

Environment-Aware Clock Skew Estimation and Synchronization for Wireless Sensor Networks

Zhe Yang*, Lin Cai*, Yu Liu⁺, and Jianping Pan*

*University of Victoria, BC, Canada, ⁺University of New Orleans, LA, USA

Abstract—Clock synchronization is a fundamental requirement for network systems. It is particularly crucial and challenging in wireless sensor networks (WSNs), because WSN environments are dynamic and unpredictable. To tackle this problem, how to accurately estimate clock skew, the inherent reason causing clock desynchronization, is investigated. According to the measurement results, clock skew is a *non-stationary* random process highly correlated to temperature, and its measurements contain severe noises. Based on the observation, an additional information aided multi-model Kalman filter (AMKF) algorithm is proposed, which uses temperature measurements to assist clock skew estimation. Using AMKF, an environment-aware clock synchronization (EACS) scheme is proposed to dynamically compensate clock skew. The scheme is simple, scalable, and of low computation and energy cost. Using EACS as an additional component of the conventional synchronization protocols, the clock is updated with local information before the clock re-synchronization process is triggered, so it can substantially prolong the re-synchronization period, which not only reduces the energy consumption but also is essential for the scenarios where frequent synchronization is infeasible. The theoretical lower bound of clock estimation error is derived as a benchmark. Extensive simulation and test (by implementing EACS in a sensor testbed) results have demonstrated the feasibility and effectiveness of the proposed scheme which can prolong the time resynchronization period by an order of magnitude in dynamic environments and its performance is close to the theoretical bound.

I. INTRODUCTION

Low-cost, low-power and miniaturized sensor nodes can construct wireless sensor networks (WSNs) supporting many applications, such as smart grid, environment monitoring, and wild animal tracking [1]–[4]. Many WSN functions require all nodes have a common notion of *time* to facilitate data transmission, localization, sleep and transmission scheduling, information fusion, etc. However, the output value of a node’s local clock, *local time*, is often different from that of another node, leading to clock offset. An unbounded clock offset will degrade the network performance and even endanger the proper functioning of WSNs.

Therefore, clock synchronization, the process of mitigating the clock offset between different nodes to obtain a common notion of time, is a fundamental requirement of WSNs. It is also a very challenging issue. First, sensor nodes are deployed in dynamic and even hostile environments, so the clock synchronization should be adaptive to the changing environment. Second, sensor nodes have very limited computation capacity and power supply. Third, a large number of sensor nodes may be needed to fulfill a given task due to the limited coverage of a single node, so scalability is another important requirement.

Clock skew is the inherent and dominant reason causing

clock desynchronization [5]. It is the instantaneous clock drift rate between two or more clocks, i.e., the difference of ticks from different clocks. The output value of a clock is the accumulated number of its ticks and therefore the effect of clock skew can be accumulated. The accumulation of clock skew will lead to an unbounded clock offset, so clock resynchronization by exchanging timestamps must be conducted after a certain period, depending on the required synchronization accuracy. The re-synchronization protocol design has been heavily investigated in the literature [6]–[9]. Here, we focus on clock skew estimation and compensation which can be used to further enhance the performance of the existing protocols [6], [8], [10].

How to accurately estimate clock skew and compensate it is non-trivial [11]. Previously, a constant clock skew was assumed for skew estimation, which is not accurate according to the measurement results. Kalman filters have been used to improve online skew estimation performance, which can achieve the minimum mean square error (MSE) for linear systems with white Gaussian noise [12]–[15]. However, in WSNs, clock skew is non-stationary due to environment variations. Different from the previous approaches, in this paper, we use the correlation of temperature and clock skew in a multi-model Kalman filter to accurately estimate the clock skew in a dynamic environment. Here, the dynamics of the environment mainly refers to the changing environment temperature.

The main contributions of this paper are threefold. First, based on our measurements, we demonstrate that the clock skew is highly correlated to temperature, so we use a hybrid two-model system to describe the clock skew. For the hybrid system, the main challenge is to determine which model should be used in the Kalman filter for a specific time period, as clock skew measurements contain severe noise. To tackle this problem, we propose an additional information aided multi-model Kalman filter (AMKF) to obtain the model likelihood for clock skew based on the local temperature model likelihood estimation. Second, using the clock skew and temperature estimation outcomes, we propose an environment-aware clock synchronization (EACS) scheme to dynamically compensate the clock skew. The proposed solution can substantially improve the clock skew estimation accuracy and prolong the resynchronization period. Besides, the proposed EACS scheme relies on local information only and therefore it is simple and scalable. Third, we obtain the Cramer-Rao lower bound for the clock skew estimation. The theoretical bound is useful to determine an upper bound for the local clock skew compensation period. Finally, simulation and experimental

results using our sensor testbed are presented to demonstrate the superior performance of the proposed EACS scheme, which can reduce the resynchronization frequency (and thus the energy cost) by more than an order of magnitude, and its performance is close to the theoretical bound.

The rest of the paper is organized as follows. Section II introduces the related work, and the important notations are summarized in Sec. III. The clock skew measurement results are presented in Sec. IV. In Sec. V, the clock skew models are discussed and formulated based on the measurement results. The environment-aware clock skew compensation protocol is given in Sec. VI. The performance lower bound for clock skew estimation and performance evaluation by trace-driven simulation and experiment are given in Sec. VII, followed by concluding remarks and further research issues in Sec. VIII.

II. RELATED WORK

Clock synchronization protocol design has attracted extensive research work and many approaches have been proposed for different scenarios. For example, the Network Time Protocol (NTP) [6] has been widely used in the Internet, which allows computers to extract the timestamp information from NTP packets to update their local clocks. For multi-hop wireless networks, the Flooding Time Synchronization Protocol (FTSP) [8] has been proposed. It calculates the clock skew in a moving window and uses linear-regression algorithms to mitigate both clock offset and clock drift in a hierarchical way. Clock skew is the instantaneous clock drift rate caused by imperfect oscillators and/or Phase Locked Loop (PLL) modules, which results in the different durations of clock ticks. FTSP takes the clock skew into account, therefore, it can achieve a relatively high synchronization accuracy.

With the estimation of clock skew, clock offset can be compensated to prolong the resynchronization period. Recently, several clock skew estimation algorithms have been proposed. A linear optimization problem was formulated and solved by linear programming in [16]. A convex hull algorithm for offline clock skew estimation was proposed in [17], and this algorithm was also used for real-time clock skew estimation and mitigation. In [18], how to jointly estimate clock offset and clock skew with unknown synchronization delay was addressed. In [19], a directed clock skew removal technique was proposed. However, the relationship between clock skew variation and working environment has not been considered in such work, and the skew estimation using a moving window might not be able to track the change of clock skew accurately.

In general temperature is a key parameter affects the operation of a clock as the crystal oscillator and/or clock circuit are sensitive to temperature. Although the temperature compensated crystal oscillator (TCXO) does exist, e.g., MAXIM DS32KHz, the cost is still too high (\$2.81@1K [20]) for some sensor nodes, and the TCXO also increases the complexity, energy consumption and footprint of wireless devices. In addition, even with a TCXO, the clock skew is still non-negligible (e.g., up to ± 7.5 ppm [20]). Thus, we are motivated to devise low-cost, flexible software solutions.

In [21], a temperature-compensated time synchronization algorithm was proposed to compensate the clock according to the environment temperature and can achieve a much longer resynchronization period. The limitation of that work is that the calibration algorithm treated the clock skew as a stationary random process, which may not be true in dynamic environments and will thus degrade the estimation accuracy.

On the other hand, it is well known that Kalman filter is optimal for linear systems with white Gaussian noise in terms of MSE. Therefore, it is a good choice for real-time clock skew estimation and several approaches have been proposed based on Kalman filter [12]–[15]. The algorithms proposed in [12], [13] assumed that the clock skew is constant, which works well in a relatively stable environment, while the performance will degrade in dynamic environments. For example, the temperature may vary more than 20°C within 12 hours in an outdoor environment. Temperature changes may be more dramatic for some applications such as wild fire detection. The circuit temperature changes can also be faster due to the heat generated by the circuit components.

In [14], an $AR(1)$ model was developed to describe the changing clock skew, which provides some adaptivity w.r.t the changing clock skew. In [15], an interacting multi-model (IMM) Kalman filter was employed in clock skew estimation to consider the changing clock skew. However, due to the high noise level of clock skew measurement, the model uncertainty problem still exists. In addition, how to utilize the high correlation between temperature and clock skew to better estimate clock skew has not been investigated in [14], [15].

Different from the previous approaches, in this work, we first identify the correlation between temperature and clock skew from the measurement results, and propose to accurately estimate clock skew with the assistance of temperature measurements using a low-cost Kalman filter. Based on the estimations, we use the correlation between temperature and clock skew to compensate the instantaneous clock skew, so it can maintain the required clock synchronization level for a much longer duration without exchanging timestamp messages. Thus, much less energy consumption is needed for sensor nodes to exchange timestamp messages over the air. The proposed solution can be easily extended to any other wireless devices equipped with temperature sensors.

III. NOTATIONS

For presentation clarity, we first define the important terms used in this paper. The notations are consistent with the previous work [15].

Clock is the device to measure time. Generally speaking, it consists of a periodic component that ticks at a given frequency and a counter that counts the number of ticks from a starting point. Denote by $C_A(t)$ the output value of clock A at a given reference time instant t .

Clock offset is the difference between the time reported by two or more clocks. If the time at clock A and at clock B are $C_A(t)$ and $C_B(t)$, respectively, the offset of clock A relative to clock B can be written as $\theta_{A,B}(t) = C_A(t) - C_B(t)$.

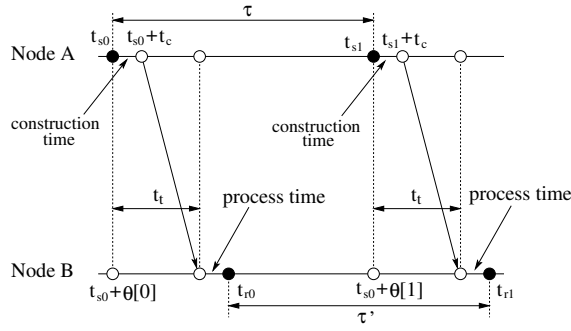


Fig. 1. Synchronization between two nodes.

Clock skew is the differential coefficient of the clock offset. The clock skew reflects the tick duration difference of two or more clocks, and is denoted by

$$\alpha_{A,B}(t) = \frac{d\theta_{A,B}(t)}{dt} \approx \frac{\theta_{A,B}(t + \tau(t)) - \theta_{A,B}(t)}{\tau(t)}, \quad (1)$$

where $\tau(t)$ is the sampling interval.

Using a uniform sampling where $\tau(t) = \tau$, the clock offset and clock skew can be discretized, respectively, as

$$\theta_{A,B}[n] = C_A[n] - C_B[n], \quad (2)$$

$$\text{and } \alpha_{A,B}[n] = (\theta_{A,B}[n+1] - \theta_{A,B}[n]) / \tau. \quad (3)$$

There are several sources that may cause errors in synchronization when exchanging timestamps. Denote by $D[n]$ the observed time difference between two remote clocks from the n -th sample, and it is expressed as

$$D[n] = \theta[n] + t_c[n] + t_t[n] + t_p[n], \quad (4)$$

where $\theta[n]$ denotes the inherent clock offset, $t_c[n]$ and $t_p[n]$ are the message construction delay at the sender side and the processing delay at the receiver side, respectively, and $t_t[n]$ is the propagation delay. $t_c[n]$ and $t_p[n]$ may be different from system to system due to the CPU capacity and the system load. Their variations are usually at μs level or even smaller, by setting the corresponding processes at a high priority. Hence, we can approximate them as constants and their variations can be included in the system observation noise. $t_t[n]$ can also be approximated as a constant if the channel is stable and not busy, thus its variance can be included in the system observation noise as well. Therefore, we can rewrite the observed time difference as

$$D[n] = \theta[n] + t_c + t_t + t_p + \omega'[n], \quad (5)$$

where $\omega'[n]$ is the observation noise for the n -th sample.

Considering a simple synchronization case with two nodes, let a node broadcast one-way messages to the other node periodically to obtain the clock skew measurements, which is part of the typical two-way message exchange for conventional synchronization protocols. As shown in Fig. 1, node A periodically sends timestamp messages to node B . Based on the timing relationship, we can get the observed time difference from the first timestamp message as $D[0] = t_{r0} - t_{s0}$, where $t_{s0}[n]$ and $t_{r0}[n]$ are the time instants of sending and receiving

the first timestamp, respectively. Hence, the clock offset can be expressed as $\theta[0] = D[0] - t_c - t_t - t_p - \omega'[0]$.

Since we do not know the values of t_c , t_t and t_p , we need another observed time difference $D[1]$ to obtain the clock skew measurement. Similar to $D[0]$ and $\theta[0]$, we can get $D[1]$ and $\theta[1]$, respectively, as $D[1] = t_{r1} - t_{s1}$ and $\theta[1] = D[1] - t_c - t_t - t_p - \omega'[1]$.

The observed clock skew then can be expressed as

$$\alpha[0] = \theta[1] - \theta[0] = D[1] - D[0] + \omega[0], \quad (6)$$

where $\omega[0] = \omega'[0] - \omega'[1]$ represents the observation errors caused by the approximations mentioned above, which is assumed to be white Gaussian noise with variance σ_ω . In this manner, we can obtain a series of clock skew measurements $\{\alpha[0], \alpha[1], \alpha[2], \dots, \alpha[n]\}$.

IV. CLOCK SKEW MEASUREMENTS

To accurately estimate clock skew, we first investigate its property from measurement results. As all oscillators and PLL modules are sensitive to temperature, and sensor nodes need to work in dynamic environments, in addition to the existing measurement results [15], [22]–[24], we conduct further measurements over a time span of six months to verify and quantify the correlation of temperature and clock skew.

A. Measurement Setting

We used one laptop as the reference node and the widely used sensor node, Berkeley Mica2 Mote [1], as the client to send timestamps periodically, where the period duration is set to one second for simplicity. We used two Mica2 nodes and put them together. One Mica2 node was connected to the MIB board [25], which offers the serial interface to send timestamp messages to the laptop. To avoid the buffer jitter, when a serial port event interrupt is triggered, the laptop will read the system time right away and then process the timestamp messages. Using the wired link to send the messages can minimize the noise introduced by the transmission delay, so the measured clock skews can be more accurate. The second node was connected to a sensor board MTK300 [26], which has a temperature sensor. The reason we used two nodes here is that the sensor board MTK300 is too thick for Mica2 to connect with the MIB board.

To obtain the clock skew measurement in a dynamic environment, we used a heater to increase and a fan to reduce the environment temperature for the Mica2 sensor nodes. This can mimic the outdoor situation where the sunshine and wind can increase and reduce the temperature. The Mica2 node equipped with the temperature sensor sent the temperature measurement to the other node on demand. The Mica2 node connected to the MIB board periodically transmitted timestamp messages containing the temperature information to the laptop. Hence, we can use these messages to obtain the clock skew measurement, as shown in Fig. 1.

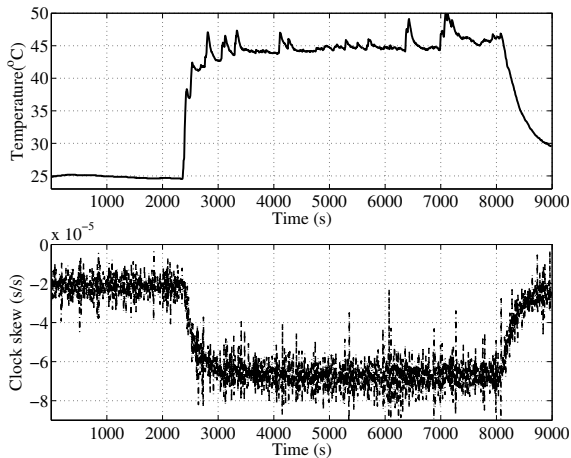


Fig. 2. Experiments on temperature and clock skew.

B. Measurement Results and Analysis

The measurement results are shown in Fig. 2. From the figure, when the environment is stable, the clock skew can be well modeled as a constant with additive noise. On the other hand, the absolute value of clock skew increases (decreases) as the temperature increases (decreases) and hence cannot be considered as a stationary random process, since its mean value is time-varying. Therefore, it is difficult to use a single model to describe the dynamic behavior of clock skew [15], and we use a hybrid two-model system to describe the clock skew as

$$\begin{aligned} \alpha[n] &= \alpha + \phi[n] & (a), \\ \alpha[n] &= \alpha[n-1] + \tau\rho + \phi[n] & (b), \end{aligned} \quad (7)$$

where ρ is the changing rate of clock skew and τ is the sampling interval and $\phi[n]$ represents the processing noise, which can be considered as crystal jitter caused by electromagnetic interference and/or some other uncertain factors.

If the temperature is relatively stable, the change of clock skew is marginal and the constant clock skew model in (7-a) should be employed. If the temperature changes, we can use the constant velocity model shown in (7-b) to characterize the change of clock skew, because the temperature change is relatively smooth during the sample period and the constant velocity model has been widely used in kinematics.

However, given the severe noise level of clock skew measurements, it is very difficult if not impossible to determine which model to use at a time instant. In addition, from the measurement results, we note that the change of clock skew is not necessarily linear, and sometimes it follows a quadratic curve. However, we avoid to introduce other models, such as constant acceleration model, as more models will further exaggerate the model uncertainty problem. In addition, the constant velocity model has the ability to keep up with the constant acceleration process if the sampling rate is sufficiently high, and in this case, the model imperfection can be treated as part of the processing noise.

A more important observation is that, in Fig. 2, the trend of the temperature curve matches the clock skew curve very

well. We repeated the experiments, and the results all show the similar tendency. To reveal the correlation, we first used the raw measurement data to calculate the Pearson product-moment correlation coefficient ($\rho_{X,Y} = E[(X - u_X)(Y - u_Y)] / (\sigma_X \sigma_Y)$) of the clock skew and temperature measurements. The coefficient result is -0.505 only, as the clock skew and temperature measurements both contain noises. We then used a moving window of size 5 to smooth the measurement results. We used the average of the measurement results within the window and calculated their correlation coefficient again, and the result is -0.973 , indicating that the temperature and clock skew are indeed highly correlated (negative temperature coefficient). However, we need to use advanced estimation techniques to use this correlation, as the raw measurements, particularly for clock skew, contain severe noise.

Furthermore, from our measurements repeated over a half-year span, for the same pair of clocks, the relationship of clock skew w.r.t. temperature is stable; but different pairs of clocks exhibit different clock skew change patterns w.r.t. temperature. With this observation, we propose to dynamically compensate the clock skew for each sensor node by first estimating the relationship between the clock skew (w.r.t. the reference clock) and temperature, which can enhance the effectiveness of the clock offset compensation process and prolong the resynchronization period.

Also, the impact of temperature on clock skew is dominant when compared with other environment factors, such as humidity, shock etc. [24], [27], which have not been studied in this paper, and we treat them as system noise.

V. AMKF CLOCK SKEW ESTIMATION

According to the measurement results, clock skew is dependent on and sensitive to the temperature and is modeled as a hybrid system with two models. It cannot be considered as a stationary random process and many estimators designed for stationary random processes are not applicable here, such as Maximum-likelihood, Wiener filter, etc. Here, we employ the Kalman filter for clock skew estimation.

Considering the hybrid system of clock skew models, it is difficult to determine the suitable model for a specific time, because the switching of models is random and unpredictable. In our previous work [15], an IMM Kalman filter is used to tackle this model uncertainty in clock skew, which can combine the output of several filters based on different models and/or parameters. Instead of making explicit decisions, the IMM Kalman filter uses the weighted sum of the outputs from different filters as the system output. The output combination is based on a soft decision by assigning different probabilities to different models. The transition between these models is considered as a Markov chain with preset transition probabilities. In each iteration, every model processes the measurement data independently and the likelihood function is calculated based on the estimation of each filter. Then we can obtain the dynamic weights, which can be considered as the posterior probability of each filter based on these likelihood functions.

Although the IMM Kalman filter has the potential to deal with model uncertainty, it is still difficult to accurately estimate the model probability because the measurement noise is relatively large. The measurement noise may lead to severe errors in selecting the clock skew models, so the previous solution in [15] is not robust and accurate enough.

On the other hand, we note that the measurement noise for temperature is much smaller than that of clock skew as there are more additive error sources for clock skew measurement. Given the high correlation of temperature and clock skew, we can use the temperature information to assist the decision of which model to use for clock skew estimation.

Therefore, we propose the additional information aided multi-model Kalman filter (AMKF), which can estimate the model probability for one process using the model probability of another related process of which the model probability is easier to obtain. This approach is different from the traditional approaches, where the decision is based on the estimated process itself only.

To use AMKF, we need to know the model probability of the related process, which is temperature in this work. Given the fact that the temperature measurements contain relatively small noise and the environment temperature can be either stable or slowly changing, temperature is also modeled as a hybrid two-model system with a constant model and a constant velocity model. We use the IMM Kalman filter [28] to process the temperature first and obtain the corresponding model probability. At any time instant, the model for the temperature is assumed identical to the model for the clock skew because these two processes are highly positive-correlated.

The structure of the IMM Kalman filter is shown in Fig. 3. The IMM Kalman filter takes three steps: reinitialization, model conditioned filtering, and output estimation. The first step is to reinitialize the estimation state $\hat{x}_{k-1|k-1}^{(i)}$ and covariance matrix $\tilde{P}_{k-1|k-1}^{(i)}$ based on the estimation results $\hat{x}_{k-1|k-1}^{(i)}$ and the corresponding covariance matrix $\hat{P}_{k-1|k-1}^{(i)}$ of all filters in the system. In the second step, we maintain several standard Kalman filters based on different models independently to process the measurements z_k . At last, the output $\hat{x}_{k|k}$ is estimated based on the estimation results from different filters and the corresponding model likelihoods Λ_k^i , which is calculated based on the estimation residuals and corresponding covariance matrix.

Once we obtain the model probability from the IMM filter for temperature, we can set a threshold to determine which model to use for clock skew estimation. In our work, the threshold may not be 0.5. This is because the constant-velocity model is a higher order model when compared with the constant model. Therefore, the estimation error caused by the inaccurate model selection of using the constant velocity model for a constant process is smaller than vice versa. Hence, the constant-velocity model is slightly preferable, so it is associated with a larger likelihood. Since we largely eliminate the model uncertainty of clock skew, we can use the sequential Kalman filter here with different state transition matrix and

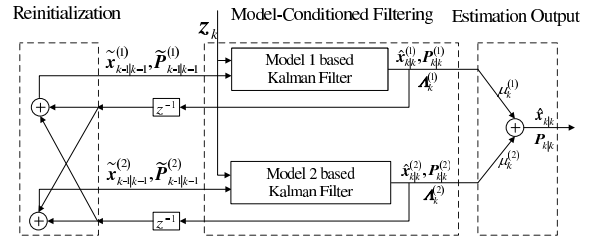


Fig. 3. Structure of the IMM Kalman Filter.

covariance matrix of the processing noise. The mathematical details of IMM [29] and Kalman filter [30] are not presented here due to the space limitation.

The computational complexity of Kalman filter is mainly caused by the matrix inversion. As it comes to our problem, the matrices involved are all 2×2 and therefore the inversion can be calculated efficiently, so the computational complexity of the proposed solution is affordable even for sensor nodes.

VI. ENVIRONMENT-AWARE CLOCK SYNCHRONIZATION

An effective compensation of clock skew can prolong the clock re-synchronization period and thus less frequent timestamp exchange is needed, which is particularly desirable for WSNs where communication energy is costly and frequent re-synchronization may even not be feasible due to mobility, sleeping schedule, etc. As discussed in the previous section, clock skew is highly correlated to temperature and sensor nodes are typically deployed in dynamic environments, we propose the environment-aware clock synchronization (EACS) scheme to dynamically compensate clock skew.

For dynamic clock compensation, we need to build a clock skew table containing clock skew w.r.t. temperature for each node. This can be conducted online or offline. The proposed AMKF estimator can be used to obtain such a table stored in each sensor node. The table size is moderate, varying from a few hundred to kilo bytes depending on the temperature range and the required granularity.

During the operating phase, for sensor node i , a local variable $\theta_{(i)}[n]$ is used to indicate the clock offset in the n -th time slot, which is the accumulated clock skew since the previous synchronization. Due to the thermal inertia, temperature does not change very quickly if the temperature sampling rate is high enough, such as in a few seconds and therefore we assume that clock skew does not change during a sampling period. Thus, the instantaneous temperature estimation can represent that for the whole sampling period. We can use the current temperature and a linear approximation to obtain the instantaneous clock skew $\alpha_{(i)}[n]$ based on the entries of the clock skew table. Then we can update the clock offset as

$$\theta_{(i)}[n] = \theta_{(i)}[n-1] + \tau \alpha_{(i)}[n]. \quad (8)$$

If the current temperature is beyond the temperature range of the clock skew table, a resynchronization is triggered to keep up with the reference clock and estimate the clock skew accordingly. Then we can update the clock skew table with new entries for the out-of-range temperature. If the clock offset

exceeds a given threshold (such as half of the granularity of the local clock), we compensate it to the local clock as follows:

$$C_i[n] = C_i[n] - \theta_i[n]. \quad (9)$$

The EACS scheme is described in Algorithm 1.

Algorithm 1 Environment-aware clock synchronization

Input \mathcal{T}_{cs} : clock skew table with respect to temperature;
 T_h, T_l : the highest and lowest temperature in \mathcal{T}_{cs} , respectively;
N: default, course-grained synchronization period;
u: local clock tick unit;

Start:
initialize the clock offset $\theta[0] = 0$;
for $k = 1$ to N (time slot) **do**
 obtain the current working temperature t_k
 if $t_k > T_h$ or $t_k < T_l$ **then**
 if reference clock is available **then**
 conduct resynchronization and update clock skew table
 else
 select clock skew corresponding to T_h or T_l
 end if
 end if
 in \mathcal{T}_{cs} , find the largest $t_u \leq t_k$ and the smallest $t_l \geq t_k$ and the corresponding skew α_u and α_l
 if $t_u == t_l$ **then**
 $\alpha_k = \alpha_u = \alpha_l$
 else
 $\alpha_k = \frac{(\alpha_u - \alpha_l)}{t_u - t_l}(t_k - t_l) + \alpha_l$ // linear approximation
 end if
 $\theta[k] = \theta[k-1] + \tau\alpha_k$
 if $\theta[k] \geq \frac{u}{2}$ **then**
 $C[k] = C[k] - \theta[k]$ // compensate offset to clock $C[k]$
 $\theta[k] = 0$
 end if
end for
conduct resynchronization if reference clock is available
go to **Start**

Given the unavoidable clock skew estimation errors and clock jitters, the clock will eventually drift away but at a much slower pace. Given a clock offset upper bound, resynchronization is needed after a certain period using existing protocols such as FTSP, which is beyond the scope of this paper.

Discussions: To address the challenges for clock synchronization in WSNs, first, EACS is designed for dynamic environments. Second, for the software-based EACS, the computational cost of the clock compensation is very low, which makes it feasible for those platforms with limited computation capacity. Third, as the clock skew is compensated appropriately, the synchronization period can be substantially prolonged. Given that the energy consumption due to wireless communications is significant, using the dynamic clock compensation, the overall energy consumption using EACS is much less. The proposed scheme can also benefit the scenarios where frequent clock synchronization is not feasible. Finally, as EACS does not require message exchanges during the clock maintenance stage, it is also scalable, even for multi-hop networks. The clock compensation accuracy for nodes at different hops away from the reference node is similar because the compensation relies on the local information only.

VII. PERFORMANCE STUDY

In this section, we first present the Posterior Cramer-Rao Lower Bound (PCRLB) of skew estimation error, which can be used as a performance benchmark. We next evaluate the performance of the proposed EACS scheme by simulation with the parameters obtained by the measurements, and then by experiments using our sensor testbed.

A. Error Estimation Bound

With clock skew compensation, the clock offset is mainly due to the imperfect skew estimation, so the skew estimation error bound determines the synchronization period and provides important insights for clock synchronization protocol design. The estimation covariance matrix can be used to indicate the estimation accuracy, which is bounded by the Posterior Cramer-Rao Lower Bound (PCRLB). In this section, we derive the PCRLB for clock skew estimation, which can be used as a benchmark for performance evaluation.

Denote the state to be estimated and the measurement at time instance k as $x_k \in R^n$ and $z_k \in R^m$, respectively. The posterior Fisher information matrix (FIM) for X_k is defined as

$$J(X_k) \triangleq -E \left[\frac{\partial^2 \log p(X_k, Z_k)}{\partial X_k \partial X_k'} \right] \quad (10)$$

where $X_k = [x_1^T, x_2^T \dots x_k^T]^T$, $Z_k = [z_1^T, z_2^T \dots z_k^T]^T$ and $p(X_k, Z_k)$ are the joint probability density function (PDF) of X_k and Z_k , respectively. Denote the estimation of X_k as \hat{X}_k . The estimation covariance is bounded by the PCRLB as

$$P_{X_k} = E[(\hat{X}_k - X_k)(\hat{X}_k - X_k)'] \geq \Phi_k \triangleq J_k^{-1}(X_k), \quad (11)$$

with the equality holds when $\hat{X}_k = E(X_k|Z_k)$. Since $X_k = [(X_{k-1})^T, x_k^T]^T$, Tichavsky et al. showed that the posterior FIM $J(X_k)$ can be decomposed into block matrix [31]:

$$J(X_k) = \begin{bmatrix} J_{X_{k-1}, X_{k-1}} & J_{X_{k-1}, x_k} \\ J_{X_{k-1}, x_k}^T & J_{x_k, x_k} \end{bmatrix} = \begin{bmatrix} J_k^{11} & J_k^{12} \\ J_k^{21} & J_k^{22} \end{bmatrix}. \quad (12)$$

Similarly, P_{X_k} can also be decomposed into block matrix

$$P_{X_k} = E \begin{bmatrix} P_{X_{k-1}} & (\hat{X}_{k-1} - X_{k-1})(\hat{x}_k - x_k)^T \\ (\hat{X}_{k-1} - X_{k-1})(\hat{x}_k - x_k)^T & P_{x_k} \end{bmatrix}, \quad (13)$$

where $E[P_{x_k}]$ is the estimation covariance of x_k . As $P_{X_k} - \Phi_k$ is semi-definite, the corresponding PCRLB is

$$P_{x_k} = E[(\hat{x}_k - x_k)(\hat{x}_k - x_k)^T] \geq \phi_k \quad (14)$$

where $\phi_k = (J_k^{22} - (J_k^{12})^T (J_k^{11})^{-1} J_k^{12})^{-1}$. The corresponding posterior FIM for x_k is

$$J(x_k) = J_k^{22} - (J_k^{12})^T (J_k^{11})^{-1} J_k^{12}. \quad (15)$$

It is not easy to evaluate the equation above. However, given the fact that

$$p(X_k, Z_k) = p(z_k|x_k)p(x_k|x_{k-1})p(X_{k-1}, Z_{k-1}). \quad (16)$$

Let $X^k = [(X_{k-2})^T, x_{k-1}^T, x_k^T]^T$, we have,

$$J(X_k) = J([(X_{k-2})^T, x_{k-1}^T, x_k^T]^T) = \begin{bmatrix} J_{k-1}^{11} & J_{k-1}^{12} & 0 \\ (J_{k-1}^{12})^T & J_{22}^{k-1} + D_{11}^{k-1} & D_{12}^{k-1} \\ 0 & (D_{12}^{k-1})^T & D_{22}^{k-1} \end{bmatrix}, \quad (17)$$

where

$$\begin{aligned} D_{k-1}^{11} &= -\frac{\partial^2 \log p(x_k | x_{k-1})}{\partial x_{k-1} \partial (x_{k-1})'}, \\ D_{k-1}^{12} &= -\frac{\partial^2 \log p(x_k | x_{k-1})}{\partial x_{k-1} \partial (x_k)'}, \\ D_{k-1}^{22} &= -\frac{\partial^2 \log p(x_k | x_{k-1})}{\partial x_k \partial (x_k)'}. \end{aligned}$$

Therefore, similar to (13)-(14), we can express $J(x_k)$ based on (17) as

$$J(x_k) = D_{k-1}^{22} - (D_{k-1}^{12})^T (J(x_{k-1}) + D_{k-1}^{11})^{-1} D_{k-1}^{12}, \quad (18)$$

where $J(x_{k-1}) = J_{k-1}^{22} - (J_{k-1}^{12})^T (J_{k-1}^{11})^{-1} J_{k-1}^{12}$. Given a general linear Gaussian system,

$$\begin{aligned} x_{k+1} &= Ax_k + v_k, \\ z_k &= Hx_k + \omega_k, \end{aligned} \quad (19)$$

where A is the state transition matrix, H is the observation matrix, v_k and ω_k are the process noise and observation noise respectively, and the corresponding covariance matrix are Q and R . Then, we can rewrite D_{k-1}^{11} , D_{k-1}^{12} and D_{k-1}^{22} as

$$\begin{aligned} D_{k-1}^{11} &= A^T Q^{-1} A, \\ D_{k-1}^{12} &= -A^T Q^{-1}, \\ D_{k-1}^{22} &= Q^{-1} + H^T R^{-1} H. \end{aligned} \quad (20)$$

Substitute these equations into (18), we can evaluate $J(x_k)$ recursively [31].

$$J(x_k) = Q^{-1} + HR^{-1}H^T - Q^{-T}A(J(x_{k-1}) + A^TQ^{-1}A)^{-1}A^TQ^{-1}. \quad (21)$$

We can further simplify it by applying the matrix inversion lemma [32], and we get

$$J(x_k) = (Q + AJ(x_{k-1})^{-1}A^T)^{-1} + HR^{-1}H^T. \quad (22)$$

Then we can evaluate $J(x_k)$ recursively and the numerical results are shown in the following section.

B. Performance Evaluation by Simulation

We generate the simulation traces based on the measurement results in a temperature-changing environment. The temperature alternates between two states: either being constant for a while, or changing smoothly in a certain duration. Assume that the reference clock works in a stable environment or is well calibrated, and its clock jitter can be treated as part of the noise. In the simulation, the temperature varied from 20°C to 50°C and the corresponding clock skew varied from 20 μ s to 40 μ s. The measurement noises followed a Gaussian

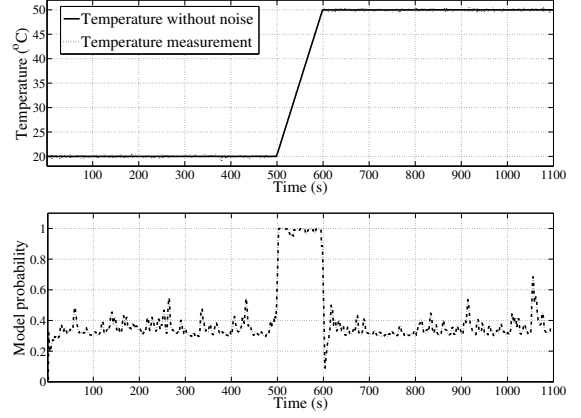


Fig. 4. Temperature measurements and the probability of constant velocity model.

distribution. We used a set of training data to obtain the table for clock skew compensation w.r.t. temperature.

We first applied the IMM Kalman filter on temperature measurements to determine the clock skew model at each time instant, and the results are shown in Fig. 4. During the period that the temperature changes from 20°C to 50°C (from the 500 to 600 s), the probability of the constant velocity model is close to one; during the period that the temperature is stable, the probability is below 0.5. As a result, during the temperature changing period, the correct model is selected for clock skew almost surely; while during the temperature stable period, there is a small probability that the constant velocity model is selected, which is acceptable as discussed in Sec. V.

For the two clock skew models, the constant and the constant velocity model, we choose the clock state vector as $x_k = [\theta_k, \alpha_k]$. The corresponding transition matrix and processing noise covariance matrix are

$$\begin{aligned} A^{(1)} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, & Q^{(1)} &= \beta \begin{bmatrix} T^2 & 0 \\ 0 & 0 \end{bmatrix}, \\ A^{(2)} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, & Q^{(2)} &= \beta \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix}, \end{aligned} \quad (23)$$

where β is a scalar to adjust the processing noise level, which reflects the accuracy of the model. It can be used to make a tradeoff between estimation accuracy and filter suitability. The similar setting is used for temperature except that the state vector contains temperature and temperature change.

Given the model probability obtained from the IMM filter on temperature measurements, the model in the Kalman filter for skew estimation is selected. The estimation results are shown in Fig. 5. From the figure, the estimation results using the proposed AMKF (the dashed line) are close to the real values (the solid line), even though the clock measurements (the dots) contain severe noise. This is because we can detect the change of model efficiently using temperature estimation, so the negative impact of model uncertainty is minimized. From the figure, the estimation errors are most obvious during the initial stage (0 to 50 s) and the period that the temperature

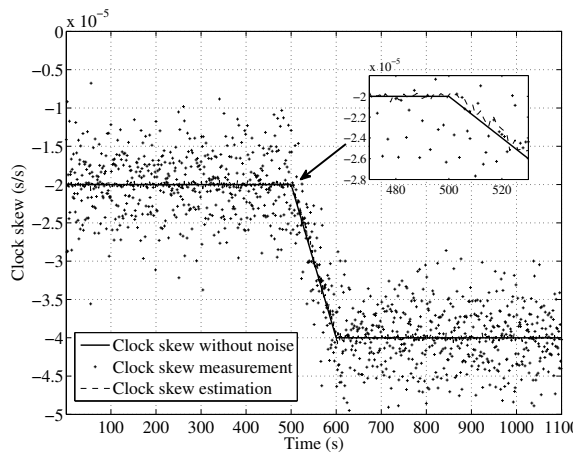


Fig. 5. Clock skew estimation.

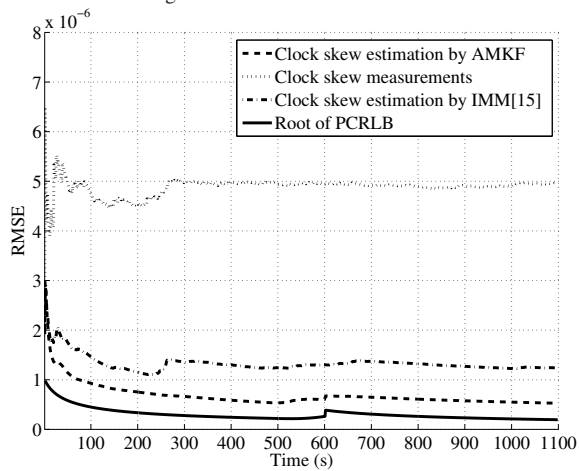


Fig. 6. RMSE of clock skew estimation.

changes to the constant state (600 to 650 s). This is because that we intentionally set the threshold of the IMM filter to favor constant velocity model, so the probability of selecting a wrong model during these periods is higher than that of other periods. Nevertheless, the estimation error of the proposed algorithm is always much less than $1 \mu\text{s/s}$ if the filter converges.

To further evaluate the estimation error, the root mean square error (RMSE) of the proposed AMKF algorithm is shown in Fig. 6. We also compare the proposed AMKF with our previous solution in [15], which used the IMM Kalman filter to process the clock skew measurements without the assistance from temperature measurements. As shown in [15], the IMM used in [15] can outperform the non-adaptive estimation algorithms (such as the one in [14]), so it is the best known solution. From the figure, the RMSE of the proposed AMKF is less than 50% of the best known results in [15]. PCRLB offers a statistic lower bound of RMSE, and the results in Fig. 6 show that the performance of the proposed AMKF is close to the bound. Generally speaking, the Kalman filter can approach the PCRLB in a linear Gaussian system with sufficient measurements. However, in dynamic environments, the process within the same model cannot be arbitrary long

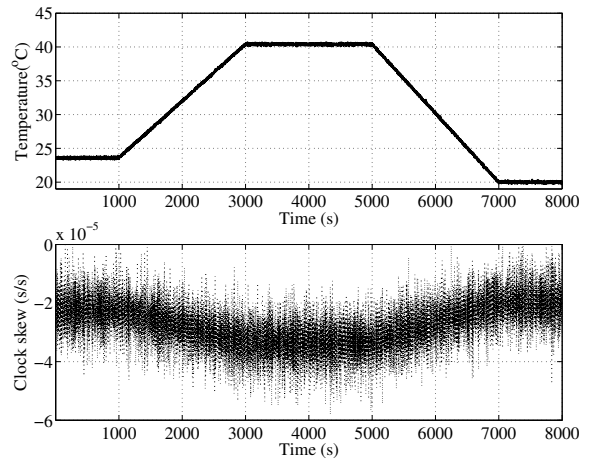


Fig. 7. Simulation setting.

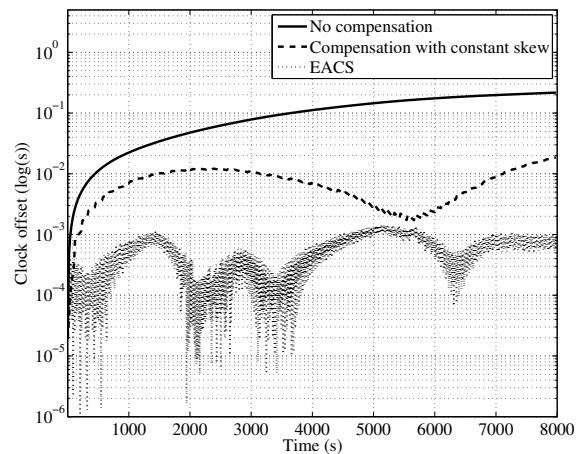


Fig. 8. Clock offset with compensation schemes.

due to the variations of clock skew and temperature models. Therefore, the gap to reach the PCRLB exists, which is mainly caused by the model change.

Next, we use the clock skew estimation error statistics to analyze the performance of the proposed EACS scheme. To evaluate the EACS scheme and compare it with the existing constant skew compensation solution, we generated simulation data as shown in Fig. 7. Initially, we set the temperature to around 23.6°C and maintained this temperature for 1,000 s, and then increased it to 40.4°C and lasted for another 2,000 s, before decreasing it to 20°C again. Clock skew has the similar trend with noise, caused by clock jitter and other uncertain factors. We set the sampling period T to be 1 s and the clock re-synchronization threshold to be 1 ms for simplicity.

The performance of EACS and constant clock skew compensation are compared in Fig. 8. We compare the EACS performance with the constant clock skew compensation, which uses the mean value of clock skew. As shown in the figure, the constant clock skew compensation only works for certain circumstances and the performance degrades significantly once the working environment changes. The clock offset is well compensated using the proposed EACS, which can

maintain the clock offset to be one order of magnitude more accurate than the constant compensation algorithm. Note that the compensation with mean skew has a wide fluctuation in clock offset, due to the temperature fluctuation.

We note that the compensated clock offset is in a zig-zag style. This is because the software-based compensation is limited by the granularity of the local clock. Therefore, only when the clock skew is accumulated to a certain threshold, it can be compensated by adding or subtracting one clock tick.

For the proposed EACS, simulation results show that the re-synchronization period can be prolonged to more than 1,500 s, while that for constant skew compensation is less than 150 s. In other words, using EACS, the synchronization communication cost is reduced by an order of magnitude. In addition, during the whole 8000 s simulation, the clock offset is always below 2 ms, which is much lower than the case without compensation or with constant skew compensation, which are more than 100 ms and 12 ms respectively. If we can conduct the calibration offline in a well controlled environment and reduce the measurement noise of clock skew by extracting the hardware timing information, the performance of EACS can be further improved.

C. Performance Evaluation by Experiment

Simulation study can use a controlled setting to investigate the performance metrics and bounds such as RMSE and PCRLB, which are not easy to obtain in real test. On the other hand, in addition to simulation, the applicability and feasibility of the proposed EACS scheme should be validated by realistic tests. We thus conducted experimental verification using the Mica2 testbed. We used the measurement results shown in Fig. 2 as the training dataset to obtain the clock skew table w.r.t. temperature. Then, six months later, we used the same sensor and the laptop to record their clock skews, where the sensor was working in a randomly changing environment, and the traces of temperature and clock skew are shown in Fig. 9. According to the clock skew compensation table obtained six-month ago, we used EACS to dynamically compensate the clock skew of the sensor node.

The test results are shown in Fig. 10. From the figure, even with a time span of six-month, the clock offset by EACS is much lower than those without compensation or with constant skew compensation. The clock offset by EACS is always below 8 ms over the 7200 s test duration, which is an order of magnitude improvement to the constant skew compensation and the no-compensation schemes, where the clock offset is up to 100 ms and 200 ms respectively. Using 1 ms as the re-synchronization threshold, the proposed EACS can enlarge the re-synchronization period to more than 1000 s compared to those for the other schemes which are less than 100 s.

On the other hand, it is not surprising that the experimental results are worse than the simulation ones for all schemes due to the following reasons. First, some often-used assumptions may not hold in a practical system, such as the measurement noise may not be strictly Gaussian distributed, which will degrade the estimation accuracy. Second, we used a laptop

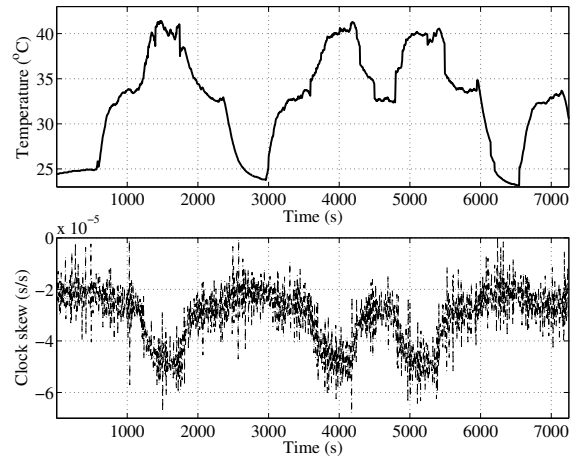


Fig. 9. Verification trace.

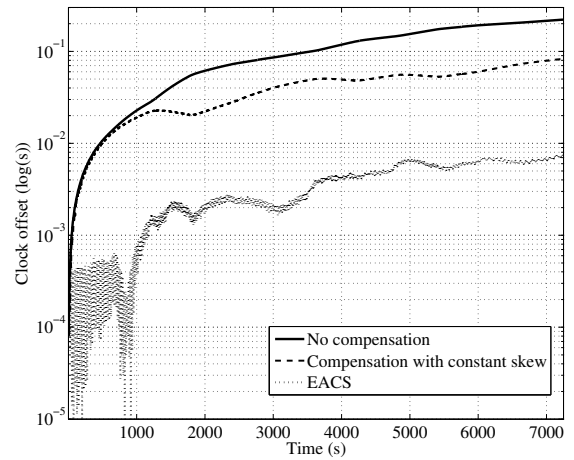


Fig. 10. Verification results.

in lab environment as the reference node, as the clock of the laptop is much more stable than that of the sensor node. However, the laptop clock circuit also suffers from thermal noise and other uncertain factors, and the lab room temperature may change slightly over the 6-month time span as our lab has no temperature control, which may introduce some errors. Third, as shown in Fig. 9, the temperature changing pattern in the real test is more random and has higher variation than that used in the simulation. These imperfect practical conditions degrade the performance of all skew estimation algorithms in the realistic test. Even with these impairments, the test results show that EACS can meet our design objective, i.e., reduce the clock offset and prolong the re-synchronization period by an order of magnitude, and its performance is reliable over several months (and probably more).

VIII. CONCLUSIONS

In this paper, we have investigated the clock skew estimation and clock synchronization for WSNs deployed in dynamic environments. Clock skew has been modeled as a hybrid system with two different models. To tackle the model uncertainty problem, we have proposed an additional

information aided multi-model Kalman filter (AMKF) which uses the temperature information to determine the suitable model for clock skew. Based on AMKF, we have proposed an EACS scheme to conduct the real-time clock compensation and re-synchronization. Simulation and experimental results have demonstrated the effectiveness and efficiency of the proposed EACS scheme, which can reduce the resynchronization frequency by an order of magnitude. How to further improve the accuracy of skew estimation especially considering about non-Gaussian noise and system jitters to approach the lower bound of the estimation error remains an open issue. One possible direction is to jointly estimate temperature and clock skew measurements with advanced filtering techniques and parameter learning. Besides temperature, more research work is needed to investigate the impact of other environment factors (such as humidity, electromagnetic interference, and vibration) on clock skew, which are treated as noise in this work as their influence is much less than temperature.

REFERENCES

- [1] Crossbow Technology. Available at http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.
- [2] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *IEEE IPSN'05*, pages 364–369, 2005.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Elsevier Comp. Netw.*, 38(4):393–422, 2002.
- [4] K. Romer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, 2004.
- [5] K.L. Noh, Q.M. Chaudhari, E. Serpedin, and B.W. Suter. Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks. *IEEE Transactions on Communications*, 55(4):766–777, 2007.
- [6] DL Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.
- [7] *Network Time Synchronization Using Clock Offset Optimization*, 2003.
- [8] M. Maroti, B. Kusy, G. Simon, and Á. Lédeczi. The flooding time synchronization protocol. In *ACM Sensys'04*, pages 39–49, 2004.
- [9] Jui-Hao Chiang and Tzi cker Chiueh. Accurate clock synchronization for IEEE 802.11-based multi-hop wireless networks. In *IEEE ICNP'09*, pages 11–20, 2009.
- [10] S. Ganeriwal, R. Kumar, and M.B. Srivastava. Timing-sync protocol for sensor networks. In *1st International conference on Embedded networked sensor systems*, pages 138–149. ACM, 2003.
- [11] N. Freris and P. R. Kumar. Fundamental limits on synchronization of affine clocks in networks. In *IEEE Conference on Decision and Control'07*, pages 921–926, Dec. 2007.
- [12] A. Bletsas. Evaluation of Kalman filtering for network time keeping. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 52(9):1452–1460, 2005.
- [13] LF Auler and R. d'Amore. Adaptive Kalman filter for time synchronization over packet-switched networks: an heuristic approach. In *IEEE COMSWARE'07*, pages 1–7, 2007.
- [14] B.R. Hamilton, X. Ma, Q. Zhao, and J. Xu. ACES: adaptive clock estimation and synchronization using Kalman filtering. In *ACM MobiCom'08*, pages 152–162, 2008.
- [15] Z. Yang, J. Pan, and L. Cai. Adaptive clock skew estimation with interactive multi-model Kalman filters for sensor networks. In *IEEE ICC'10*, 2010.
- [16] SB Moon, P. Skelly, and D. Towsley. Estimation and removal of clock skew from network delay measurements. In *IEEE INFOCOM'99*, volume 1, 1999.
- [17] L. Zhang, Z. Liu, and C. H. Xia. Clock synchronization algorithms for network measurements. In *IEEE INFOCOM'02*, pages 160–169, 2002.
- [18] M. Leng and Y.C. Wu. On Clock Synchronization Algorithms for Wireless Sensor Networks Under Unknown Delay. *IEEE Transactions on Vehicular Technology*, 59(1):182–190, 2010.
- [19] H. Khlifi and J.C. Gregoire. Estimation and removal of clock skew from delay measures. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 144–151, 2004.
- [20] MAXIM. 32.768kHz Temperature-Compensated Crystal Oscillator , Datasheet at, <http://datasheets.maxim-ic.com/en/ds/DS32kHz.pdf>.
- [21] T. Schmid, Z. Charbiwala, R. Shea, and M.B. Srivastava. Temperature compensated time synchronization. *Embedded Systems Letters, IEEE*, 1(2):37–41, 2009.
- [22] S.J. Murdoch. Hot or not: Revealing hidden services by their clock skew. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 27–36, 2006.
- [23] D. Veitch, S. Babu, and A. Pásztor. Robust synchronization of software clocks across the Internet. In *ACM SIGCOMM'04*, pages 219–232, 2004.
- [24] R. Sugihara and R. Gupta. Clock synchronization with deterministic accuracy guarantee. *Wireless Sensor Networks*, pages 130–146, 2011.
- [25] Crossbow Technology. MPR-MIB Series Users Manual. Available at http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf.
- [26] Crossbow Technology. MTS-MDA Sensor Board Users Manual. Available at http://www.xbow.com/Support/Support_pdf_files/MTS-MDA_Series_Users_Manual.pdf.
- [27] M.B. Uddin and C. Castelluccia. Toward clock skew based wireless sensor node services. In *ICST Wireless Internet Conference (WICON)'10*, pages 1–9, 2010.
- [28] HAP Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8):780–783, 1988.
- [29] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking. Part V: Multiple-model methods. In *Conference on Signal and Data Processing of Small Targets*, volume 41, pages 559–581, 2005.
- [30] G. Welch and G. Bishop. *An introduction to the Kalman filter*. University of North Carolina at Chapel Hill, Chapel Hill, NC, 1995.
- [31] P. Tichavsky, C.H. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *Signal Processing, IEEE Transactions on*, 46(5):1386–1396, 2002.
- [32] C. Liang, W. Lenan, and R. Piche. Posterior Cramer-Rao lower bound for mobile tracking in mixed LOS/NLOS conditions. In *17th European Signal Processing Conference (EUSIPCO 2009)*, pages 90–94, 2009.