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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

WUR-TS: Semi-Passive Wake-Up Radio Receiver Based Time Synchronization Method for Energy Harvesting Wireless Networks

Yu Luo *Member, IEEE*, Lina Pu *Member, IEEE*

Abstract— In this paper, a semi-passive wake-up radio receiver based time synchronization (WUR-TS) method is developed for energy harvesting wireless networks. In WUR-TS, energy harvesting nodes (EHNs) do not exchange any timing information, but only receive the wake-up signal broadcast periodically by the central node to synchronize time throughout the network. Based on the experimental data, a model is developed to accurately estimate the arrival time of each wake-up signal. By using this model, the EHN can calculate its clock drift in complex radio environments. We have implemented WUR-TS with a minimum number of commercial components. According to the experimental results, WUR-TS can achieve a synchronization accuracy of 3 μs when the power supply voltage is 2.8 V and the received wake-up signal strength is higher than -33 dBm. WUR-TS is an ultra-low-power synchronization method. If no wake-up signal is detected, the power consumption of each EHN is 3.2 μW. If the wake-up signal is detected, the EHN consumes only 3.6 μJ of energy to complete time synchronization.

Index Terms—Time synchronization, semi-passive wake-up radio receiver, ultra-low-power, energy harvesting.

1 Introduction

Energy harvesting wireless networks have wide applicability in precision agriculture, smart city, body area network (BAN), and internet of things (IoT) [1], [2]. By collecting sustainable energy from the surrounding environment, wireless networks can operate semi-perpetually without having to replace batteries. Due to the low power density in the environment, the sustainable energy that can be collected by the energy harvesting node (EHN) within a given time is usually small. Therefore, EHNs need to stay asleep most of the time and are only activated when performing certain tasks (such as sensing and communication). To achieve this goal, the communicating parties must synchronize their time to send and receive data at the same time; otherwise, a sleeping receiver is likely to miss the messages from the sender.

Conventional time synchronization methods developed for wireless networks usually rely on a timestamp exchange mechanism between nodes to estimate relative clock skew and clock offset [3]. In these methods, wireless nodes need to keep idle listening until they receive a beacon or an initial timing information to complete the first synchronization. During the initialization phase of time synchronization, the node will consume a large amount of energy for idle listening. However, due to its small energy storage capacity (tens of millijoules or less), the EHN is very sensitive to burst energy consumption [4]. Therefore, for energy harvesting wireless networks, how to improve the energy efficiency of the initial synchronization is a challenging problem.

Wake-up radio receiver (WUR) is a promising solution to the above problem. In this solution, WUR is attached to each EHN, and its role is to generate an interrupt with sufficient voltage to activate the sleeping node after receiving a wake-up signal [5]. By using WUR, the sender can transmit a specific signal to wake up the receiver for data reception, thereby preventing EHNs from idle listening. However, in existing wireless networks, WUR does not directly participate in the time synchronization process, but is only used as an auxiliary circuit to activate nodes for data reception [6]. Therefore, EHNs still need to exchange timestamps with each other to correct their clock skew and clock offset, which greatly wastes the potential of WUR. In order to make full use of WUR, we develop a semi-passive wake-up radio receiver based time synchronization (WUR-TS) method for energy harvesting wireless networks.

In WUR-TS, EHNs will not exchange any timing information, but only use the wake-up signal from the central node to synchronize time with the network. To achieve this, the central node is scheduled to periodically broadcast a wake-up signal. Afterward, the EHN calculates the arrival time of the wake-up signal to estimate the clock skew. Since there is no message exchange in the network, WUR-TS cannot help EHNs correct the clock offset. Nevertheless, the clock skew information is sufficient for EHNs to efficiently arrange the communication scheme and sleep pattern. According to the features of WUR-TS, two types of medium access control (MAC) protocols can make the best use of our time synchronization method, namely, slotted MAC protocols, which have been investigated in [7], and sleep scheduling based MAC protocols, such as [8] and [9].

There are two challenging problems when implementing WUR-TS. First of all, due to the high propagation loss of radio waves, the strength of the wake-up signal reaching an EHN is usually low. In a complex radio environment with radio interference, how to activate an EHN reliably through a weak wake-up signal is an important issue. Secondly, in WUR, there is a delay between receiving the wake-up signal and generating an effective interrupt to activate an EHN. This delay is called the interrupt delay, which is not a constant, but is affected by many factors, such as the strength of the wake-up signal and the power supply voltage of EHN. As a result, the interrupt delay varies with time, node activity, and channel response. How to accurately estimate the interrupt delay is the key to calculating the arrival time of the wake-up signal, which is a challenging problem.

After considering the energy constraints of EHN and the sensitivity of WUR, we design a semi-passive WUR to solve the first challenging problem. In our design, the power consumption of WUR is very low when continuously

- *Y. Luo is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, 39759. Email: yu.luo@ece.msstate.edu*
- *L. Pu is with the Department of Computer Science, University of Alabama, Tuscaloosa, AL 35487 Email: lina.pu@ua.edu*

listening to the wireless channel. By using a voltage booster and a low-offset comparator, WUR can reliably detect the wake-up signal as low as $-45\,\text{dBm}$. To avoid interference in complex environments, we add two resistor-capacitor (RC) networks in WUR. Those RC networks only allow WUR to generate effective interrupts when the received signal exceeds a certain length. In this way, short interferences will be filtered out to avoid incorrect synchronization. To solve the second problem, we carefully study the electrical characteristics of each component in WUR, and then model the interrupt delay of WUR through theoretical analysis and experimental measurement. By using our delay model, the EHN can accurately estimate the arrival time of each wake-up signal in real-time.

We implemented WUR-TS using a small number of off-the-shelf components: Texas Instruments LPV7215 comparator, Powercast P1110B voltage booster, and Microchip ATmega256RFR2 system-on-chip (SoC). According to experimental results, when the power supply voltage is $2.8\,\text{V}$ and the received wake-up signal strength is higher than $-33\,\text{dBm}$, the synchronization accuracy of WUR-TS is higher than $3\,\mu\text{s}$. When running WUR-TS, the power consumption of each EHN is only $3.2\,\mu\text{W}$ while idle listening. If a wake-up signal is detected, the EHN consumes only $3.6\,\mu\text{J}$ energy to complete time synchronization.

In summary, the main contributions of our work include the following three aspects:

a) We designed a semi-passive WUR, which has only an ultra-low-power comparator as an active component. The receiver is built with a minimal number of low-cost components and can work reliably in a complex radio environment.
b) We developed a model to estimate the interrupt delay of WUR. The proposed model can be used to accurately calculate the arrival time of the wake-up signal for time synchronization.
c) Based on the delay model of WUR, a new time synchronization method, called WUR-TS, is implemented. This method can synchronize the time of the EHN with the central node through a simple wake-up signal without any message exchange.

The rest of the paper is organized as follows: Section 2 introduces related work. We give an example of applying WUR-TS in precision agriculture in Section 3, and then present our WUR design in Section 4. The output characteristic of voltage booster and comparator is modeled in Section 5. In Section 6 and Section 7, we provide a model to calculate the interrupt delay of WUR. In Section 8, the analog-to-digital converter (ADC) noise problem caused by high output impedance of the voltage booster will be discussed. The performance of WUR-TS is evaluated in Section 9. Eventually, we conclude our work in Section 10.

2 Related Work

There are a considerable number of time synchronization protocols in wireless networks literature. Several protocols, such as reference broadcast synchronization (RBS) [10], timing-sync protocol for sensor networks (TPSN) [11], and flooding time synchronization protocol (FTSP) [12], form the foundation of many other methods.

RBS is a receiver-receiver synchronization [10] protocol. During time synchronization, the third party broadcasts a simple beacon to all receivers. The receivers will then record when the beacon was received according to their local time. Afterward, receivers exchange the recorded timing information with neighbors to calculate the clock offset. The main advantage of RBS is that it eliminates the uncertainty of the sender by removing the sender from the critical path. As a result, the only uncertainty of RBS is the propagation delay of RF waves and receive time of synchroniztion messages. TPSN is a traditional sender-receiver synchronization protocol [11]. It uses a tree to organize the network topology. In the level discovery phase, each node is assigned a level to form a hierarchical topology of the network. In the synchronization phase, all nodes at the same level synchronize with upper-level nodes by exchanging two-way timestamps between a pair of nodes. Compared to RBS, TPSN cannot eliminate the uncertainty of the sender. However, TPSN does not rely on the beacon from the third party, therefore can work in a multi-hop network. However, TPSN is sensitive to the changes of network topology since it needs to reinitiate the level discovery phase if the root node or the topology is changed, which generates additional overhead traffic. FTSP is another sender-receiver synchronization protocol designed for large multi-hop networks [12]. In FTSP, the root node sends a timestamp at the MAC layer as the global time to all receivers. The receiver records its local time after receiving the timestamp from the root node. Afterward, each receiver can estimate the clock offset based on the sending time and receiving time of the timestamp. The main advantage of FTSP is that it is robust in topology changes in a large-scale network. This is because FTSP uses the flooding of synchronization messages to combat link and node failure, thereby providing the ability for dynamic topology changes.

To improve the performance of the above classic protocols, many new synchronization solutions have been proposed in recent years. In [13], a rapid-flooding time synchronization protocol, called PulseSync, is developed to provide high-precision time synchronization service for large-scale wireless networks. PulseSync relies on the choice of a message delay calibration and the distribution of propagation delays between wireless nodes. Each node uses linear regression to estimate the clock of the root node. The advantage of PulseSync is that it allows the reference time to be swiftly distribute in a multi-hop network, thereby avoiding errors to accumulate exponentially with increasing delays. However, PulseSync needs nodes to overhear the channel for other synchronization messages, especially in the initial phase, so them can quickly learn about the presence of a reference node. This makes PulseSync energy inefficient when applying it in an energy harvesting network, which requires nodes to stay in the sleep mode most of the time to save the energy. In [14], a slow-flooding-based time synchronization, called FCSA, is designed. FCSA aims at reducing the undesired effect of waiting times on the synchronization accuracy when nodes propagate their timing information about the reference node hop-by-hop. This is achieved by employing a clock speed agreement algorithm to keep track of the neighboring nodes, thereby forcing all the nodes to run at the same speed. The disadvantage of FCSA is that it suffers from the crucial problem of deciding which neighbors to keep track of and which ones to discard since it is infeasible for nodes to track all their neighbors due to their memory constraints. Moreover, it takes a long time to provide tight synchronization with FCSA. In addition, it is very hard for an EHN to continuously track its neighbors due to the limited energy supply in an energy harvesting network.

In [15], an energy-efficient coefficient exchange synchronization protocol (CESP) is proposed. CESP is a receiver-receiver synchronization protocol that uses synchronization coefficients to reduce communication overhead, thereby improving the energy efficiency of RBS. The same as RBS, the performance of CESP depends on the accuracy of the time when nodes receive a beacon from the reference node. In an energy harvesting network where EHNs stay in the sleep mode most of the time and need to be activated by a WUR, the reception time of the beacon is affected by the output characteristics of the WUR. In [16], a robust time synchronization scheme, called R-Sync, is developed for the industrial internet of things (IIOT). This scheme focuses on providing some isolated nodes synchronization service. In R-Sync, a pulling timer is used to pull the isolated node into the synchronized network through a receiver-receiver protocol or a sender-receiver protocol whose initial value is set according to the level of spanning tree. Then, another timer is set up to select the backbone node, and its initial value is related to the distance to the parent node. Although R-sync requires relatively fewer numbers of messages compared to TPSN, it does not have any mechanism to avoid collisions of sync messages. This reduces its energy efficiency in a high-density wireless network.

3 Application and Objective

In this section, we provide a typical example of using WUR-TS in precision agriculture. Afterward, we briefly introduce the research objective of our work.

3.1 Application Scenario

Precision agriculture can get significant benefit from energy harvesting wireless networks. By deploying wireless energy harvesting nodes (EHNs) with sensors in farmland, people can monitor the state of the soil (e.g., the moisture, temperature, and pH value) in a large area for a long time. However, due to the low energy harvesting rate [17]–[19], the EHNs need to stay at the sleep mode most of the time and only wake up when environment sensing and communications are required. How to efficiently collect the data measured by EHNs becomes a big challenge.

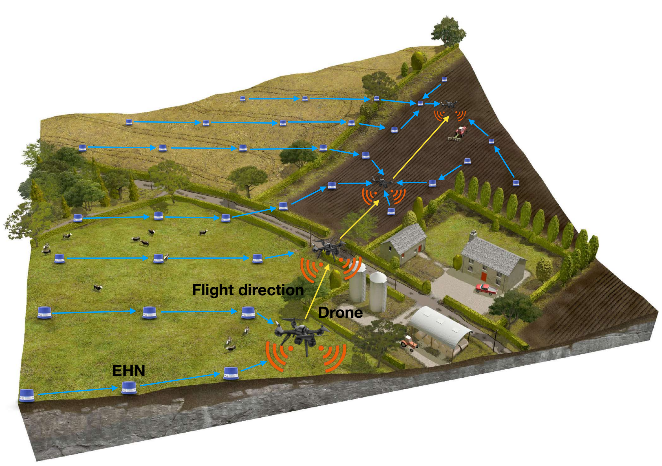


Figure 1: The application of WUR-TS in precision agriculture.

To solve the above problem, an efficient way is deploying EHNs to form a multihop network and then employ a drone as a mobile center to collect sensing data. As shown in Fig. 1, a drone is released to retrieve data. During the flight, the drone periodically broadcasts strong wake-up signals to activate EHNs with WUR. The transmission power and the flying height of the drone can be high, which allows the wake-up signal to activate EHNs far away (one or two kilometers) from the drone. The EHNs, who receive the wake-up signal, measure the environment and then transmit the data back wirelessly. Due to the size and energy constraint, the transmission power of an EHN is low, usually less than 0 dBm, which means the transmission range of the EHN can only be about 100 meters if the sensitivity of the receiver is -90 dBm [20], [21]. In this situation, data from the edge of the network cannot reach the drone directly. Therefore, EHNs need to forward packet hop-by-hop, which requires communication parties to wake up at the same time so the packet will not missed by a sleeping receiver.

In the above scenario, we use the mobile center (drone) rather than a sender in a communication party to wake-up the receiver; otherwise, the sender needs to broadcast the wake-up signal before each data transmission, which causes high energy consumption. To be specific, the sensitivity of a WUR is much lower than that of the wireless communication module. To wake-up the WUR, the strength of the wake-up signal needs to achieve -50 dBm or higher. By contrast, for wireless communication module like Atmel ATmega256RFR2 [20] and Texas Instrument CC2500 [21], the strength of the data packet only needs to reach -90 dBm or even lower for successful data reception. This indicates that the transmission power of wake-up signal needs to be 10^4 times higher than that of data packet to activate the receiver, which may not be allowed by EHNs with limited energy supply.

It is worth noting that the drone cannot simultaneously wake up all nodes in the coverage. This is because there is an interrupt delay between receiving the wake-up signal and generating an valid interrupt to activate an EHN. The interrupt delay is a variable affecting by many factors, such as the strength of the wake-up signal and the power supply voltage of an EHN. According to our experiment, the variation of the interrupt delay can over 1 ms if there is significant changes of wireless channel or power supply voltage. Due to the variation of the interrupt delay, the sender cannot know exactly when the intended receiver will be activated by the wake-up signal. In this situation, the communication parties need to wait enough time after being activated by the drone to ensure that the both sender and the receiver are awake during communication, causing extra time and energy consumption.

To synchronize the whole network through a single wake-up signal, a model is required to accurately estimate the interrupt delay of WUR, which can be used to calculate the arrival time of the wake-up signal. Afterwards, the drone only needs to broadcast the wake-up signal every several minutes to help EHNs correct the clock drift.

3.2 Research Objective

The objective of WUR-TS is to help communication nodes in an energy harvesting wireless network synchronize with the mobile center for successful communications. Due to the limited energy supply of EHN, WUR-TS subjects to a strict energy constraint at the receiver side. Therefore, EHNs need to reduce or even completely cancel sync message exchange between each other to save the energy. To achieve this goal, in WUR-TS, the wake-up signal from the mobile center has to responsible for two major tasks: (a) activate sleeping nodes, and (b) serve as a special synchronization signal.

Let $T_r(i)$ and T_p^n be the time that the mobile center sends the i^{th} wake-up signal and the propagation delay of RF signal between the mobile center and node n, respectively. Denote the delay between receiving the wake-up signal i and successfully awakened node n by $t_s^n(i)$. Let $T_l^n(i)$ be the local time recorded by node n when it is activated by the wake-up signal i. Therefore, we have that

$$T_l^n(i) = \alpha_{cd}^n \, T_r(i) + t_{co}^n + T_p^n + t_s^n(i), \tag{1}$$

where α_{cd}^n and t_{co}^n are the clock drift and clock offset with respect to the mobile center and node n.

Assume the mobile center is scheduled to broadcast the wake-up signal every T_x second, then according to (1), it can be obtained that

$$\alpha_{cd}^n = \frac{[T_l^n(i+1) - T_l^n(i)] - [t_s^n(i+1) - t_s^n(i)]}{T_x}. \tag{2}$$

Every time a new wake-up signal is received by node n, α_{cd}^n will be updated accordingly. If EHNs are scheduled to communicate every T_y seconds as in a slot-based MAC, where $T_y = T_x/N_x$, $N_x \in \mathbb{Z}$, and $N \gg 1$, then node n and its neighbors can wake-up simultaneously if they generate internal timer interrupts at the following local time, which is denoted by T_n, after receiving the i^{th} wake-up signal from the center:

$$T_n = T_l^n(i) - t_s^n(i) + n \, \alpha_{cd}^n \, T_y, \quad n = 1, 2, \ldots, N_x. \tag{3}$$

In (2) and (3), T_x and T_y are predetermined parameters, $T_l^n(i)$ is the local time recorded by EHN. Therefore, the only parameter a node needs to know is $t_s^n(i)$, which is a variable that changes with the environment. How to accurately estimate $t_s^n(i)$ is the focus of this paper.

In WUR-TS, the wake-up signal can only help nodes to estimate the clock drift. If EHNs want to calculate the clock offset between them and the mobile center, the central node needs to record the timestamp, $T_r(i)$, and then broadcasts it after transmitting the wake-up signal. The EHN, which obtains the clock drift from (3), can calculate t_{co}^n based on (1) and the local time that it is activated by the wake-up signal. The clock offset is a constant, which only needs to be calculated once in the whole synchronization process. Therefore, the time and energy spent on estimating the clock offset is negligible.

4 Wake-up Radio Receiver Design

Fig. 2 is the architecture of WUR designed for time synchronization. As shown in the figure, the antenna is connected to a Powercast P1110B chip, which integrates an impedance matching network, a rectifier, and a multistage voltage booster. The impedance matching network makes the input impedance of the antenna equal to the output impedance of the radio frequency (RF) circuit to minimize the antenna's reflection coefficient. Afterward, the rectifier converts the output voltage of the matching circuit from alternating current (AC) to direct current (DC), which is then amplified by the voltage booster.

In order to maximize the synchronization range, WUR should be able to detect a weak signal. Hence, we choose Texas Instruments LPV7215, which has a low offset voltage (about 0.7 mV at 25°C [22]), as the comparator. Compared with other comparators having larger offset voltage, the power consumption of LPV7215 is a bit high. As we measured at 2.8 V power supply, when the outputs of the comparator are low (logic 0, idle mode) and high (logic 1, active mode), the power consumptions of LPV7215 are 2.1 μW and 47.8 μW, respectively.

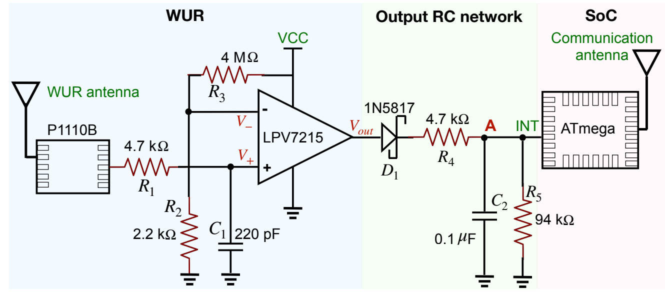


Figure 2: The WUR designed for time synchronization.

To accurately estimate the interrupt delay, we use a static reference voltage created by power supply voltage, R_2, and R_3, instead of an adaptive one at the inverting input of the comparator. Although the use of an adaptive reference voltage can increase the dynamic range of WUR for both weak and strong wake-up signal detection while reducing power consumption [23], [24], the threshold voltage of the comparator will be determined by the output voltage of the booster. Due to the effect of internal resistors and capacitor components in P1110B, the output voltage of the booster cannot immediately rise up after receiving the wake-up signal. As a result, if the reference voltage changes adaptively with the strength of the wake-up signal, it will be very difficult to accurately model the interrupt delay. Therefore, in our implementation, we choose a static reference voltage for time synchronization.

In the real world, if there are interferences having the same frequency as the wake-up signal, the voltage booster will convert the interference into DC voltage, with may toggle the output of the WUR by mistake. This problem can be solved by adding the addressing function in WUR [23], [24]. In the address-enabled WUR, the identification (ID) of the intended receiver is modulated by the on-off keying (OOK) scheme and then transmitted as the wake-up signal. After receiving the OOK wake-up signal, a low-power microcontroller is activated first to demodulate the address information and then compare it with its own address. If the addresses are matched, the microcontroller generates an interrupt signal to wake-up the main circuit for sensing and communications.

Applying the address-enabled WUR for time synchronization has three main disadvantages. First of all, it usually needs an extra microcontroller to recognize the pattern (address) of the wake-up signal, causing additional energy consumption of the receiver. Secondly, the wake-up signal cannot be short to contain address information. This significantly increases the energy consumption of the sender, especially when the transmission power of the wake-up signal is high to cover a large-scale network. Thirdly, the address decoder is essentially a digital circuit. It erases the analog feature (e.g., the signal strength and the rising time) of the wake-up signal, thereby making it difficult to accurately estimate the arrival time of the wake-up signal for high-precision time synchronization.

To balance the synchronization accuracy and the reliability of the WUR in complex radio environments, we do not use the address-enabled WUR but insert an output RC network between the SoC and the comparator. By selecting an appropriate time constant for the RC network, we can avoid most of short interferences to toggle the comparator by mistake. In the worst case where long interferences occur multiple times in a short period, the output RC network will only wake up the SoC once, and thus reduces the energy consumption caused by continuous interferences.

5 Output Models of Voltage Booster and Comparator

As illustrated in Fig. 2, to improve the sensitivity of the WUR, we need to increase the input voltage at the non-inverting input of the comparator given the strength of the wake-up signal. This can be achieved by using a multi-stage voltage booster, as will be discussed next.

In Fig. 3, we feed P1110B a short wake-up signal, and then compare the maximum output voltage of P1110B with and without the amplification of a multi-stage voltage booster. As shown in the figure, when the intensity of the incident wake-up signal is -20 dBm, the output voltage of the rectifier is about 0.7 mV, which is the minimum differential voltage required to toggle the output of the LPV7215 comparator. However, if the voltage booster is applied, the output voltage of P1110B can reach 1 mV even if the strength of the wake-up signal is lower than -45 dBm. Therefore, the multi-stage voltage booster can significantly improve the sensitivity of WUR, thereby increasing the synchronization range of WUR-TS.

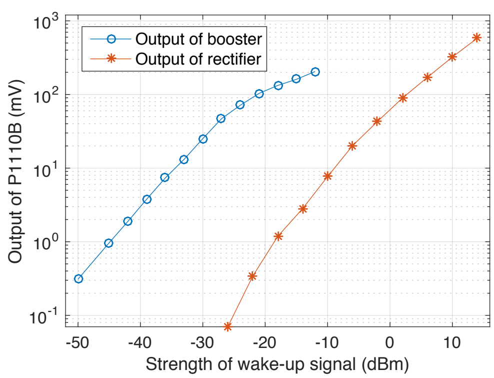


Figure 3: Output voltage of P1110B with and without amplification of a multi-stage voltage booster. The length and frequency of the wake-up signal are 1 ms and 915 MHz, respectively.

It is worth noting that although a voltage booster with more stages has a larger amplification factor, the input voltage of the booster will decrease at a given incident signal strength. Specifically, the capacitive component and the reactive component in the input impedance of the voltage booster are respectively proportional to and inversely proportional to the number of stages. As the reactive component decreases, the loaded quality factor, Q, of resonator between the antenna and the rectifier circuit becomes smaller. This reduces the voltage gain from the resonator, resulting in low voltage feed to the booster.

From the above analysis, it can be realized that the voltage booster needs to have an appropriate number of stages to maximize WUR sensitivity. Too few stages will not provide enough amplification factor; too many stages will reduce the input voltage of the booster. In both cases, the voltage that can be fed into the comparator to detect the wake-up signal is reduced. This conclusion is different from that obtained in [24], which aims to maximize the input voltage of the rectifier rather than the output of the voltage booster.

Now, we study the relationship between the strength of the wakeup signal and the output voltage of the voltage booster. The results will be used to calculate the interrupt delay in Section 6. Let I_r and V_m represent the strength of the incident wake-up signal and the maximum output of the voltage booster, respectively. Then we can calculate V_m from I_r using the following modified sigmoid function:

$$V_m = \rho_3 + \frac{\rho_2}{e^{\rho_1(-I_r-\rho_4)}+1} \tag{4}$$

where $[\rho_1,\ \rho_2,\ \rho_3,\ \rho_4]$ = [0.16, 0.29, -0.0036, 17.3] are non-linear least square (NLS) coefficients calculated via Levenberg-Maquardt algorithm. The units of V_m and I_r are V and dBm, respectively. In Fig. 4, we change I_r and then compare V_m calculated from (4) with the value obtained from the experimental measurements. The results prove that the sigmoid function is a good approximation to the real data.

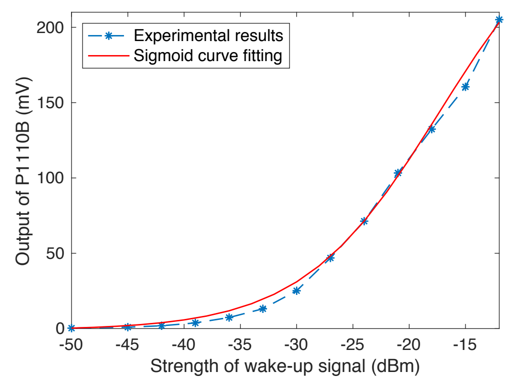


Figure 4: V_m with respect to I_r.

The output of the voltage booster does not need to achieve the maximum value, V_m, to toggle the comparator. As shown in Fig. 2, when the voltage difference between V_+ and V_- is larger than the input offset of the comparator, V_{out} will change from "low" to "high". Therefore, in order to toggle the output of the comparator, V_+ should satisfy

$$V_+ > V_{0ff} + \frac{\gamma}{1+\gamma} V_{cc}, \tag{5}$$

where $\gamma = R_2/R_3$ and $V_{off} = 0.7$ mV for the LPV7215 comparator.

According to the experiment, we observed that when $R_3 = 4\,\mathrm{M}\Omega$ and $V_{cc} = 3.3\,V$, R_2 cannot be less than $1\,\mathrm{k}\Omega$; otherwise, the output of the comparator will become unreliable, causing a large number of false alarms on wake-up signal detection. When $R_2 = 1\,\mathrm{k}\Omega$, WUR can sense the wake-up signal as low as -45 dBm with 95% confidence level. If R_2 increases to $2.2\,\mathrm{k}\Omega$, then V_+ must be greater than 2.5 mV to toggle the output of the comparator. In this case, the sensitivity of WUR decreases to -43.5 dBm.

6 Delay Model

In this section, we first outline the interrupt delay generated by WUR. Then, how to model each component of the interrupt delay through experimental measurements and theoretical analysis will be discussed.

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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

6.1 Overview

After receiving the wake-up signal, if WUR can immediately activate the associated SoC, the synchronization process will be very simple. In this case, there is only a propagation delay between the sender and receiver, which is ignorable due to the high propagation speed of radio waves. Therefore, if the interrupt delay is zero, the central node can send two wake-up signals at a fixed interval, and the EHN can calculate the clock skew based on the receiving time of these two signals.

Unfortunately, in practical applications, due to the internal resistors and capacitors of the voltage booster as well as the RC networks added at the input and output of the comparator, WUR cannot create an effective interrupt immediately after receiving the wake-up signal. Moreover, the interrupt delay is not constant, but varies with the strength of the wake-up signal and the power supply voltage. Therefore, we need to estimate the delay caused by each component of WUR in real-time to achieve high-precision time synchronization.

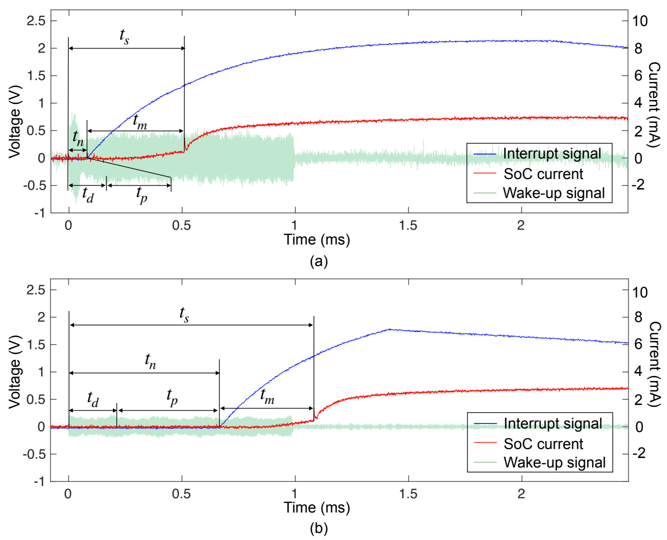


Figure 5: Examples of voltage and current waveforms, where the amplitude of the wake-up signal is multiplied by 50 to make it readable in the figure. (a) Strong wake-up signal, and (b) weak wake-up signal.

To give insight into each delay component, we set the power supply voltage to 3.3 V, and then show the interrupt signal measured at point A of Fig. 2, the received wake-up signal, and the power supply current of the ATmega256RFR2 SoC in Fig. 5, where t_s is the interrupt delay, i.e., the delay between receiving the wake-up signal and generating the effective interrupt to successfully activate the sleeping SoC. As shown in the figure, t_s can be divided into two parts: the input delay t_n and the output delay t_m. Furthermore, t_n can be divided into two parts: (a) the charging delay generated by the voltage booster and the input RC network, and (b) the propagation delay of the comparator. Moreover, t_m is the delay caused by the output RC network. In the rest of this section, we analyze each delay component in detail.

6.2 Delay of Voltage Booster and Input RC Network

Let t_d be the delay introduced by the voltage booster and input RC network. To calculate t_d, we use Fig. 6 to model the voltage booster and input RC network. In the figure, V_m is an ideal voltage source. Given the strength of the wake-up signal, we can get V_m by (4).

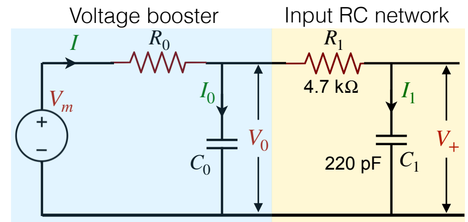


Figure 6: Model of the voltage booster and input RC network.

Based on Fig. 6 and the energy storage characteristic of the capacitor, we can write the following differential equations to calculate the voltage change at the non-inverting input of the comparator with time:

$$\begin{cases} I = I_0 + I_1, \\ I_0 = \dfrac{\mathrm{d}Q_0}{\mathrm{d}t}, \quad I_1 = \dfrac{\mathrm{d}Q_1}{\mathrm{d}t}, \\ V_0 = I_1 R_1 + \dfrac{Q_1}{C_1}, \\ IR_0 + V_0 = V_m. \end{cases} \tag{6}$$

In (6), R_0 and C_0 are the output resistance and output capacitance of the voltage booster; Q_0 and Q_1 are the charge on capacitors C_0 and C_1 at time t, respectively; I, I_0, and I_1 are the current flowing through R_0, C_0, and C_1, respectively.

After a simple derivation, we can obtain the folllowing second-order non-homogeneous differential equation from (6):

$$R_0R_1C_0\frac{\mathrm{d}^2Q_1}{\mathrm{d}t^2}+\left(\frac{R_0C_0}{C_1}+R_0+R_1\right)\frac{\mathrm{d}Q_1}{\mathrm{d}t}+\frac{Q_1}{C_1}=V_m. \tag{7}$$

The solution of (7) is

$$Q_1 = \eta_1 e^{r_1 t} + \eta_2 e^{r_2 t} + V_m C_1, \tag{8}$$

where

$$r_1=\frac{-\left(\frac{1}{R_1C_1}+\frac{R_0+R_1}{R_0R_1C_0}\right)+\sqrt{\left(\frac{1}{R_1C_1}+\frac{R_0+R_1}{R_0R_1C_0}\right)^2-\frac{4}{R_0R_1C_0C_1}}}{2}, \tag{9}$$

and

$$r_2=\frac{-\left(\frac{1}{R_1C_1}+\frac{R_0+R_1}{R_0R_1C_0}\right)-\sqrt{\left(\frac{1}{R_1C_1}+\frac{R_0+R_1}{R_0R_1C_0}\right)^2-\frac{4}{R_0R_1C_0C_1}}}{2}. \tag{10}$$

In (8), η_1 and η_2 are two coefficients that can be calculated by the following two boundary conditions: (a) $Q_1 = 0$ when $t = 0$, and (b) $Q_1 = V_m C_1$ when $C_0 = 0$ and $t \to \infty$. Submitting the two boundary conditions to (8) separately and solving the equation set, it can be obtained that $\eta_1 = -V_m C_1$ and $\eta_2 = 0$. Then, according to $Q_1 = C_1 V_+$, we have that

$$V_+ = V_m(1 - e^{r_1 t}). \tag{11}$$

According to (5), (11), and the definition of t_d, it can be obtained that

$$V_m(1 - e^{r_1 t_d}) = V_{0ff} + V_{cc}\frac{\gamma}{1+\gamma}. \tag{12}$$

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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

From (12) we have that

$$t_d = \frac{1}{r_1} \ln\left(1 - \frac{V_{off}}{V_m} - \frac{V_{cc}\,\gamma}{V_m\,(1+\gamma)}\right). \tag{13}$$

As revealed in (9) and (13), t_d is affected by R_0 and C_0. In a practical voltage booster, R_0 and C_0 are not constant, but vary with the strength of the received wake-up signal. In Section 6.3, we will model R_0 and C_0 according to experimental measurements.

6.3 Output Characteristics of Voltage Booster

6.3.1 Model of Output Resistance

To get R_0, we can calculate the ratio of the open-circuit voltage to the shortcut current of the booster under different I_r. As shown in Fig. 7, R_0 is inversely proportional to the strength of the received wake-up signal. When I_r is lower than -35 dBm, R_0 is as high as 200 kΩ. As I_r further decreases, R_0 remains unchanged. When I_r increases from -35 dBm to -20 dBm, R_0 gradually decreases from 200 kΩ to 22 kΩ. When the strength of the wake-up signal is greater than -20 dBm, R_0 gradually approaches zero.

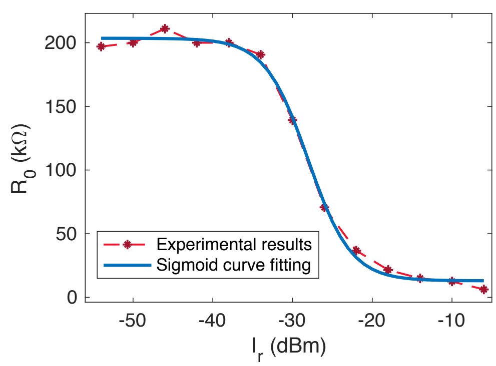


Figure 7: The output resistance of the voltage booster with respect to the strength of the received wake-up signal.

From Fig. 7, it can be observed that R_0 is an S-shape curve with respect to I_r. Therefore, we can use the following modified sigmoid function to describe R_0:

$$R_0 = \beta_3 - \frac{\beta_2}{e^{-\beta_1(I_r+\beta_4)}+1}, \tag{14}$$

where [β_1, β_2, β_3, β_4] = [0.37, 190.5, 203.5, 28.0] are the NLS coefficients obtained from the Levenberg-Maquardt algorithm; the units of I_r and R_0 are dBm and kΩ, respectively. In Fig. 7, we compare R_0 obtained from (14) with the experimental measurements. The results show that (14) matches the real curve very well. Next, we introduce how to model C_0.

6.3.2 Model of Output Capacitance

As illustrated in Fig. 6, we can use an ideal voltage source in series with R_0 and C_0 to model the voltage booster. Let τ_0 be the time constant of RC network, which is the product of R_0 and C_0. By disconnecting P1110B from the main circuit, and then measuring the time required for the output voltage of the booster to rise from zero to 63.2% of the maximum value (i.e., V_m), τ_0 will be available, as marked in Fig. 6. Then, according to the definition of the RC time constant, we get $C_0 = \tau_0/R_0$.

In Fig. 8(b), we show the change of C_0 with I_r. As illustrated in the figure, when I_r is between -50 dBm and

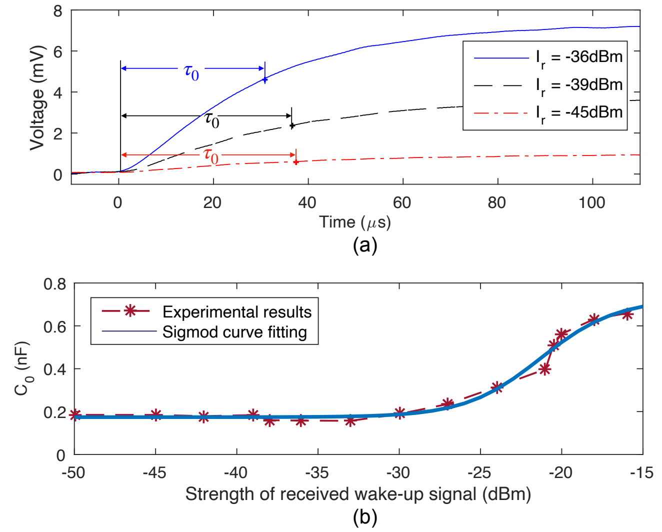


Figure 8: The output characteristics of the voltage booster. (a) The output voltage of the booster over time. (b) C_0 with respect to I_r.

-15 dBm, C_0 is proportional to the strength of the received wake-up signal. From (9), (13), and Fig. 8(b), it can be realized that the impact of C_0 on t_d cannot be ignored, especially when the wake-up signal is strong. Specifically, when I_r is less than -30 dBm, the output capacitance of the voltage booster is about 200 pF, which is comparable with C_1 in the input RC network. As I_r increases, C_0 will rise rapidly and make significant impact on t_d.

Similar to R_0, we can use the following modified sigmoid function to describe C_0 with I_r:

$$C_0 = \lambda_3 + \frac{\lambda_2}{e^{-\lambda_1(I_r+\lambda_4)}+1}, \tag{15}$$

where the units of C_0 and I_r are nF and dBm, respectively. In (15), coefficients [β_1, β_2, β_3, β_4] = [0.43, 0.55, 0.17, 21.3] are the NLS coefficients obtained from the Levenberg-Maquardt algorithm. As depicted in Fig. 8(b), the C_0 calculated from (15) matches the experimental data very well.

6.4 Propagation delay of Comparator

The propagation delay of the comparator is the time required for the output of the comparator to reach the 50% point of a transition after the differential input signals cross the offset voltage. The propagation delay is affected by many factors, such as the junction temperature, common-mode voltage, input overdrive, and supply voltage.

Under normal circumstances, the propagation delay of LPV7215 exceeds 10 μs. This value is comparable to the delay caused by the input RC network and cannot be ignored during time synchronization. Let t_p and V_{od} represent the propagation delay and the input overdrive of the comparator, respectively. According to the data provided in the data sheet [22], we choose the following exponential function to describe t_p with V_{od}:

$$t_p = \kappa_3 + \frac{\kappa_2}{e^{\kappa_1(V_{od}+\kappa_4)}-1}, \tag{16}$$

where the units of t_p and V_{od} are μs and mV, respectively. According to the definition of input overdrive and the

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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

architecture of our WUR, V_{od} can be calculated as follows:

$$V_{od} = V_m - V_{cc}\frac{R_2}{R_2 + R_3} - V_{off}. \tag{17}$$

In (16) [$\kappa_1, \kappa_2, \kappa_3, \kappa_4$] = [0.02, 100, 10, 0.5], [0.034, 299, 5.8, 102], and [0.042, 1500, 13, 99] for $V_{cc} = 1.8\,\text{V}$, 2.8 V, and 5 V, respectively. The accuracy of t_p obtained from (16) will be evaluated in Section 9.2.

In the real world, t_p may vary with the junction temperature and the common-mode voltage applied to the comparator input. In general, the $t_p - V_{od}$ curves shown in Fig. 15 moves upward as the junction temperature increases. Therefore, in order to further improve the accuracy of (16), the coefficient κ_3 can be written as an increasing function of the junction temperature. In addition, $t_p - V_{od}$ curves shifts to the right as the common-mode voltage decreases. Accordingly, the coefficient κ_4 becomes an increasing function of the common-mode voltage. In this paper, we ignore the effect of the junction temperature and common-mode voltage on t_p because their impact is much less than the power supply voltage and the wake-up signal strength.

6.5 Delay of Output RC Network

As shown in Fig. 2, let V_f be the forward voltage drop of D_1, which is 0.45 V for the Schottky diode 1N5817; t_m is the duration that the comparator keeps output high to wake up the sleeping SoC. According to the characteristics of the Schmitt trigger [25], the voltage of C_2 should reach $\alpha \times V_{cc}$ to wake up the SoC through an external interrupt, where α is a scale factor. In a practical SoC, α is not a constant, but is affected by the power supply voltage.

From experimental measurements, we observed that $\alpha = 0.54$ when $V_{cc} = 2.6\,\text{V}$, and then linearly decreases to 0.48 when V_{cc} drops to 3.8 V. According to this observation, it can be obtained that $\alpha \approx -0.05V_{cc} + 0.67$. Based on the charging equation of the RC circuit, we can estimate the minimum time required to activate the sleeping SoC as follows:

$$t_m = -\tau_b \ln\left(1 - \frac{\alpha\, V_{cc}}{V_A}\right), \tag{18}$$

In (18), $\tau_b = R_p\, C_2$ is the time constant of the output RC network, which determines the robustness of WUR against interference. When the output of the comparator changes to "high", it takes a certain time to charge the capacitor C_2 to generate a valid interrupt. Therefore, short interference will not be able to wake up the EHN. R_p is the parallel resistance of R_4 and R_5. V_A is the voltage at point A of Fig. 2, which is

$$V_A = (V_{cc} - V_f)\frac{R_5}{R_4 + R_5}. \tag{19}$$

According to the above analysis, there is a trade-off between the anti-interference ability of a WUR and the energy efficiency of the central node. To be specific, increasing the time constant of the output RC network can protect EHN from short interferences. However, to successfully activate the SoC, the duration of the wake-up signal had to be extended, which increases the energy consumption of the central node.

7 Interrupt Delay Calculation

Based on the models introduced in Section 5 and Section 6, the SoC can estimate the interrupt delay given the power supply voltage and the wake-up signal strength. The detailed steps are as follows:

Process of Calculating Interrupt Delay

Initialization: After being awakened by the wake-up signal, the SoC measures V_{cc} and V_m several times to get their average values.

Step 1: Calculate the strength of the received wake-up signal, I_r, according to the inverse function of (4).
Step 2: Calculate R_0 and C_0 from (14) and (15), respectively.
Step 3: Based on (9) and (13), the SoC get t_d.
Step 4: Estimate the propagation delay of the comparator, t_p, through (16) and (17).
Step 5: The SoC calculates the input delay by $t_n = t_d + t_p$.
Step 6: Base on (18), the SoC get t_m.
Step 7: The interrupt delay is available through $t_s = t_m + t_n$.

In Fig. 9, we show the input delay of WUR obtained from step 1 to step 5. As illustrated in the figure, t_n is proportional to R_2. This is because t_d and t_p in t_n increase as R_2 increases by affecting γ and V_{od} in (13) and (16), respectively. In addition, it can be observed that t_n decreases significantly with the increase of I_r, which will be analyzed below.

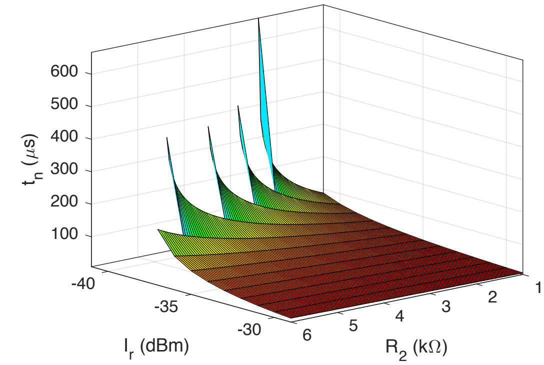


Figure 9: t_n with respect to R_2 and I_r, where $V_{cc} = 2.8\,\text{V}$.

When I_r is lower than $-28\,\text{dBm}$, the output capacitor of the voltage booster, C_0, is almost a constant, as shown in Fig. 8(b). However, when I_r increases from $-40\,\text{dBm}$ to $-28\,\text{dBm}$, the output resistance of the booster, R_0, decreases rapidly, as shown in Fig. 7. Therefore, the coefficient r_1 in (13) becomes larger as I_r increases, thereby reducing t_d. Furthermore, as illustrated in Fig. 3, the maximum output of the voltage booster, V_m, is proportional to I_r. According to (13), t_d is inversely proportional to V_m. Therefore, feeding a strong wake-up signal to WUR can greatly reduce t_d. Moreover, as revealed in (16), the propagation delay of the comparator is inversely proportional to V_{od}, which increases as the wake-up signal becomes stronger. Therefore, t_n is inversely proportional to I_r.

Compared with battery-power wireless nodes, the supply voltage of EHN may vary significantly in a short time (several milliseconds) with the activity of the node. In Fig. 10, we use a 1 mF capacitor as energy storage to drive an EHN to show how the power supply voltage and ATmega256RFR2 SoC current change with the data transmission. In the figure, the SoC is programmed to periodically send data packets with the IEEE 802.15.4 protocol. In the experiment, the data payload is 105 octets and the transmission power is 3.5 dBm. As shown in the figure, due to the

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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

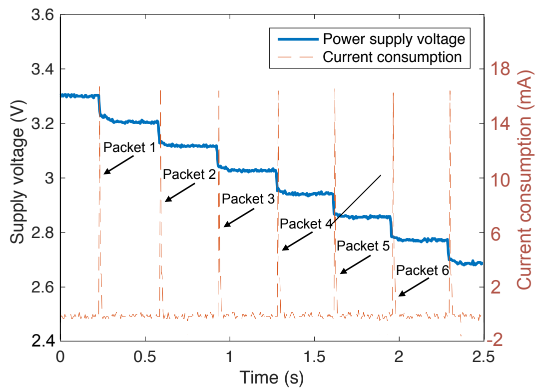


Figure 10: Power supply voltage and current change with the transmission activity, where a 1 mF capacitor is used as the energy storage component.

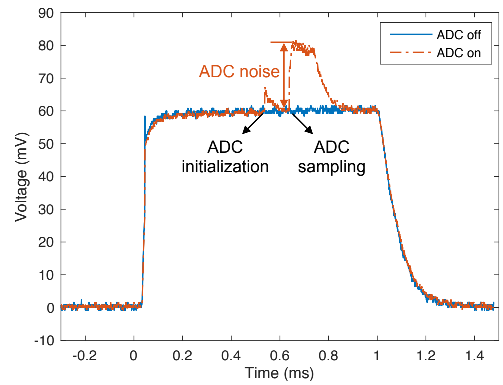


Figure 11: Comparison between the output of P1110B measured with and without ADC noise when receiving a wake-up signal, where $I_r = -25$ dBm and $V_{cc} = 2.8$ V.

low capacity of energy storage and high energy consumption of data transmission, the power supply voltage drops rapidly after each transmission activity. Therefore, the SoC needs to measure V_{cc} in real-time to calculate the interrupt delay of WUR for high-precision time synchronization.

8 ADC Noise Mitigation

This section first discusses the ADC noise problem caused by the high output impedance of the voltage booster. This problem would lead to inaccurate sampling of V_m during time synchronization. After that, a solution is provided to mitigate the effect of ADC noise on voltage sampling.

8.1 ADC Noise Problem

As described in Section 7, WUR-TS needs to know V_m in order to calculate the input delay of WUR. However, V_m is always contaminated by circuit noise. Therefore, the SoC needs to measure the voltage at the non-inverting input of the comparator multiple times, and then do the average to reduce the noise impact. However, in addition to circuit noise, the internal ADC introduces strong interference during ADC conversion, which cannot be eliminated by the average operation.

The internal ADC in ATmega256RFR2 employs a sample-and-hold circuit to sample the data. In this circuit, when the switch is closed, a hold capacitor needs to discharge through the source impedance. According to the user manual, the internal ADC of ATmega256RFR2 is optimized for a signal source with an output impedance of 3 kΩ or less. However, as measured in Fig. 7, the output resistance of P1110B can over 200 kΩ if the wake-up signal is weak. The source impedance will be even higher after considering its imaginary part. In this case, if the ADC acquisition time is short, the undischarged voltage remaining on the hold capacitor cannot drop under 0.5 least significant bit (LSB), thereby causing interference.

In Fig. 11, we feed a 1 ms wake-up signal to WUR, and then use an oscilloscope to observe the output of P1110B with and without ADC noise. The results show that if the ADC is turned off, the output of P1110B has only random fluctuations caused by circuit noise. By contrast, once the ADC is turned on, V_m suddenly increases during voltage sampling. As illustrated in the figure, when the strength of the received wake-up signal is -25 dBm and the power supply voltage is 2.8 V, the ADC noise can reach 20 mV, which is 30% of the true V_m. In order to accurately estimate the interrupt delay of WUR, this strong noise cannot be ignored.

It is worth noting that the average voltage of the ADC noise is not constant, but changes with the ADC acquisition time and the wake-up signal strength. The former determines the discharge level of the hold capacitor, while the latter determines the discharge speed of the hold capacitor by affecting the source impedance, as analyzed in Fig. 7 and Fig. 8(b). Therefore, we cannot simply subtract a constant from the V_m measurement to estimate the true V_m.

In order to mitigate the impact of ADC noise on V_m measurements, we need to extend the minimum ADC acquisition time, which can be estimated by the following equation [20]:

$$\tau_c = (Z_0 + 2000) * 0.097\,\text{ns}, \tag{20}$$

where Z_0 is the output impedance of the signal source. As shown in Fig. 7, when the strength of the received wake-up signal is lower than -35 dBm, the real part of P1110B output impedance will exceed 200 kΩ. In this case, τ_c will be greater than 19.6 μs, which means that the ADC sampling rate must be lower than 51 ks/s.

8.2 Solution of ADC Noise Problem

Reducing the sampling rate can reduce ADC noise. However, it will take the SoC more time to measure the average V_m, thereby increasing energy consumption. In order to eliminate the effect of ADC noise while maintaining a high sampling rate, there are two solutions: (a) hardware impedance transformation, and (b) algorithm correction. For the first solution, a buffer op-amp can be inserted between P1110B and ADC to minimize the output impedance of the P1110B. However, the use of external buffer op-amp consumes extra energy, which should be avoided in energy harvesting wireless networks with limited energy supply. Therefore, we will focus on the second solution.

Let $\tilde{V}_m$ be the voltage measured by the ADC at the non-inverting input of the comparator, that is, actual V_m added by the ADC noise. To use algorithm correction, we scan $\tilde{V}_m$ under different I_r, and then use the following logarithmic function to calculate I_r corresponding to $\tilde{V}_m$:

$$I_r = \zeta_1 \log\left(\zeta_2 \tilde{V}_m - \zeta_3\right) - \zeta_4, \tag{21}$$

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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

where $[\zeta_1, \zeta_2, \zeta_3, \zeta_4] = [5.43, 2.25, 0.06, 15]$ are NLS coefficients obtained from Levenberg-Maquardt algorithm; the units of $\tilde{V}_m$ and I_r are V and dBm, respectively. After getting I_r from $\tilde{V}_m$, the SoC can calculate the actual V_m according to (4).

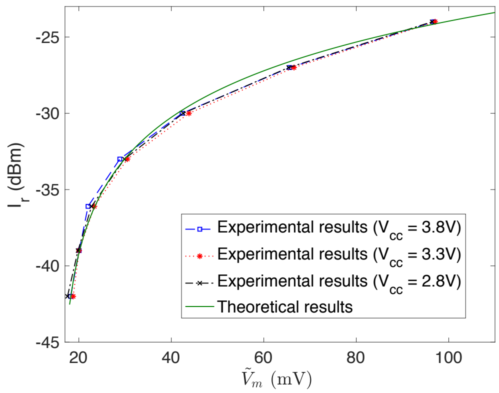


Figure 12: I_r with respect to $\tilde{V}_m$ and V_{cc}.

In Fig. 12, we change V_{cc} and $\tilde{V}_m$, and then compare I_r obtained from (21) with experimental measurements. From the figure, it can be observed that the I_r-$\tilde{V}_m$ curves measured under different V_{cc} are overlapping. This indicates that the ADC noise does not change with the power supply voltage. Additionally, as illustrated in the figure, the I_r curve calculated by the logarithmic function matches the experimental results very well. Therefore, (21) is a good approximation to the real I_r-$\tilde{V}_m$ curve.

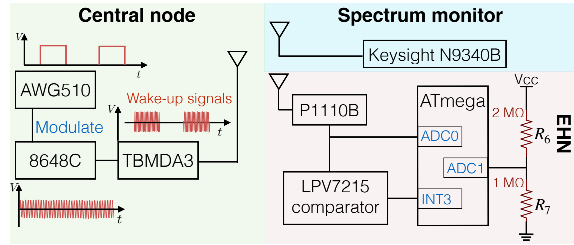
9 Performance Evaluation

This section evaluates the performance of WUR-TS through experiments. We first introduce the experimental setup. Then, the accuracy of each delay component is assessed separately. Afterward, the energy consumption of WUR-TS is measured. Finally, we evaluate the synchronization accuracy of WUR-TS under different settings.

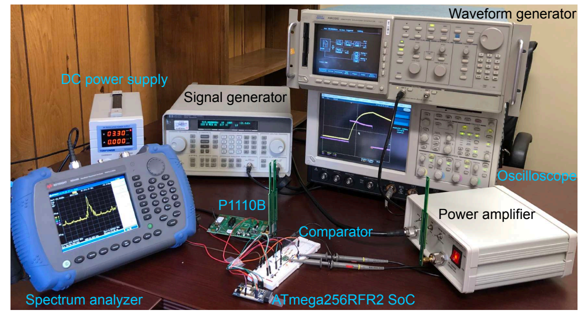
9.1 Experimental Setup

Fig. 13 is the platform built for WUR-TS. At the central node, the Tektronix AWG510 arbitrary waveform generator is programmed to produce a square wave with a specific pulse width. Its output is connected to the modulation input of the HP 8648C signal generator, which creates 915 MHz continuous wave (CW) as a carrier signal. After the signal is modulated, the HP 8648C outputs a single-frequency pulse wave as the wake-up signal, which is then amplified by the TekBox TBMDA3 RF power amplifier, and then transmitted by a 1 dBi omnidirectional patch antenna or a 6 dBi directional (120-degree) patch antenna. The Keysight N9340B spectrum analyzer is employed to monitor the intensity of the incident wake-up signal.

In the experiment, Microchip ATmega256RFR2 SoC serves as the controller and wireless transceiver of the EHN. During the sleep mode and the active mode, a 32.768 kHz crystal oscillator and a 8 MHz RC circuit are used to drive the ATmega256RFR2 SoC, respectively. Between adjacent synchronizations, the EHN's clock will drift with time. To correct the clock skew, the central node broadcasts the wake-up signal periodically, and then EHN can calculate the clock drift based on the local time that receives the



(a)



(b)

Figure 13: Experiment platform built for WUR-TS. (a) Diagram of the platform. (b) Photo of the platform.

neighboring wake-up signal. The shorter the time interval between adjacent wake-up signals, the more accurate the clock drift that EHNs can estimate, and the more energy consumed by time synchronization. Therefore, how often the EHN can correct the clock drift is affected by its energy harvesting rate.

As introduced in Section 7, WUR-TS needs to know the power supply voltage, V_{cc}, in order to calculate the input delay. To successfully drive the SoC, V_{cc} must be higher than 1.8 V, which is greater than the internal reference voltage of ATmega256RFR2 (1.5 V or 1.6 V). To avoid ADC code overflow, the voltage dividers, R_6 and R_7, are used to reduce the voltage measured by ADC. The total resistance of R_6 and R_7 must be high to minimize the current flowing through the voltage dividers.

In order to get V_m, we let ADC sample the non-inverting input of the comparator, as shown in Fig. 13(a). In the experiment, the resolution and the reference voltage of ADC are set to 10 bits and 1.6 V, respectively. Accordingly, the minimum V_m that the ADC can measure is about 1.6 mV V_m, corresponding to -43 dBm wake-up signal, as shown in Fig. 3. To wake up the SoC reliably, the output of the LPV 7215 comparator is connected to the INT3 pin of ATmega256RFR2 through the RC network illustrated in Fig. 2 for interrupt detection.

9.2 Delay Model Accuracy

In Fig. 14, we show how the output delay of WUR, t_m, changes with the power supply voltage, where the theoretical result comes from (18). As can be observed from the figure, when the power supply voltage is lower than 3.2 V, the theoretical result is very consistent with the experimental data. As shown in the figure, the power supply voltage has a significant impact on the output delay of WUR. For example, when V_{cc} is 3.8 V, the output delay is only 392 μs. However, if V_{cc} is reduced to 2.6 V, then t_m will increase to 516 μs, which is 31.6% longer than that measured at $V_{cc} = 3.8$ V.

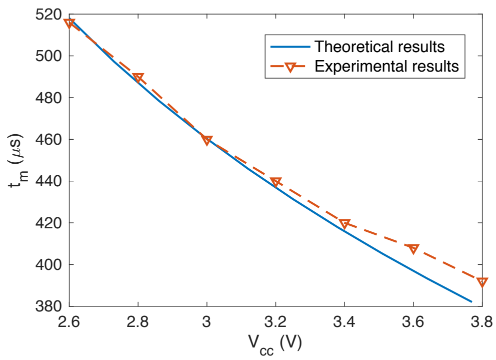


Figure 14: Output delay of WUR.

In Fig. 15, we compare the propagation delay of the comparator calculated by (16) with the data obtained from the data sheet of LPV7215 [22]. The results show that the propagation delay obtained from (16) matches the results published on the data sheet very well. As illustrated in the figure, the propagation delay in the practical comparator decreases as the input overdrive increases. Taking $V_{cc} = 5$ V as an example, when $V_{od} = 10$ mV, the propagation delay is as high as 29 μs; if V_{od} is greater than 80 mV, i.e., $I_r \approx -28.5$ dBm, then t_p gradually approaches 14 μs.

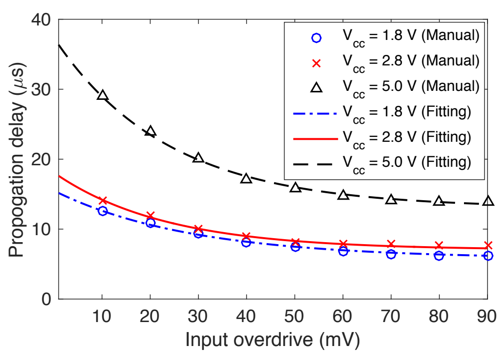


Figure 15: Propagation delay of LPV7215 comparator with respect to the input overdrive, where the common-mode voltage is 0.5 V and the junction temperature is 25°C.

Let $\hat{t}_n$ represent the estimated t_n obtained from steps (1) to (5) in Section 7. Denote the estimation error of t_n by e_n, which is $e_d = t_n - \hat{t}_n$. In Fig. 16, we show how e_n changes with I_r and R_2. As shown in the figure, when the wake-up signal strength is high, e_n is small. However, once I_r is lower than -33 dBm, the estimation error increases rapidly. This is because the circuit noise becomes comparable with V_m, thus unreliably triggering the comparator output.

From Fig. 16, it can be observed that the impact of R_2 on e_n becomes heavier as I_r decreases. For instance, if I_r is higher than -25 dBm, when R_2 is changed from 1.3 kΩ to 6 kΩ, e_n is very small, which varies between 1.2 μs and 1.6 μs. If I_r decreases to -34 dBm, e_n is only 1.2 μs when $R_2 = 2$ kΩ. However, if we do not change I_r but only increase R_2 to 6 kΩ, the estimation error of e_n will reach -14.6 μs.

In our model, we use a typical value, 0.7 mV, measured at 25°C as the offset voltage of the comparator. However, in applications where the ambient temperature changes

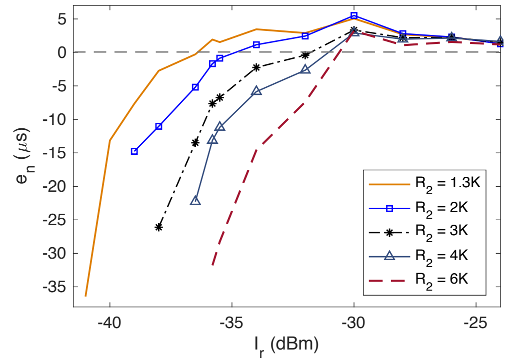


Figure 16: Estimation error of t_n with respect to I_r, where the power supply voltage, the junction temperature, and the common-mode voltage are 2.8 V, 25°C, and 0 V, respectively. Each point is an average of 15 measurements.

significantly with time, the input offset may deviate from the typical value. According to the data listed in [22], the input offset average drift of LPV7215 comparator is only ± 1 μV/°C, which is calculated by dividing the change in offset voltage at temperature extremes into the total temperature change. Let T_j be the current junction temperature, then the current offset voltage of the comparator at T_j, is

$$V_{off}(T_j) = 0.7\,\text{mV} + (T_j - 25°\text{C}) \times 1\,\mu\text{V}/°\text{C}. \tag{22}$$

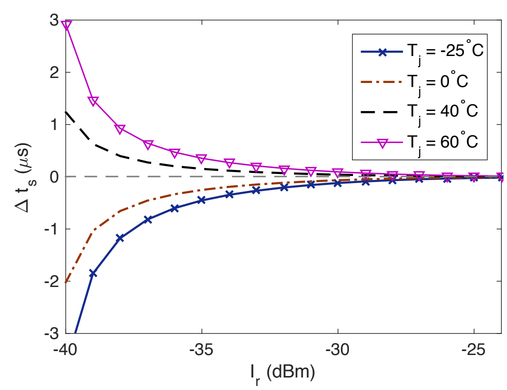


Figure 17: Impact of offset drift on sync accuracy of WUR-TS.

The impact of the offset drift on the synchronization accuracy is evaluated in Fig. 17. In the figure, Δt_s is the synchronization error caused by the offset drift, which is the difference of t_s when V_{off} are 0.7 mV and that calculated from (22) with different T_j. From the figure, it can be observe that when $I_r \geq -37$ dBm, even if T_j is higher than 60°C or lower than -25°C, the synchronization error caused by the offset drift of the comparator is less than 1 μs. Compared with total synchronization error of WUR-TS that will be shown in Fig 19, Δt_s is negligible, especially when the strength of the received wake-up signal is strong.

9.3 Energy Consumption

Fig. 18 shows the power consumption of the ATmega256RFR2 SoC during a single time synchronization process, where f_{osc} is the system clock frequency. As illustrated in the figure, although the SoC consumes more power when f_{osc} is high (8 MHz), the time spent on time synchronization can be greatly reduced. This is because

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/TMC.2021.3064374, IEEE Transactions on Mobile Computing

running at a high system clock frequency allows the SoC to measure V_{cc} at a high sampling rate while calculating the interrupt delay of WUR faster. Moreover, at a given f_{osc}, the power consumption of the SoC increases slightly as the power supply voltage increases. However, a higher power supply voltage allows the SoC to run more reliably at a high system clock frequency.

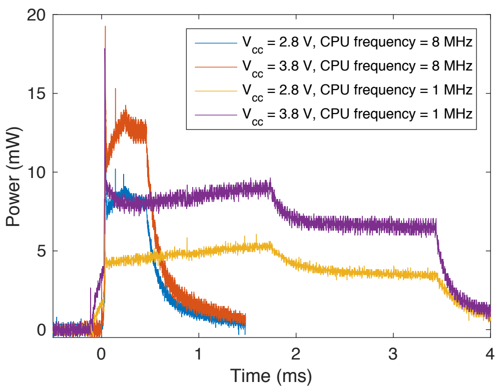


Figure 18: The power consumption of ATmega256RFR2 during a single time synchronization process, in which the ADC samples V_{cc} and V_m 15 times to do the average. The ADC sampling rates are 25 ks/s and 200 ks/s at f_{osc} = 1 MHz and 8 MHz, respectively.

If no wake-up signal is received, the ATmega256RFR2 SoC will stay in the deep sleep mode; the power consumption is as low as 1.08 μW. In this case, the comparator LPV7215 outputs zero and consumes only 2.1 μW of power. Therefore, the total power consumption of EHN is about 3.2 μW. If EHN received a wake-up signal, the SoC will be activated for time synchronization. In this situation, the power consumption of the entire circuit mainly determined by the SoC, which is 4.8 mW at V_{cc} = 2.8 V and f_{osc} = 1 MHz or 12.9 mW at V_{cc} = 3.8 V and f_{osc} = 8 MHz.

According to Fig. 18, in order to calculate the interrupt delay of WUR for each time synchronization, the EHN will consume about 3.6 μJ, 5.4 μJ, 15.0 μJ, and 26.6 μJ of energy when $\{V_{cc}, f_{osc}\}$ are {2.8 V, 8 MHz}, {3.8 V, 8 MHz}, {2.8 V, 1 MHz}, and {3.8 V, 1 MHz}, respectively. This result indicates that when running WUR-TS, increasing the system clock frequency while reducing the power supply voltage can significantly save the energy of the EHN. Next, we analyze the energy consumption of the central node.

The energy consumption of the central node is determined by the synchronization range and the required synchronization accuracy. As will be introduced in Section 9.4, WUR-TS can have a good performance when the strength of the received wake-up signal is higher than −33 dBm. Assume that the frequency of the wake-up signal is 915 MHz, and all nodes are equipped with 6 dBi antennas for signal transmission and reception. To achieve the synchronization range of 10 m, 100 m, and 1 km, the transmission power of the central node should be at least 4.7 mW, 0.47 W, and 47 W, respectively. Because the length of a wake-up signal is 1 ms, the corresponding energy consumed by the central node for each time synchronization will be 4.7 μJ, 0.47 mJ, and 47 mJ, respectively. If EHNs correct the clock offset less frequently (for example, do time synchronization every 5 minutes), this energy consumption is completely affordable in most applications.

When performing WUR-TS, the energy consumption of the EHN is much lower than timestamp-based synchronization methods since there is no idle listening or message exchange in the network. For example, if the wireless network runs the IEEE standard 802.15.4 protocol to exchange timestamps. Assume that the size of each timestamp is 40 octets, that is, the MAC service data unit (MSDU) is 40 octets. After considering the overhead of the IEEE 802.15.4 data frame (physical preamble, MAC header, frame check sequence, etc.) and the wake-up delay, the ATmega256RFR2 SoC takes about 4 ms to receive a timestamp message [26]. In receiving mode, the current consumption of ATmega256RFR2 is 10 mA. Therefore, when the power supply voltage is 2.8 V, the energy consumed to receive a timestamp is 112 μJ , which is 31 times higher than the WUR-TS method for each time synchronization. If a node needs to receive multiple timestamp messages for each synchronization process, the energy consumption will be even higher.

9.4 Synchronization Accuracy

In Fig. 19, we change the strength of the received wake-up signal, and then show the synchronization accuracy of WUR-TS, which is the difference between the actual time and the estimated time of the wake-up signal reaching the EHN. As illustrated in the figure, when the power supply voltage is 2.8 V and the strength of the received wake-up signal is higher than −33 dBm, WUR-TS can achieve a synchronization accuracy of at least 3 μs. However, with the decrease of I_r, the synchronization accuracy will drop quickly. For instance, if the wake-up signal strength is −40 dBm and the power supply voltage is 2.8 V, the synchronization accuracy of WUR-TS is only 80 μs.

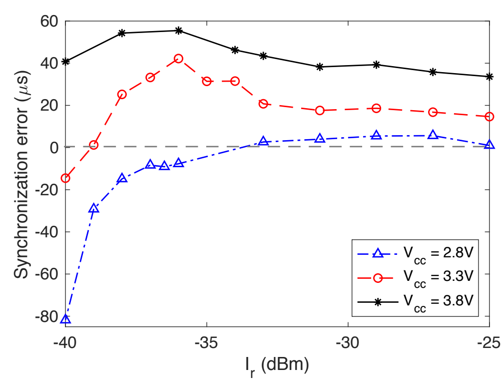


Figure 19: Uncalibrated synchronization accuracy of WUR-TS under different settings

In addition, from Fig. 19 it can be observed that as the power supply voltage increases, the synchronization accuracy of the WUR-TS decreases. This is caused by two factors. First of all, when V_{cc} is higher than 3.2 V, the output delay of WUR, i.e., t_m, estimated by (18) is not accurate. As revealed in Fig. 14, if V_{cc} is lower than 3.2 V, the estimation error of t_m is less than 3 μs. However, when V_{cc} increases to 3.8 V, the estimation error of t_m can reach 12.4 μs. Secondly, the data sheet of LPV7215 only providers the propagation delay of the comparator at V_{cc} = 1.8 V, 2.8 V, and 5.0 V. As a result, the model in (16) cannot accurately estimate t_p between 2.8 V and 5.0 V. In our implementation, we use the propagation delay measured at V_{cc} = 2.8 V to approximate t_p at V_{cc} = 3.3 V and 3.8 V. This approximation introduces some errors in the t_p estimation, especially when V_{cc} is far from 2.8 V.

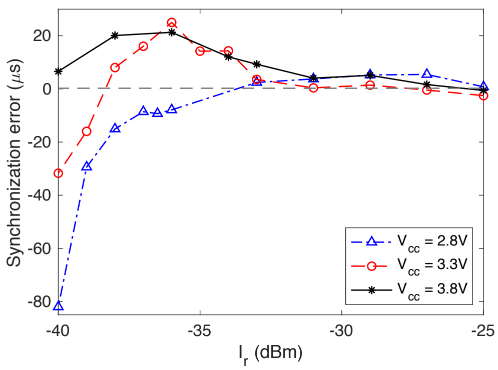


Figure 20: Calibrated synchronization accuracy of WUR-TS under different settings

By adding a variable, $34V_{cc} - 95$, to the interrupt delay estimated by our delay model, the synchronization accuracy at high power supply voltage can be significantly improved. This variable moves down the synchronization error curves in Fig. 19 according to V_{cc}. In Fig. 20, we show the calibrated synchronization accuracy of WUR-TS. After the calibration, it can be observed that when the strength of the wake-up signal is higher than -33 dBm, WUR-TS can achieve a synchronization accuracy of at least 9 μs and 4 μs at V_{cc} = 3.3 V and 3.8 V, respectively.

9.5 Performance Comparison

WUR is the key component of WUR-TS method. Therefore, we first compare the performance of our WUR built with Texas Instruments LPV7215 comparator and Powercast P1110B voltage booster with several existing WUR. The results are listed in Table 1.

The WURs shown in Table 1 are all semi-passive WURs. Compared with a fully-passive WUR, which is commonly used in RF energy harvesting without any physical power connection [17], [27], a semi-passive WUR usually contains a comparator as an active component for wake-up signal detection, thereby consuming more energy. However, the sensitivity of the semi-passive WUR is much higher (over 30 dBm) than the fully-passive WUR to support the time synchronization in a large-scale network.

Table 1: WUR Performance Comparison

	Power con (μW)	Freq (MHz)	Sens (dBm)	Interference filter
WUR-TS	2.1	915	-43.5	RC network
[23]	0.27	434	-51	OOK address
[28]	50	868	-51	OOK address
[29]	2.4	868	-71	OOK address
[30]	0.152	868	-32	OOK address
	1.2		-55	
[31]	0.0045	113.5	-69	–

As shown in Table 1, compared with other methods, the energy consumption of our WUR has no advantages. This is because we generate a static reference voltage at the inverting input of the comparator. As a result, there is a small quiescent current (around 0.7 μA at V_{cc} = 2.8 V) flowing through R_3 and R_2 in Fig. 2, which consumes 2 μW power that can be saved if an adaptive reference voltage is applied. However, allowing the reference voltage to change with the strength of the input signal will make V_+ become a function of I_r, which greatly increases the complexity of the delay model. When the strength of the wake-up signal is weak, the impact of the t_n on the interrupt delay becomes heavy. In this case, an inaccuracy delay model may greatly increase the synchronization error.

From Table 1, it can be observed that most WURs modulate the address of the intended receiver by OOK as the wake-up signal. This can significantly improve the robustness of a WUR against interferences in a complex radio environment. However, if the channel quality is bad or the distance between the communication parties is long, the length of the wake-up signal cannot be short to guarantee that the receiver can successfully decode the address. In addition, the OOK detector, e.g, pulse-width modulation decoder or a microcontroller, is essentially a digital system, which considers the envelope of OOK signal as useful information but discard the analog characteristics (e.g., the voltage and the rising time) of the wake-up signal during demodulation. This makes it difficult to estimate the interrupt delay for high-precision time synchronization.

In Table 2, we compare the performance WUR-TS with several popular time synchronization methods during a pairwise synchronization. It is worth noting that the hardware design of wireless nodes, such as the clock frequency and stability, transmission/reception delay, and communication power, can significantly affect the performance of a synchronization protocol. Therefore, running the same protocol on different platforms may obtain different synchronization accuracy and energy consumption. In order to compare the energy consumption of synchronization methods fairly, in the table, we list the number of synchronization messages that each synchronization protocol needs to send and receive. In this way, the total energy consumed by each protocol can be calculated based on the performance of the specified platform. In addition, the test conditions and the platform used by each synchronization method can refer to the references listed in the table.

As listed in Table 2, for RBS and CESP, the reference node broadcasts a beacon in each synchronization period, which requires one transmission. Both the master node and the slave node need to receive the beacon, thereby consuming two receptions. Afterward, master node in RBS sends its received beacon as timestamp to the slave consuming one more transmission and one more reception, whereas CESP exchanges coefficients consuming two more transmissions and two more receptions every synchronization period. By contrast, TPSN requires two transmissions and two receptions for each synchronization process. In WUR-TS, the center broadcasts a wake-up signal in each synchronization period. Nodes receive the wake-up signal estimate the arrival time of wake-up signal, which needs only one reception.

From Table 2, it can be realized that the energy consumption of receiver with WUR-TS is much lower than conventional methods since there is no information exchange among receivers. However, the energy consumption of the reference node running WUR-TS is very high. This is because the transmission power of the wake-up signal is much higher (> 50 dBm) than the data packet to reliably active a sleeping node. Therefore, WUR-TS needs to select a center with enough energy supply as the reference node. In addition, the synchronization accuracy of WUR-TS is affected by the strength of the received wake-up signal (I_r). If I_r is higher than -33 dBm, WUR-TS can provide a very

Table 2: Synchronization Performance Comparison

	Condition	Accuracy	Energy consumption	Topology
WUR-TS	Ir >-33 dBm	3 μs	Reference node: send 1 wake-up signal Receiver: receive 1 wake-up signal	Centralized
	Ir = -40 dBm	80 μs		
RBS [3]	–	3.7 μs	Reference node: send 1 beacon Master node: receive 1 beacon, transmit 1 packet Slave node: receive 1 beacon and 1 packet	Semidistributed
CESP [15]	MAC timestamp disabled	11 μs	Reference node: 1 beacon Mater node: receive 1 beacon, transmit 1 packet, receive 1 packet Slave node: receive 1 beacon, receive 1 packet, transmit 1 packet	Semidistributed
	MAC timestamp enabled	2.2 μs		
TPSN [3]	–	22.7 μs	Reference node: sender 1 packet, receive 1 packet Receiver: receive 1 packet, send 1 packet	Centralized

good synchronization accuracy.

It is worth noting that when testing the performance of conventional time synchronization methods, nodes never enter the sleep mode, and thus consume a lot of energy on idle listening, which is not included Table 2. Therefore, the total energy consumed by RBS, CESP, and TPSN will be much higher than WUR-TS in a real application. In addition, the conventional time synchronization methods do not wake up nodes with the WUR since nodes keep working. Therefore, the delay uncertainty caused by the WUR is not included in Table 2. In the real world, the accuracy of RBS, CESP, and TPSN may be reduced by the interrupt delay if WUR is used to save the energy.

10 Conclusions

In this paper, we proposed a new time synchronization method, called semi-passive wake-up radio receiver based time synchronization (WUR-TS), for ultra-low-power energy harvesting wireless networks. WUR-TS does not rely on the timestamp exchange mechanism, but directly utilize the wake-up signal to synchronize the time in the entire network. This is achieved by accurately estimating the arrival time of each wake-up signal. We implemented WUR-TS on a minimal number of off-the-shelf components. Experimental results show that if the strength of the received wake-up signal is higher than -33 dBm, WUR-TS can provide 3 μs, 9 μs, and 4 μs synchronization accuracy when the power supply voltage is 2.8 V, 3.3 V, and 3.8 V, respectively. When running WUR-TS, the power consumption of each EHN is only 3.2 μW when the network is idle. If the wake-up signal is detected, the EHN consumes only 3.6 μJ of energy to complete each time synchronization. WUR-TS can be used in any nodes that use the comparator to detect the wake-up signal. However, some parameters (e.g., the propagation delay and the input offset voltage of comparator) in the model may need to be recalibrated if the hardware is changed. However, once the hardware of the WUR is determined, the WUR-TS is resilient to environmental changes like temperature and channel quality.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond duty cycling: wake-up radio with selective awakenings for long-lived wireless sensing systems," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 522–530.

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

[7] M. Doudou, D. Djenouri, N. Badache, and A. Bouabdallah, "Synchronous contention-based mac protocols for delay-sensitive wireless sensor networks: A review and taxonomy," *Journal of Network and Computer Applications*, vol. 38, pp. 172–184, 2014.

[8] Z. Yuan, L. Wang, L. Shu, T. Hara, and Z. Qin, "A balanced energy consumption sleep scheduling algorithm in wireless sensor networks," in *2011 7th International Wireless Communications and Mobile Computing Conference*. IEEE, 2011, pp. 831–835.

[9] F. Liu, C.-Y. Tsui, and Y. J. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 7, pp. 2258–2267, 2010.

[10] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 147–163, 2002.

[11] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, 2003, pp. 138–149.

[12] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004, pp. 39–49.

[13] 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.

[14] 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.

[15] 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.

[16] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M. L. Wymore, and A. K. Sangaiah, "A robust time synchronization scheme for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3570–3580, 2017.

[17] 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.

[18] 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.

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

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

[21] Texas Instruments, *Low-cost low-power 2.4 GHz RF transceiver*, Texas Instruments, Inc., Dallas, TX, USA, August 2017.

[22] ——, *LPV7215 micropower, CMOS input, RRIO, 1.8-V, push-pull output comparator*, Texas Instruments, Dallas, TX, U.S.A., September 2005.

[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] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini, "Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 633–644, 2016.

[25] Texas Instruments, *Understanding Schmitt Triggers*, Texas Instruments, Dallas, TX, U.S.A., September 2011.

[26] Y. Luo and L. Pu, "Practical issues of RF energy harvest and data transmission in renewable radio energy powered IoT," *IEEE Transactions on Sustainable Computing*, 2020.

[27] A. N. Parks, A. P. Sample, Y. Zhao, and J. R. Smith, "A wireless sensing platform utilizing ambient RF energy," in *2013 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems*. IEEE, 2013, pp. 154–156.

[28] 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.

[29] 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.

[30] 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.

[31] P.-H. P. Wang, H. Jiang, L. Gao, P. Sen, Y.-H. Kim, G. M. Rebeiz, P. P. Mercier, and D. A. Hall, "A near-zero-power wake-up receiver achieving- 69-dBm sensitivity," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 6, pp. 1640–1652, 2018.