DualSync: Taming Clock Skew Variation for Synchronization in Low-Power Wireless Networks

Meng Jin[1], Tianzhang Xing[1], Xiaojiang Chen[1], Xin Meng[1], Dingyi Fang[1], Yuan He*[2]
[1]Northwest University, Xi'an, China
[2]School of Software, TNLIST, Tsinghua University
mengj@stumail.nwu.edu.cn, {xtz,xjchen,dyf}@nwu.edu.cn,
xinmnwu@gmail.com, he@greenorbs.com

***Abstract*—The low-cost crystal oscillators embedded in wireless sensor nodes are prone to be affected by their working condition, leading to undesired variation of clock skew. To preserve synchronized clocks, nodes have to undergo frequent re-synchronization to cope with the time-varying clock skew, which in turn means excessive energy consumption. In this paper, we propose DualSync, a synchronization approach for low-power wireless networks under dynamic working condition. By utilizing time-stamp exchanges and local measurement of temperature and voltage, DualSync maintains an accurate clock model to closely trace the relationship between clock skew and the influencing factors. We further incorporate an error-driven mechanism to facilitate interplay between Inter-Sync and Self-Sync, so as to preserve high synchronization accuracy while minimizing communication cost. We evaluate the performance of DualSync across various scenarios and compare it with state-of-art approaches. The experimental results illustrate the superior performance of DualSync in terms of both accuracy and energy efficiency.**

I. Introduction

Time is indispensable information in most sensor network applications. Clock Synchronization, i.e. to have a common clock among all the networked nodes, is a fundamental task in wireless sensor networks (WSNs). Accurate and efficient clock synchronization is prerequisite for many important functions of WSNs, such as localization [1], network scheduling [2], cooperative transmission [3], data fusion [4], etc.

Compared to synchronization in conventional network systems, clock synchronization in WSNs is particularly difficult. This is essentially due to two reasons: (1) the uncertainty of clock skews (clock skew generally refers to the difference in stepping speeds of clocks); (2) the resource constraints on sensor nodes. Due to the intrinsic hardware diversity on sensor nodes, clock skews become different from one node to another. As a measure to overcome clock skews, the existing approaches (e.g. RBS [5], TPSN [6], FTSP [7] and many optimizations [8–13]) mostly rely on periodical exchange of time-stamps among the nodes to synchronize nodes' clocks in a timely manner. Clearly, clock accuracy comes at the cost of frequent communication and energy consumption.

The performance of the existing approaches, however, is far from being satisfactory. The common knowledge from physics tells that the frequency of oscillators is susceptible to factors like temperature and voltage. Many sensor network systems nowadays are deployed in unattended area, where the working condition in terms of temperature and voltage are dynamic. This may cause instability of clock skews [14–16], and makes the clock skews hard to estimate. Sensor nodes thus have to undergo more frequent re-synchronization to mitigate error cumulation, which means excessive cost of energy and network bandwidth, and sometimes even hindering the normal network operations. To address the above issues, some works propose to compensate clock offsets according to a certain factor [14–16]. However, those approaches generally require prior knowledge of the exact relationship between the influencing factors and the clock skew. To obtain such knowledge means prohibitive labor cost, especially for the large-scale and long-term deployments [17].

Can we simultaneously achieve high synchronization accuracy and low cost even under dynamic working condition?

The answer is yes. Through extensive empirical studies, we have observed traceable relationship between clock skew and the influencing factors, namely temperature and voltage. That motivates us to exploit such information to accurately predict changes in clock skews and design highly efficient approaches of clock synchronization. We may meet two critical challenges towards the above goal: first, considering the hardware diversity and environmental dynamics, we need an accurate model to trace the time-varying properties of clock skews at runtime; second, given the updated offsets and skews, how to adaptively adjust the tempo of synchronization while preserving high accuracy and low cost, is still an open problem.

In this paper, we propose DualSync, a practical time synchronization approach tailored for WSNs under dynamic working condition. Unlike the existing works that merely tolerate errors and passively compensate clock skews, we propose to utilize the interfering factors (temperature and voltage) as important clues to improve synchronization accuracy and efficiency. The design of DualSync consists of two main components: *Inter-Sync* and *Self-Sync*, two alternate phases that interplay in a mutual beneficial manner. Specifically, Inter-Sync enables a node to opportunely trace the relationship between clock skew and influencing factors via time-stamp exchange. Based on the result of Inter-Sync, Self-Sync utilizes only local information for skew estimation, and thus significantly reduces the overhead of synchronization. To obtain a delicate trade-off between accuracy and cost, an *error-driven mechanism* is further proposed to adaptively adjust switching between *Inter-Sync* and *Self-Sync*. The contributions of this paper are summarized as follows

- Based on extensive experiments and observation, we

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

present a novel clock model to accurately characterize the relationship between skew variation and the influencing factors. This model is utilized to trace the variational clock skew at runtime.
- We propose the DualSync approach, in which Inter-Sync and Self-Sync interplay under the control of an error-driven adaptive mechanism. DualSync achieves guaranteed synchronization accuracy while minimizing the communication cost.
- We implement DualSync and evaluate its performance across various scenarios. The experiment results show that DualSync outperforms the state-of-art approaches in terms of accuracy and energy efficiency.

The rest of the paper is organized as follows. Section II summarizes the related work. In Section III we introduce the preliminary concepts in this work. The major design of DualSync is given in Section IV. Performance analysis of DualSync is given in Section V. Section VI shows the performance evaluation results obtained from both experimental and simulation results. Section VII concludes the paper.

II. Related Work

Early works on time synchronization in WSNs mostly focus on enhancing clock accuracy and typically rely on frequent time-stamp exchange among the nodes. Particularly, RBS [5] eliminates the delay in send and access operations by applying a receiver-receiver scheme. TPSN [6] proposes a "pair-wise" model to calculate the unknown propagation delay. FTSP [7] largely increases the synchronization accuracy by utilizing multiple MAC layer time-stamping. Although the above approaches are able to achieve high accuracy, they typically lead to overly high communication overhead and significant power drain in the system and thus are unsuitable for the energy constrained WSNs.

To address the above problems, recent works begin to address the clock uncertainty and reduce the time synchronization cost. Some of them focus on mitigating the impact of external factors. For example, in [12] and [15], nodes can wisely select the best synchronization parents, which mitigate the impact of the temperature and energy heterogeneity, and thus minimizes error propagation along the synchronization path. To eliminate the impact of temperature, EACS [14] uses a hybrid two-model system to describe the clock skew in dynamic working temperature, and uses temperature measurements to assist model selection and clock skew estimation. ODS [8] studies how synchronization interval affects the synchronization performance, and suggests adaptively adjusting the synchronization period for a trade-off between synchronization accuracy and energy consumption.

On the other hand, researchers try to study the fundamentals of clock uncertainty (i.e. clock skew) for better synchronization accuracy. Specifically, ACES [9] suggests skew tracking with complex Kalman filtering and period-level sampling adaption. In [10] the authors propose to estimate clock skew using the maximum likelihood estimators, which successfully address the unknown delay in synchronization. However, these approaches typically enmesh in an instinct that high clock accuracy is inevitably at the cost of extra energy for communication. To break such an instinct, an emerging type of solutions are proposed, which make use of certain external signal sources as common time references. Typical references include WiFi [18], FM radio [19], fluorescent lighting [20], etc. On the other hand, researchers also try to improve the synchronization performance with the assistance of the temperature or voltage measurement, such as EACS [14], EATS [15] and TACSC [21]. However, the above solutions either need specific hardware components or depend on prior knowledge, and thus are unsuitable for WSNs. Different from those above works, DualSync is not constrained by those limitations and can utilize the interfering factors as important clues to simultaneously achieve high synchronization accuracy and low cost even under dynamic working condition.

III. Preliminaries

In this section, we introduce preliminary concepts of this paper, including terminologies and initial empirical results that motivate this work.

A. Clock offset and Skew

We first define the important terms used in this paper.

Local Clock: the device used to measure time called clock. We denote the reading of clock A at time t as $C_A(t)$.

Clock offset is the difference between clock reading of a certain clock and the reference clock. We denote the clock offset of clock A at time t as $\theta_A(t) = C_A(t) - t$.

The slope of offset is defined as *clock skew*. To give an example, the clock skew of clock A at time t can be calculated as:

$$\alpha_A = \frac{d\theta_A(t)}{dt} = \frac{\theta_A(t + \zeta(t)) - \theta_A(t)}{\zeta(t)} \quad (1)$$

where ζ is the interval measured in real time. Skew is evaluated in ppm (parts per million) and normally ranges from $\pm$5ppm to $\pm$100ppm [8].

An important fact of clock skew is that it exhibits both long-term and short-term instabilities [8]. That's to say, α_A in Equation (1) is essentially a function of time $\alpha_A(t)$. In addition to fabrication and component aging, clock skew is also affected by the varying working condition such as *supply voltage* and *temperature*, which make clock skew seem to be random and hard to estimate. However, we show in the following empirical study that clock skew is actually predictable.

B. Clock Skew Characterization

To better investigate the characteristics of clock skew, we further conduct empirical study in different working condition. Here, the working condition mainly refers to the supply voltage and the environment temperature. The experimental setup is shown in Figure 1.

Voltage. In the experiment, we measure the frequency of a 32.768KHz oscillator on a MICAz under different supply voltages with a LeCroy oscilloscope, as shown in Figure 1. The nodes are powered by a Direct Current Electrical Source (DCES). The supply voltage of the node can be changed by varying the DCES. Figure 2 depicts the clock skew under different supply voltages. It can be observed that the clock skew increases as expected when the supply voltage decreases. A more important observation is *the relationship between clock skew and voltage is an approximate linear function.*

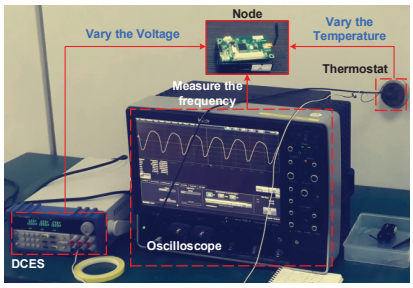

Fig. 1: The experimental equipment.

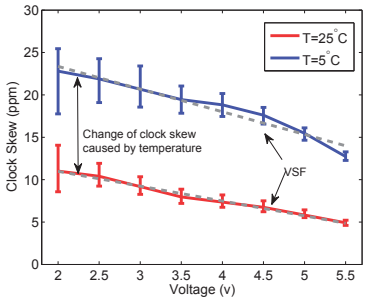

Fig. 2: Skew v.s. Voltage.

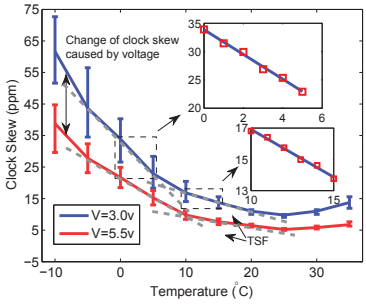

Fig. 3: Skew v.s. Temperature.

Temperature. In order to understand the impact of temperature, we put a node in a thermostat and change the environmental temperature by adjusting the thermostat. Figure 3 plots the clock skew under different temperature. We can observe that the relationship between temperature and clock skew is roughly a parabolic function, which has been verified in prior works [21]. A more interesting observation is that *the short-term relationship between clock skew and temperature is roughly linear*, as shown in the subfigure of Figure 3.

Combination of temperature and voltage. In this work, we further investigate *how the skew changes under both the impact of voltage and temperature,* which is never investigated in prior works. According to the results in Figure 2 and 3, we can conclude that: 1) The impact from voltage and temperature on clock skew can be linearly added up. 2) Skew at the temperature of 5°C is more sensitive to the voltage than that at 25°C. Similarly, skew at the voltage of 3v is more sensitive to the temperature than that at 5v.

To avoid the measurement bias on individual nodes, we further repeat the above experiments for 50 times and replace the node with a new one each time. The results of the additional 50 nodes are omitted because they follow the same trend.

Based on the above experiments, the characteristics of clock skew can be summarized as follows:

- The relationship between skew and the influencing factors is traceable, which gives us an opportunity to estimate the skew using the voltage and temperature information.
- Relying on prior knowledge of the above-mentioned relationship to estimate clock skew is not a generalizable approach.

IV. Design

In this section, we first introduce the overview of this work, and then describe the environmental impact model. After that we present the design details of DualSync.

A. Overview

DualSync is an energy efficient clock synchronization approach that delivers high clock accuracy even in dynamic working condition. Following a common practice in clock synchronization, DualSync still involves periodical time-stamp exchanges for skew and offset compensation, which we call

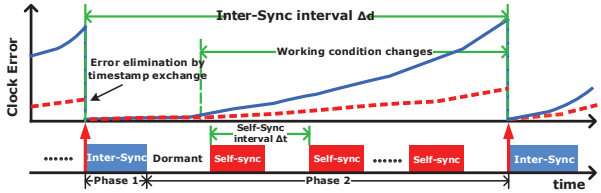

Fig. 4: Illustration of Inter-Sync and Self-Sync.

Inter-Sync. Different from the existing works, DualSync enables a node to opportunely trace the relationship between skew and influencing factors during Inter-Sync by analysing the obtained time-stamp messages and the voltage and temperature measurements. Based on the obtained relationship, DualSync is able to use the local voltage and temperature information to compensate the clock skew during the Inter-Sync intervals. We name this phase as *Self-Sync*.

We give a simplified example in Figure 4 to illustrate our idea. As depicted in Figure 4, Self-Sync is kept to compensate the impact of dynamic working condition for its low-energy profile. Inter-Sync is launched less frequently to calibrate the time error and update the relationship since it requires high communication overhead. We can see that, the error accumulation between two Inter-Sync is largely mitigated by Self-Sync, thus DualSync can achieve both high time accuracy and low energy consumption even in dynamic working condition.

However, it is easy to understand that nodes obtain the highest synchronization confidence right after each Inter-Sync. This confidence of accuracy, as shown in Figure 4 (the red dotted line), degrades gradually in the following Self-Sync phase because of the accumulated uncertainty from dynamic clock skew and the skew estimation error. To further provide a trade-off between accuracy and energy consumption, DualSync applies an *error-driven mechanism* to adaptively adjust switching between Inter-Sync and Self-Sync.

B. Clock Model

As discussed in Section III, the clock skew is not stable and prone to be affected by the working condition. Thus we model the clock skew as:

$$\alpha(t_0 + t) = \alpha(t_0) + \alpha_{VAR}(t_0 + t) \quad (2)$$

where $\alpha_{VAR}(t_0+t)$ is the change of skew during t, which is mainly related to two condition parameters, i.e., supply voltage V and temperature T [14, 15].

According to our empirical studies in Section III, the relationship between skew and voltage can be modeled as:

$$\alpha_V(t_0+t) = \alpha(t_0) + VSF(t_0+t)\cdot(V(t_0+t) - V(t_0)) \quad (3)$$

where VSF is the *Voltage-Skew Sensitivity Factor*, i.e., the change rate of the clock skew corresponding to voltage. $VSF(t_0+t)$ is denoted by a function of time because it is not stable and prone to be affected by the temperature as shown in Figure 2. We can periodically estimate and update VSF through online linear fitting in Inter-Sync.

The relationship between clock skew and temperature is usually modeled as a parabolic function as follows:

$$\alpha_T(t) = k\cdot(T(t) - T_0)^2 \quad (4)$$

where k is the temperature coefficient (typically around 0.035 ± 0.01ppm/°C) and T_0 is the turn-over temperature (typically around 25°C) [21]. According to our empirical result in Section III, k is not a constant value which is prone to be affected by the supply voltage. Thus, to keep the model updated, a straightforward method is to periodically estimate k through online quadratic fitting. Due to the limited storage resources, nodes can only store small amount of data with a relatively narrow temperature range for online quadratic fitting, which may lead to high estimation error. Moreover, according to [21], synchronization accuracy is keenly sensitive to the temperature coefficient k. Thus even a tiny error in k may result in a significant skew estimation error.

Fortunately, we find in Section III that the clock skew is roughly proportional to the temperature in a small temperature range. Suppose that temperature does not change very quickly, we can model the relationship between clock skew and temperature as a linear function and periodically update the linear coefficient through linear fitting.

$$\alpha_T(t_0+t) = \alpha(t_0) + TSF(t_0+t)\cdot(T(t_0+t) - T(t_0)) \quad (5)$$

TSF is the *Temperature-Skew Sensitivity Factor*, which is similar to VSF in Equation (3).

In this paper, we assume the impact from the two parameters is independent in a short time period. Thus, the change in clock skew α_{VAR} during t can be modeled as:

$$\begin{aligned}\alpha_{VAR}(t_0+t) =& VSF(t_0+t)\cdot(V(t_0+t) - V(t_0)) \\ &+ TSF(t_0+t)\cdot(T(t_0+t) - T(t_0))\end{aligned} \quad (6)$$

C. Inter-Sync

The objective of Inter-Sync is to eliminate the error accumulation during the Self-Sync phase based on the time-stamps. In addition, it estimates VSF and TSF for the subsequent self-calibration phase.

1) Offset and skew estimation: During the n-th Inter-Sync phase, nodes firstly exchange time-stamp messages with their neighbours. Specifically, nodes that in the broadcast radius of the reference can collect time-stamp messages directly from it. Nodes outside the broadcast radius of the reference can gather time-stamp messages indirectly through other synchronized nodes that are located closer to the reference. When a node collects enough time-stamp messages, it estimates the offset $\theta[n]$ and skew $\alpha[n]$ of its own local clock through linear regression, and becomes synchronized. According to [8], $\theta[n]$ and $\alpha[n]$ are unbiased estimators with error variance $\sigma^2_\theta[n]$ and $\sigma^2_{\alpha[n]}$.

Such time-stamp based synchronization method is widely adopted in existing approaches [7, 10], and its design detail is therefore omitted here since it is not our focus.

2) Calculating VSF and TSF: In each Inter-Sync phase, nodes note their current voltage and temperature measurement $V[n]$ and $T[n]$, and the estimated clock skew $\alpha[n]$ (we term such information as a DataPoint($V[n]$,$T[n]$,$\alpha[n]$)). $VSF[n]$ and $TSF[n]$ are further examined: linear regression is used to find the line best approximates the past W (W is the window size) DataPoints and $VSF[n]$ and $TSF[n]$ are analyzed.

Algorithm 1 gives a high level overview on how a node estimates the current TSF (the estimation of VSF is similar to that of TSF). The algorithm consists of three steps:

1) Collect a new DataPoint.
2) Store the new DataPoint in a fitting table T_{ts}.
3) Once enough DataPonits are collected in T_{ts}, calculate the $TSF[n]$ using a regression algorithm.

We assume that the $VSF[n]$ and $TSF[n]$ do not change before the next Inter-Sync phase, and thus the node can subsequently switch into the Self-Sync phase.

Algorithm 1 TSF Calculation.

Require: T_{th}: the temperature variation threshold for calculating a new TSF;
1: Obtain a DataPoint($V[n]$,$T[n]$,$\alpha[n]$);
2: **if** $\mid T[n] - \forall T_{ts-T}[w] \mid \geqslant T_{th}$ **then**
3: **if** length(T_{ts}=W) **then**
4: In T_{ts}, find the w ($1 \leqslant w \leqslant W$) that have the largest $|T_{ts-V}[w] - V[n]|$;
5: $T_{ts}[w] = (V[n], T[n], \alpha[n])$
6: **else**
7: $T_{ts}[length(T_{ts})+1] = (V[n], T[n], \alpha[n])$
8: **end if**
9: **end if**
10: **if** $length(T_{ts}) > 3$ **then**
11: $TSF[n]$=LinearRegression($T_{ts}(T, \alpha)$);
12: **end if**
13: n=n+1;

D. Self-Sync

During the Self-Sync phase, the nodes periodically measure their current supply voltage $V(t_i)$ and temperature $T(t_i)$. After every measurement, the current clock skew $\alpha_n(t_i)$ (clock skew in the i-th Self-Sync during the n-th Inter-Sync interval) can be updated by using Equation (6):

$$\begin{aligned}\alpha_n(t_i) =& \alpha[n] + VSF[n]\cdot(V(t_i) - V[n]) \\ &+ TSF[n]\cdot(T(t_i) - T[n])\end{aligned} \quad (7)$$

Obviously, the skew estimation quality is subject to the voltage and temperature measurement error. Similar to the majority of existing research efforts [21], we consider them following approximately normal distribution as $\delta_T \sim N(0, \sigma^2_T)$

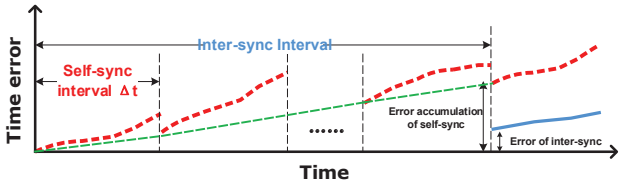

Fig. 8: Illustration of error accumulation.

and $\delta_T \sim N(0, \sigma_V^2)$. Then we can derive that Equation (7) is an unbiased estimator for clock skew with error variance as:

$$\sigma^2_{\alpha_n(t_i)} = 2 \cdot (VSF[n] \cdot \sigma_V^2 + TSF[n] \cdot \sigma_T^2), \tag{8}$$

As shown in Equation (8), the variance of the skew estimation error is highly related to VSF and TSF, which are impacted by the working condition. In other words, *the skew estimation quality is prone to be affected by the working condition.*

After the update in clock skew, the local time afterward can be predicted by using such a new clock skew till the next Self-Sync. Assuming that the clock skew does not change during the Self-Sync interval, the clock offset in the $(i+1)$-th Self-Sync can be updated as:

$$\theta_n(t_{i+1}) = \theta_n(t_i) + \Delta t \cdot \alpha_n(t_i) \tag{9}$$

where Δt is the Self-Sync interval, and a smaller Δt will obviously lead to higher synchronization accuracy. We will investigate the impact of Δt in Section V. The second term in Equation (9) represents the change in clock offset (the accumulated clock skew), which is termed as *offset drift*. If the clock offset exceeds a given threshold (such as half of the granularity of the local clock) , we conduct the compensation to the local clock as:

$$C_n(t_{i+1}) = C_n(t_{i+1}) + \theta_n(t_{i+1}) \tag{10}$$

Now the time error during the Inter-Sync interval can be eliminated using the local voltage and temperature information.

E. Error-Driven Mechanism

The proposed synchronization method faces a trade-off between Inter-Sync and Self-Sync. Particularly, Self-Sync can use local information for skew compensation, which can reduce energy consumption but suffers error accumulation. On the other hand, Inter-Sync produces a very good performance, but at the cost of a higher energy consumption for time-stamp exchange. Figure 8 gives an example of the performance of Self-Sync (dotted line) and Inter-Sync (solid line). Clearly, different lengths of the Inter-Sync interval will lead to different levels of energy consumption and error accumulation. Thus, the key to preserving high synchronization accuracy while minimizing communication cost is to repeat the Inter-Sync right on time, not too early or too late. However, as discussed in Section IV-D, the synchronization quality of Self-Sync is highly affected by the working condition, which means a *time-varying feature* of error accumulation rate during the Inter-Sync interval. This makes the problem more challenging.

To address the above problem, we suppose to apply an error-driven mechanism for adaptive Inter-Sync, which enables a node to estimate its current error accumulation and adaptively trigger the Inter-Sync based on the current error accumulation and the cost of Inter-Sync and Self-Sync as:

$$e_{self}(t_i) \cdot E_{cal} \cdot \beta > e_{inter} \cdot (E_{cal} + E_{trans}) \tag{11}$$

$$e_{self}(t_i) = e_{self}(t_{i-1}) + \Delta t^2 \cdot \sigma^2_{\alpha_n(t_i)} + \frac{\Delta t^3}{3} \cdot \sigma_\eta^2. \tag{12}$$

Where e_{self} and e_{inter} are the error uncertainty of the Self-Sync and Inter-Sync, respectively. e_{inter} can be obtained through measurements reported in previous literatures [8]. e_{self} can be calculated through integration on the error uncertainty introduced by each Self-Sync interval by using Equation (12). We will give the proof of Equation (12) in Section V. β is the error controlling factor. A larger value of β indicates a stricter accuracy requirement, and we will investigate the impact of the factor β in Section V. E_{cal} and $E_{cal} + E_{trans}$ are the energy consumption of Self-Sync and Inter-Sync, where E_{cal} and E_{trans} are the energy cost for calculation and message transmission. Equation (11) states that if e_{self} rapidly accumulates, Inter-Sync should be triggered to eliminate error. Otherwise, DualSync will resume the Self-Sync for energy efficiency.

V. Proof and Analysis

In this section, we study the synchronization performance of DualSync in terms of synchronization accuracy and energy consumption.

A. Synchronization Accuracy Analysis

1) Error Accumulation: Synchronization error is considered mainly due to the imperfect skew estimation [8, 21] which may accumulate with time. We have discussed the skew estimation quality of DualSync in Section IV-D. In this subsection, we are interested in its accumulation with time.

Theorem 1. *Equation* (9) *is an unbiased estimation of clock offset, and its error variance is:*

$$\sigma^2_{\theta_{drift}(t_i+\Delta t)} = \Delta t^2 \cdot \sigma^2_{\alpha_n(t_i)} + \frac{\Delta t^3}{3} \cdot \sigma_\eta^2 \tag{13}$$

where σ_η^2 is the step variance [8] of clock skew during the dormant interval Δt.

Proof. Given that the short-term stability of the working condition is good [8], we model the real clock skew during the i-th Self-Sync interval as

$$\alpha_n(t_i + t) = \alpha_n(t_i) + \int_{t_i}^{t_i+t} \eta(u)\, du \tag{14}$$

Where $\eta \sim N(0, \sigma_\eta^2)$. A larger value of σ_η^2 indicates worse stability of the clock skew [8]. The value of σ_η^2 can be obtained from frequency tolerance provided by oscillator specifications.

The accumulated drift offset θ_{drift} during the i-th period can be expressed as

$$\theta_{drift}(t_i + \Delta t) = \int_{t_i}^{t_i+\Delta t} \alpha_n(t)\, dt \tag{15}$$

From Equation (14) and (15), we can obtain that:

$$\theta_{drift}(t_i+\Delta t) = \Delta t \cdot \alpha_n(t_i) + \int_{t_i}^{t_i+\Delta t} \int_{t_i}^{t_i+t} \eta(u) du dt \tag{16}$$

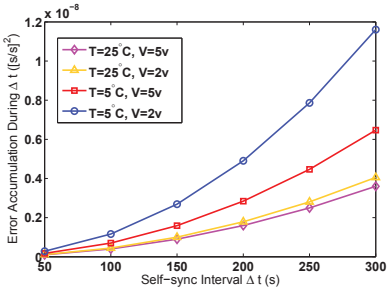

Fig. 5: σ^2_{drift} vs. Δt under different working condition.

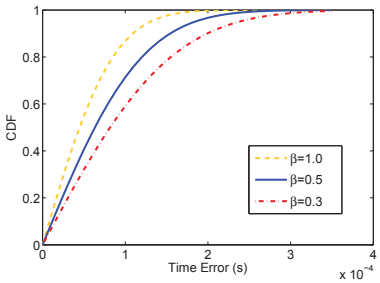

Fig. 6: Error probability vs. β.

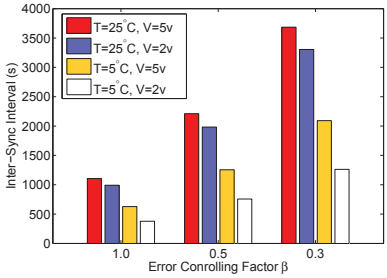

Fig. 7: Inter-Sync interval vs. β under different working condition.

From Equation (9) and (16), the error of offset obtained by Equation (9) is

$$\delta_{\theta_{drift}(t_i+\Delta t)} = \Delta t \cdot \delta_{\alpha_n(t_i)} + \int_{t_i}^{t_i+\Delta t}\int_{t_i}^{t_i+t} \eta(u)dudt \quad (17)$$

where $\delta_{\alpha_n(t_i)} = \hat{\alpha}_n(t_i) - \alpha_n(t_i)$, and $E[\delta_{\alpha(t_i)_n}] = 0$ according to [8]. Thus we have

$$E[\delta_{\theta_{drift}(t_i+\Delta t)}] = E\left[\int_{t_i}^{t_i+\Delta t}\int_{t_i}^{t_i+t} \eta(u)dudt\right] = 0$$

Namely, $\hat{\theta}_{drift}(t_i + \Delta t)$ is an unbiased estimator.

We denote $X = \int_{t_i}^{t_i+\Delta t}\int_{t_i}^{t_i+t} \eta(u)dudt$. Then the MSE (Mean Square Error) of $\delta_{\theta_{drift}(t_i+\Delta t)}$ is

$$E\left[(\delta_{\theta_{drift}(t_i+\Delta t)})^2\right] = \Delta t^2 \cdot \sigma^2_{\alpha_n(t_i)} + E[X^2]$$

because $E\left[(\delta_{\alpha_n(t_i)})^2\right] = \sigma^2_{\alpha_n(t_i)}$, $E[X] = 0$, and $\delta_{\alpha_n(t_i)}$, X are independent. It is proved in ODS [8] that $E[X^2] = \frac{\Delta t^3}{3} \cdot \sigma^2_\eta$. As a result, the error variance of the estimated drift offset in Equation (9) can be expressed as

$$\sigma^2_{\theta_{drift}(t_i+\Delta t)} = \Delta t^2 \cdot \sigma^2_{\alpha_n(t_i)} + \frac{\Delta t^3}{3} \cdot \sigma^2_\eta \quad (18)$$

This finishes the proof. □

Theorem 1 reveals that the error accumulation during one Self-Sync interval is determined by three factors: 1) stability of the clock, evaluated by σ^2_η; 2) quality of skew detection $\sigma^2_{\alpha_n(t_i)}$, which is affected by the working condition; and 3) the calibration period Δt. Figure 5 gives an example pattern of $\sigma_{\theta_{drift}}$ against Δt in different working condition (i.e. different VSF and TSF). As shown in Figure 5, the working condition acts as the deterministic factor as Δt grows. Thus, in order to control the accumulated error uncertainty σ_{drift} below $40\mu s$ in any working condition, we set $\Delta t = 100s$ in our work.

After deriving the error accumulation during Self-Sync interval, we can derive the error accumulation during the Inter-Sync interval through integration on the error uncertainty introduced by each Self-Sync interval as:

$$\sigma^2_{Inter-Sync} = \sigma^2_{\theta[n]} + \sum_{i=1}^{N}(\Delta t^2 \cdot \sigma^2_{\alpha_n(t_i)} + \frac{\Delta t^3}{3} \cdot \sigma^2_\eta) \quad (19)$$

where $\sigma^2_{\theta[n]}$ is the error of the clock offset estimation in the n-th Inter-Sync, and N is the number of Self-Sync between two consecutive Inter-Syncs.

2) Error Probability: After the analysis of the error accumulation, we give the following analysis of the error probability of DualSync.

According to Equation (11), with the error-driven mechanism, the error uncertainty during the Inter-Sync interval can be controlled as:

$$\frac{e_{inter} \cdot (E_{cal} + E_{trans})}{E_{cal} \cdot \beta} \leqslant \sigma^2 \leqslant \frac{e_{inter} \cdot (E_{cal} + E_{trans})}{E_{cal} \cdot \beta} + \sigma^2_{drift} \quad (20)$$

According to Theorem 1 the error uncertainty σ^2 can be further estimated by:

$$\begin{aligned}\sigma^2 &= \frac{e_{inter} \cdot (E_{cal} + E_{trans})}{E_{cal} \cdot \beta} + \int_0^{\Delta t}(t \cdot \sigma^2_\alpha + \frac{t^2}{3} \cdot \sigma^2_\eta) \cdot \frac{1}{t}\, dt \\ &= \frac{e_{inter} \cdot (E_{cal} + E_{trans})}{E_{cal} \cdot \beta} + \frac{\Delta t^2}{2} \cdot \sigma^2_\alpha + \frac{\Delta t^3}{9} \cdot \sigma^2_\eta\end{aligned} \quad (21)$$

Based on the probability theory, we can obtain the error probability p as:

$$p = erf(\frac{\epsilon}{\sqrt{2}\sigma}) \quad (22)$$

where erf is the Gaussian error function, and ϵ is the required accuracy. Figure 6 plots the error probability of different time error under different error controlling factor β. We can observe from Figure 6 that with a strict error requirement (e.g. β=1.0), the time error can be bounded by $200\mu s$. With a looser requirement (e.g. β=0.3), the time error increases and bounded by $300\mu s$.

B. Energy Consumption Analysis

Energy consumption of DualSync is reflected in its Inter-Sync interval. According to Equation (11), the average Inter-Sync interval can be simply derived as:

$$d = \frac{e_{inter} \cdot (E_{cal} + E_{trans})}{\beta \cdot E_{cal} \cdot \sigma^2_{drift}} \cdot \Delta t \quad (23)$$

We can easily conclude that the Inter-Sync interval is subject to two factors: 1) the error controlling factor β; 2) σ_{drift}, the error accumulation in each Self-Sync interval which is mainly subject to the working condition according to the analysis in

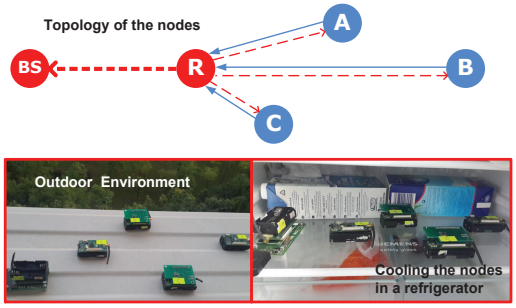

Fig. 9: Illustration of several experimental settings.

TABLE I: Default Implementation Configurations

σ_η	σ_T	σ_V	β	Δt	E_{cal}	E_{trans}
10^{-9}	0.1	0.05	0.5	100s	$0.86mJ$	$6.9mJ$

Section V-A. Figure 7 gives an example pattern of Inter-Sync interval under different β and working condition.

We can observe from Figure 7 that: 1) with DualSync, the Inter-Sync can be prolonged to 3800s, delivering a $300\mu s$ error bound; 2) DualSync can achieve a Inter-Sync interval of 1200s (the time error is bounded by $300\mu s$) even under a harsh working condition (T=5°C and V=2v).

VI. Experiment and Simulation

We implement DualSync on the MICAz mote and evaluate it with multiple experiments. To reveal its performance at scale, we also report a simulation study in this section.

A. Real-World Experiment

In the experiment, we use one node (denoted as R) as the reference node, and three other nodes (denoted as A, B and C) as the slave nodes to synchronize with node R (as shown in Figure 9). Node A runs the basic DualSync (basic DualSync is the DualSync approach with fixed Inter-Sync interval), Node B runs EATS [15] and Node C runs EACS [14]. We set the time-stamp exchange (Inter-Sync) interval to 1200s and the skew estimation (Self-Sync) interval to 100s, which is the same value as the authors of EACS [14] and EATS [15] used in their evaluation. Table I lists default configurations of DualSync. Where E_{cal}, E_{trans} and σ_η can be obtained through measurement report in previous literatures [2, 8]. σ_T and σ_V are profiled by pre-deployment measurements. Specifically, we collect more than 30000 effective samples of voltage and temperature measurements from 10 nodes, and σ_T and σ_V is equal to standard deviation of the average measured value of V and T. We perform two different experiments: *controlled experiment* and *outdoor experiment*.

Controlled experiment. We first perform a controlled experiment in different working condition which allows a detail look at what happens if the clock skew significantly changes within a short period. To investigate how temperature change impact the synchronization performance, we periodically put the nodes in a fridge (as shown in Figure 9). The initial voltages of the three slave nodes are all 3.0v, and the pre-deployment measurement for EACS and EATS to obtain the prior knowledge of the relationship between influencing factors and the clock skew is also conducted under 3.0v.

Figure 10 shows the synchronization performance of the three approaches in the third hour of the experiment (after the initialization of the VSF and TSF of DualSync). The figure tells that: 1) When the working condition is stable, all the three approaches produce small estimation errors which, however, accumulate with time, and the sudden drops of the time error are due to the local time updating in the Inter-Sync phase. 2) EATS suffers significant accuracy degradation when the temperature changes since it cannot compensate the change of temperature. 3) DualSync and EACS achieve comparable synchronization accuracy when the temperature changes.

However, EACS starts to lose accuracy after 10 hours (as show in Figure 11). We can see from the figure that EACS suffers a serious accuracy degradation when the temperature changes. This is because the relationship between temperature and clock skew may change with time, which is caused by the change in supply voltage (as discussed in Section III) and other environmental factors. However, EACS liberally use a preserved relationship for skew and offset estimation, and thus inevitably leads to high time error.

To further quantify the performance of the three approaches, we draw the CDF of time errors in Figure 12. The figure tells that: 1) EATS has a median error of $250\mu s$, and a 90th percentile of 3ms. Such high time error is due to the rapid change of temperature. 2) EACS has a median error of $200\mu s$, and a 90th percentile of $500\mu s$, $5\times$ more accurate compared to the EATS, owning to EACS's ability to (even partially) compensate the change of clock skew caused by the temperature change. 3) DualSync has a median error of $100\mu s$, outperforming EACS and EATS by $2\times$ and $2.5\times$. Its 90th percentile is $260\mu s$.

Outdoor Experiment. A controlled environment allows us to rule out other effects of inaccuracy in the synchronization process, which are not easy to obtain in an outdoor environment. Therefore, the applicability and feasibility of the proposed DualSync scheme should also be validated by outdoor experiment. In this experiment, we investigate how DualSync, EACS, and EATS handle a more variable working condition by placing Nodes A, B and C on a windowsill where both sunshine and wind are able to reach them. We observe the synchronization accuracy over a period of 18 hours.

Figure 13 shows the network synchronization error over a period of 18 hours of all the three approaches. The result tells that: 1) the error of DualSync kept less than $500\mu s$ even in a highly dynamic environment; 2) EATS suffers a clear rise in the synchronization error during day time when the temperature changes markedly; 3) EACS generally incurs a $500\mu s$ time error, and the time error rises after 12 hours when the supply voltage seriously decays.

B. Multi-hop Simulation

In the previous subsection, we have tested the effectiveness of 1-hop DualSync with MICAz mote. However, as WSN applications usually involve a large number of sensor nodes which may not be all reached by one-hop communications and thus multi-hop communication is adopted to retain a common clock throughout the whole network. To further evaluate the

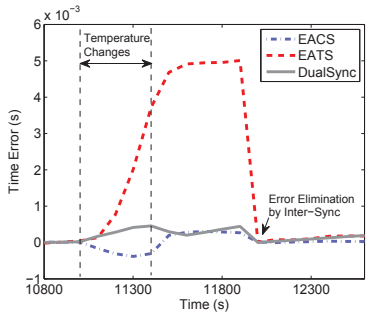

Fig. 10: Time error in the 3-rd hour.

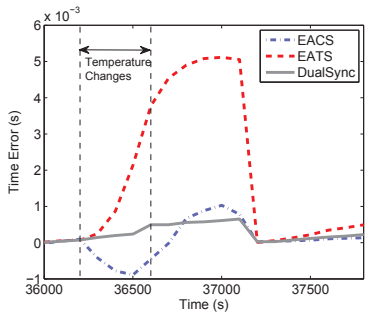

Fig. 11: Time error in the 10-th hour

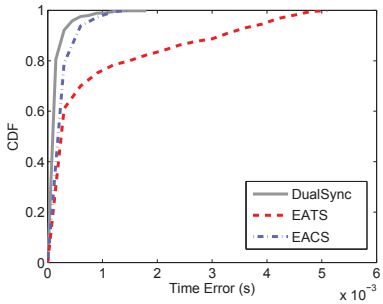

Fig. 12: CDF of time error.

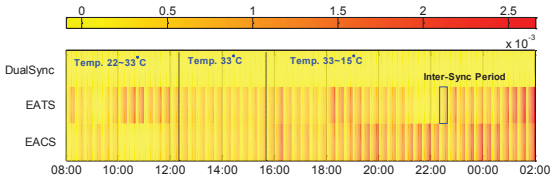

Fig. 13: Time error in an outdoor environment.

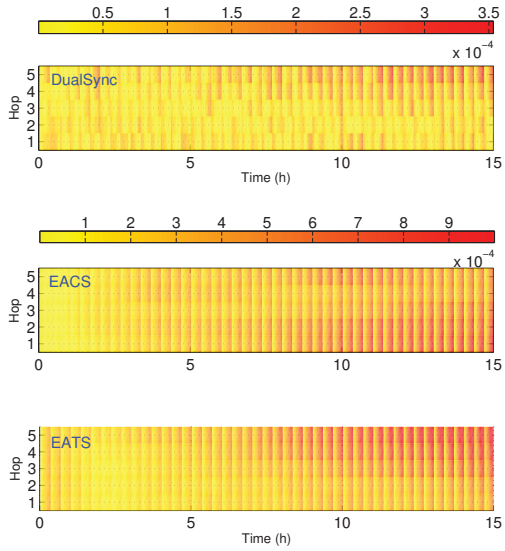

Fig. 15: Time error in Multi-hop Network.

performance of DualSync in a multi-hop network, we conduct multi-hop simulations in this section.

In the simulation, 150 nodes are randomly deployed in a 200×200 area and the reference is located at the center of the area. The energy draining rate of the battery is set to be:

$$\begin{cases} 0.00001v/1kbps, & V \geqslant 2.8v \\ 0.0001v/1kbps, & V < 2.8v \end{cases} \quad (24)$$

Figure 14 gives examples of the discharge curves for two typical nodes. To mimic a dynamic working condition, we also manually trigger the temperature variance during the simulation. Unless noted otherwise, default parameters in Table I are also used in the simulation. All statistics reported are mean values averaged over 100 runs for high confidence.

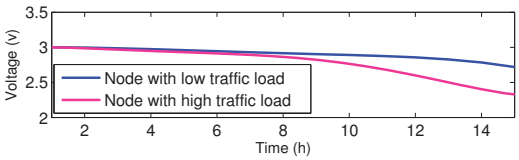

Fig. 14: Discharging curves of batteries.

Performance in Multi-hop Networks. Figure 15 shows a detail per-hop error analysis of DualSync, EATS and EACS over a period of 15 hours. As shown in the figure, the synchronization error of DualSync and EATS approaches significantly increases as the number of hops increases because more uncertainties such as delay, clock jitter, etc. are introduced. However, for EACS, the situation is exactly reversed. This interesting phenomenon is caused by the fact that the nodes close to the sink suffer a more variable voltage supply. That's because those nodes not only generate their own traffic, but also relay the traffic for other nodes to the sink, which leads to a dynamic energy consumption rate and thus fluctuates the clock skew. Unfortunately, EACS can solely compensate the impact of temperature, and thus suffers significant performance degradation when the voltage supply varies. On the other hand, we can see from this figure that the error of DualSync keeps less than $350\mu s$ even in the 5-th hop. However, the error of EATS in the 5-th hop is around 500 μs, and the maximum error achieves 1ms. According to statistic, DualSync improves the synchronization performance by $3\times$ in the 5-th hop and at least $5\times$ in the 1-st hop.

Performance with Different Inter-Sync Interval. To understand the energy efficiency of DualSync, we investigate the impact of period length on synchronization error of DualSync, and then compare it with EACS, EATS and the basic DualSync. We adjust factor β to be 1, 0.5 and 0.3. For each β, the simulation lasts 15 hours. When the experiment terminates, we calculate the average length of Inter-Sync interval to be 10min, 25min and 45min. Then, we repeat the experiment for the same duration by using EACS, EATS and basic DualSync with the calculated interval length.

Figure 16 plots the average time error observed in the experiment. We can see that: 1) error of DualSync keeps less than $300\mu s$ even when the average period length achieves 45min. 2) Compared with DualSync, EACS and EATS incurs $3\times$ and $5\times$

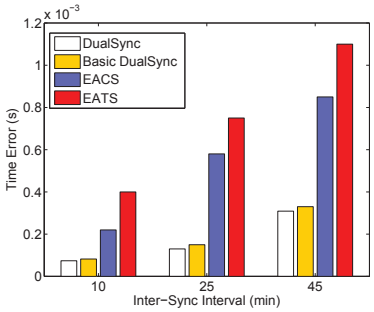

Fig. 16: Impact of Inter-Sync interval.

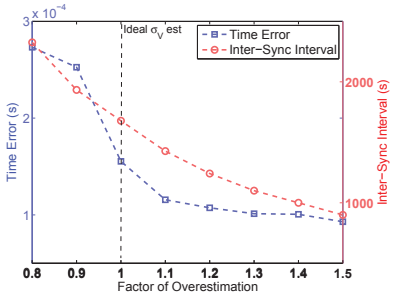

Fig. 17: Impact of σ_V estimation.

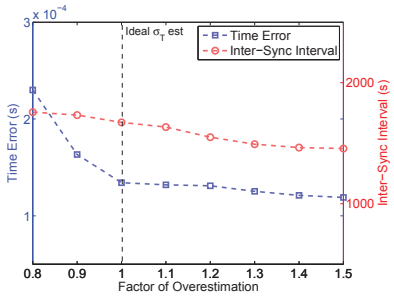

Fig. 18: Impact of σ_T estimation.

higher time error respectively, and the gap between DualSync, EACS and EATS enlarge as the average period increases. 3) Fixed Inter-Sync interval manner (basic DualSync) generally incurs a larger error than the adaptive manner.

Impact of Voltage and Temperature Measurement Quality. In this experiment, we investigate the impact of biased σ_V and σ_T estimation. Figure 17 plots mean values of the time error and the Inter-Sync interval under different factors of biased estimation, in steps of 0.1. This figure tells that: 1) with increasing overestimation, the synchronization accuracy gets improved, but the overhead enlarges linearly; 2) when underestimated, the accuracy performance degrades quickly. Curves in Figure 18 show the impact of biased σ_T estimation, which have similar trends as that in Figure 17 except that the impact of σ_T is less significant than that of σ_V. By comparing results in Figure 17 and Figure 18, we can conclude that effective estimation of σ_V plays a more important role in practical systems using DualSync.

VII. Conclusion

This paper presents DualSync, a practical design for efficient and accurate clock synchronization in low-power wireless networks under dynamic working condition. By smartly utilizing of the temperature and voltage information, DualSync makes it possible to simultaneously achieve high synchronization accuracy and low cost. The experiment and simulation results demonstrate that DualSync outperforms state-of-art approaches in terms of accuracy and energy efficiency. In our future work, we plan to take other in-network interfering factors, e.g. the instability of the parent nodes, into account and explore the synchronization issue in heterogeneous wireless networks.

Acknowledgment

This work was supported by National Basic Research Program (973 program) under Grant of 2014CB347800, The NSFC No.61170218, No.61272461, No.61202393, No.61572402, National Science Fund for Excellent Young Scientist No.61422207.

References

[1] Youngmin Kwon and Agha Gul. Passive Localization: Large Size Sensor Network Localization Based on Environmental Events. In *ACM/IEEE IPSN*, 2008.

[2] Hao Huang, Jihoon Yun, Ziguo Zhong, Songmin Kim, and Tian He. PSR: Practical Synchronous Rendezvous in Low-duty-cycle Wireless Networks. In *IEEE INFOCOM*, 2013.

[3] Yongmin Zhang, Shibo He, and Jiming Chen. Data Gathering Optimization by Dynamic Sensing and Routing in Rechargeable Sensor Networks. *IEEE/ACM Transactions on Networking*, 2016. DOI: 10.1109/TNET.2015.2425146.

[4] Hong Luo, Huixiang Tao, Huadong Ma, and Sajal K. Das. Data fusion with desired reliability in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(3):501–513, 2011.

[5] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained Network Time Synchronization Using Reference Broadcasts. In *OSDI*, 2002.

[6] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. Timing-sync Protocol for Sensor Networks. In *ACM SenSys*, 2003.

[7] Miklos Maroti, Branislav Kusy, Gyula Simon, and Akos Ledeczi. The Flooding Time Synchronization Protocol. In *ACM Sensys*, 2004.

[8] Ziguo Zhong, Pengpeng Chen, and Tian He. On-Demand Time Synchronization with Predictable Accuracy. In *IEEE INFOCOM*, 2011.

[9] Benjamin R. Hamilton, Xiaoli Ma, Qi Zhao, and Jun Xu. ACES: Adaptive Clock Estimation and Synchronization Using Kalman Filtering. In *ACM MobiCom*, 2008.

[10] Xuanyu Cao, Feng Yang, Xiaoying Gan, Jing Liu, Liang Qian, Xiaohua Tian, and Xinbing Wang. Joint Estimation of Clock Skew and Offset in Pairwise Broadcast Synchronization Mechanism. *IEEE Transactions on Communications*, 61(6):2508–2521, 2013.

[11] Jianping He, Jiming Chen Peng Cheng, and Xianghui Cao. Secure Time Synchronization in Wireless Sensor Networks: A Maximum Consensus-Based Approach. *IEEE Transactions on Parallel and Distributed Systems*, 25(4):501–513, 2014.

[12] Thomas Schmid, Zainul Charbiwala, Zafeiria Anagnostopoulou, Mani B. Srivastava, and Prabal Dutta. A Case Against Routing-Integrated Time Synchronization. In *ACM Sensys*, 2010.

[13] Xingfa Shen, Xin Qian, Bei Zhao, Qiming Fang, and Guojun Dai. Clapping and Broadcasting Synchronization in Wireless Sensor Networks. *Tsinghua Science and Technology*, 16(6):632–639, 2011.

[14] Zhe Yang, Lin Cai, Yu Liu, and Jianping Pan. Environment-Aware Clock Skew Estimation and Synchronization for Wireless Sensor Networks. In *IEEE INFOCOM*, 2012.

[15] Zhenjiang Li, Wenwei Chen, Mo Li, and Jingsheng Lei. Incorporating Energy Heterogeneity into Sensor Network Time Synchronization. *IEEE Transactions on Parallel and Distributed Systems*, 26(1):163–173, 2014.

[16] Thomas Schmid, Zainul Charbiwala, Roy Shea, and Mani B. Srivastava. Temperature Compensated Time Synchronization. *IEEE Embedded Systems Letters*, 1(2):37–41, 2009.

[17] Guillermo Barrenetxea, Francois Ingelrest, Gunnar Schaefer, and Martin Vetterli. SensorScope: Out-of-the-Box Environmental Monitoring. In *ACM/IEEE IPSN*, 2008.

[18] Tian Hao, Ruogu Zhou, Guoliang Xing, and Matt Mutka. Wizsync: Exploiting Wi-Fi Infrastructure for Clock Synchronization in Wireless Sensor Networks. In *IEEE RTSS*, 2011.

[19] Liqun Li, Guoliang Xing, Limin Sun, Wei Huangfu, Ruogu Zhou, and Hongsong Zhu. Exploiting FM Radio Data System for Adaptive Clock Calibration in Sensor Networks. In *IEEE MobiSys*, 2011.

[20] Zhenjiang Li, Wenwei Chen, Cheng Li, Mo Li, Xiang-Yang Li, and Yunhao Liu. FLIGHT: Clock Calibration and Context Recognition Using Fluorescent Lighting. In *IEEE Transactions on Mobile Computing*, 2012.

[21] Zhe Yang, Liang He, Lin Cai, and Jianping Pan. Temperature-Assisted Clock Synchronization and Self-Calibration for Sensor Networks. *IEEE Transactions on Wireless Communications*, 13(6):3419–3429, 2014.