 2013, pp. 113–129.

[18] J. Yi, W.-H. Ki, and C.-Y. Tsui, "Analysis and design strategy of UHF micro-power CMOS rectifiers for micro-sensor and RFID applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 153–166, 2007.

[19] T. Umeda, H. Yoshida, S. Sekine, Y. Fujita, T. Suzuki, and S. Otaka, "A 950-MHz rectifier circuit for sensor network tags with 10-m distance," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 35–41, 2005.

[20] S. Scorcioni, L. Larcher, A. Bertacchini, L. Vincetti, and M. Maini, "An integrated RF energy harvester for UHF wireless powering applications," in *2013 IEEE Wireless Power Transfer (WPT)*. IEEE, 2013, pp. 92–95.

[21] B. Van der Doorn, W. Kavelaars, and K. Langendoen, "A prototype low-cost wakeup radio for the 868 MHz band," *International Journal of Sensor Networks*, vol. 5, no. 1, pp. 22–32, 2009.

[22] C. Hambeck, S. Mahlknecht, and T. Herndl, "A 2.4 μW Wake-up Receiver for wireless sensor nodes with- 71dBm sensitivity," in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE, 2011, pp. 534–537.

[23] S. J. Marinkovic and E. M. Popovici, "Nano-power wireless wake-up receiver with serial peripheral interface," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1641–1647, 2011.

[24] J. Oller, I. Demirkol, J. Casademont, and J. Paradells, "Design, development, and performance evaluation of a low-cost, low-power wake-up radio system for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, no. 1, pp. 1–24, 2013.

[25] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[26] F. Gong and M. L. Sichitiu, "CESP: A low-power high-accuracy time synchronization protocol," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2387–2396, 2015.

[27] X. Huan, K. S. Kim, S. Lee, E. G. Lim, and A. Marshall, "A beaconless asymmetric energy-efficient time synchronization scheme for resource-constrained multi-hop wireless sensor networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1716–1730, 2019.

[28] K. S. Yildirim and A. Kantarci, "Time synchronization based on slow-flooding in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 244–253, 2013.

[29] K. S. Yıldırım, R. Carli, and L. Schenato, "Adaptive proportional–integral clock synchronization in wireless sensor networks," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 2, pp. 610–623, 2017.

[30] C. Lenzen, P. Sommer, and R. Wattenhofer, "PulseSync: an efficient and scalable clock synchronization protocol," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 717–727, 2014.

[31] Microchip Technology, *Atmel ATmega256RFR2 datasheet*, Microchip Technology, Chandler, AZ, U.S.A., September 2014.

[32] AVX Company, *AVX BestCap Ultra-low ESR high power pulse supercapacitors*, AVX Company, Greenville, SC, U.S.A.

[33] Y. Luo and L. Pu, "ESTS: Energy Stimulated Time Synchronization for Energy Harvesting Wireless Networks," in *IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2020, pp. 1–6.

[34] ——, "WUR-TS: Semi-Passive Wake-Up Radio Receiver Based Time Synchronization Method for Energy Harvesting Wireless Networks," *IEEE Transactions on Mobile Computing*, 2021.

[35] R. Vyas, B. Cook, Y. Kawahara, and M. Tentzeris, "E-WEHP: a battery-less embedded sensor-platform wirelessly powered from ambient digital-TV signals," *IEEE Transactions on microwave theory and techniques*, vol. 61, no. 6, pp. 2491–2505, 2013.

[36] Y. Luo, L. Pu, G. Wang, and Y. Zhao, "RF energy harvesting wireless communications: RF environment, device hardware and practical issues," *Sensors*, vol. 19, no. 13, p. 3010, 2019.

[37] Y. Luo, L. Pu, Y. Zhao, W. Wang, and Q. Yang, "A nonlinear recursive model based optimal transmission scheduling in RF energy harvesting wireless communications," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3449–3462, 2020.

[38] DJI Company, "MATRICE 600PRO," https://www.dji.com/matrice600-pro, 2021, [Accessed: June, 2021].