Due to the ability of sensor nodes to collaborate, time synchronization is essential for many sensor network operations. This work presents a novel clock synchronization method for energy-constrained wireless sensor networks (WSNs). To conserve WSN energy, this study adopts the flooding time synchronization scheme based on one-way timing messages. Via the proposed approach, the maximum-likelihood (ML) estimation of clock parameters, such as clock skew and clock offset, can be obtained for clock synchronization. Additionally, with the proposed scheme, the clock skew and offset estimation problem will be transformed into a problem independent of random delay and propagation delay. In addition to good performance, the proposed method has low complexity.

I. INTRODUCTION

Advancements in fabrication technologies have enabled the development of tiny low-power devices capable of onboard sensing, computing, and communication tasks. Wireless sensor networks (WSNs) are a special ad hoc network formed by networking these tiny devices within a certain area without any infrastructure [1], [2]. In recent years, WSNs have received considerable attention due to their promising applications in various areas. Notably, WSNs comprise an infrastructure for future ubiquitous communication environments.

Generally, two different approaches, sender-receiver synchronization (SRS) or receiver-receiver synchronization (RRS), can be utilized to synchronize a pair of nodes [3]. The SRS approach is based on the classical model of two-way message exchanges between a node pair. Conversely, the RRS approach compares the time readings of a beacon packet from a common sender in a set of nodes. Most existing time synchronization protocols utilize one of these two approaches. For example, reference-broadcast synchronization (RBS) [4] is based on RRS as it requires pairs of message exchanges among child nodes (except for the reference node) to compensate for their relative clock offsets, whereas the time-sync protocol for sensor networks (TPSNs) [5]–[7] adopts SRS as it requires a series of pairwise synchronizations that assume two-way timing message exchanges.

The energy constraints of WSNs are very strict [8]. Hence, increasing overhead for enhanced synchronization is not an effective solution. To overcome both of these challenges (i.e., communication overhead and enhanced performance) simultaneously, one can employ energy-efficient protocols designed specifically for WSNs, as in [4], [5]–[7], and [9]. Another way is to reduce the amount of energy expended during clock synchronization via novel designs. This work adopts both strategies by utilizing a scalable protocol for flooding time synchronization. The conventional flooding time synchronization protocol (FTSP) [9] requiring only one-way messages from a sender, without needing other message exchanges between receivers or other senders, is modified to achieve the goal. A novel low-complexity approach, which is suitable for the modified FTSP, is then presented. The maximum-likelihood (ML) estimation of clock parameters is also obtained. With the proposed scheme, the clock skew and offset estimation problem is transformed into a problem independent of random delay and propagation delay. The aim of the proposed strategy is to reduce the amount of energy expended for both communication and computation while achieving good performance. When using the proposed method, the characteristics of the FTSP, such as scalability and low-bandwidth requirement, are preserved, while the needed complexity of the proposed method is greatly reduced. Hence, more power can be saved than that saved in previous work.

II. RELATED WORK

Efficient operation of a WSN largely depends on synchronization of time among its nodes. Coordination among nodes during power saving sleep/wake up modes, localization of sensor nodes, data aggregation, beamforming, and distributive communication protocols require that all nodes run on a common time frame. Additionally, conventional network synchronization protocols cannot satisfy the low-power and low-complexity requirements of WSNs [8].

The Internet uses the network time protocol (NTP) [10] to provide distributed synchronization, which includes adjusting the frequency of the oscillator of each node. Although NTP synchronizes computer clocks in a hierarchical manner by using primary and secondary time servers, it is not suited to WSNs because it does not account for energy consumption and bandwidth constraints [7].

Some synchronization protocols have been designed in recent years to deal efficiently with the specific requirements associated with long-term operation of WSNs. The TPSN [5] is a conventional sender-receiver protocol that assumes two operational stages: the level discovery phase followed by the
synchronization phase. During the level discovery phase, a WSN is organized as a spanning tree, and global synchronization is achieved by enabling each node to synchronize with its parent (the node located in the adjacent upper level) via a message exchange mechanism (Fig. 1) that adjusts only the clock offset of each node, where $\theta$ is clock offset, $d$ is fixed propagation delay, and $N$ is the number of observations. The fixed propagation delays are assumed to be symmetrical in both directions. At time $T_{1,j}$, node A sends the $i$-th Sync message to node B, which contains the value of $T_{1,j}$. Node B receives this message at time $T_{2,i} = T_{1,i} + d + \theta$. Note that time values are read by clock counters. Likewise, at time $T_{3,i}$, node B sends a message back to node A that contains the value of $T_{3,j}$. Node A then receives this message at time $T_{4,j}$. Based on this model, the clock offset is [5]

$$\theta = \frac{(T_{4,j} - T_{3,j}) - (T_{2,i} - T_{1,i})}{2}.$$ (1)

Node B adjusts its time to minimize clock offset, thereby synchronizing with the clock of node A. By including clock skew and random delay into the SRS model, ML estimations of clock skew and offset for different delay models are obtained in [6]. Likewise, a new ML estimation of clock skew and offset for an exponential delay model is developed in [7].

On the other hand, RBS [4] uses a broadcast message sent by one node to two or more neighboring nodes, which record their own local clocks when they receive the broadcast message. After collecting a few readings, the nodes exchange their observations and a linear regression approach is utilized to estimate relative clock offset and skew.

The aim of the FTSP [9] is to obtain a network wide synchronization of the local clocks of participating nodes by using multi-hop synchronization. In FTSP, a root node broadcasts its local time and any nodes that receive that time synchronize their clocks to that time. The receiving node calculates its clock skew and offset using linear regression on a set of these offsets versus the time of reception of the messages. Given the limited computational and memory resources of a sensor node, it can only keep a small number of reference points. Therefore, the linear regression is performed only on a small subset of the received nodes. Since this regression requires that set of updates, however, a node cannot calculate its clock skew until it receives a full set of reference messages. Therefore, there is a non-negligible initiation period for the network [11].

In receiver-only synchronization (ROS) [12], the receiver (within the radio range of its super nodes) reads a two-way timing message from its super nodes to estimate its clock skew and offset. The ROS is extended for multicluster sensor networks in [13].

All of these protocols have benefits and limitations. Choosing a protocol that only corrects clock offset (such as the TPSN [5]) increases power consumption since synchronization occurs frequently at regular intervals to prevent clock skew from drifting too far apart. On the other hand, assuming simultaneous reception of reference broadcasts is necessary in protocols that correct both clock offset and skew, such as the RBS [4], which is a simplification of the correct model and inapplicable in some cases (e.g., in underwater acoustic sensor networks) [14]. In the ROS approach in [12], two senders are required to send timing messages, and propagation delays of sensor nodes are assumed known; however, this assumption may not always hold.

The remainder of this paper is organized as follows. The system model and protocol are described in Section III. The proposed clock synchronization method and analytical result are presented in Section IV. Simulation results are given in Section V. Finally, conclusions are given in Section VI.

III. SYSTEM MODEL AND PROTOCOL

Figure 2 shows the system model, in which a WSN initially consists of sensor nodes and an initial anchor node. The randomly distributed sensor nodes, which are within the radio range of the initial anchor node, will be synchronized with the initial anchor node. The initial anchor node is assumed well synchronized to the precise clock via a global positioning system (GPS) receiver; otherwise, the sensor nodes will be locally synchronized (in contrast to being globally synchronized). The goal here is to synchronize the clocks of all sensor nodes in the network to that of the initial anchor node. After the sensor nodes within the radio range of the initial anchor node are synchronized with the initial anchor node, they become new anchor nodes. The sensor nodes, which are outside the radio range of the initial anchor node, will be synchronized with new anchor nodes. This process continues until all sensor nodes in a WSN are synchronized.

Figure 3 shows the conventional FTSP for the ($i$-th) message exchange. To save power, only one-way Sync messages are sent from an anchor node to sensor nodes, message exchanges from other anchor nodes or among sensor nodes are not required. The time information of the anchor node is conveyed by a broadcast Sync message sent to sensor nodes. At time $T_{1,j}$, the anchor node sends the $i$-th Sync message to the sensor node; the sensor node receives this message at time $T_{2,j}$. Via one-way Sync messages, all sensor nodes are synchronized with the anchor node. Notably, the protocol is scalable and robust in practical WSNs.
IV. PROPOSED CLOCK SYNCHRONIZATION

To use only one-way Sync messages, two sensor clocks are utilized in a sensor node to trigger clock counters. Figure 5 shows the one-way messaging scheme and its delayed version used by the proposed approach, where $T_{lj}$ is the time when the $i$-th Sync message is sent, and $T_j$ and $T_j'$ (or $T_j''$ and $T_j'''$) are the times of sensor clocks 2 and 1 when they receive the non-delayed (or delayed) Sync message, respectively. Note that the timing representation in Fig. 5 is different from that in Fig. 4 to emphasize that the random access delay be expended in the anchor node. After sending the original $i$-th Sync message (Type-0), which is the Sync message of the conventional FTSP, the delayed (by delay $\xi$) message (Type-1) is also sent for clock synchronization to notify sensor nodes the time elapse of $\xi$. The delay is folded and generated by a clock counter in the anchor node. The Type-1 Sync message is treated as the highest priority packet which is a short packet conveying a relative time difference to the Type-0 Sync message. Its arrival indicates receiving of a delayed message.

Figure 6 (on the next page) shows the schematic that represents the generation of two sensor clocks. Clock 2 runs at a frequency of $(1+\varepsilon)$, which is normalized to that of the anchor clock. A D-type Flip-Flop is employed to divide the frequency by two; therefore, sensor clock 2 runs at a frequency of $(1+\varepsilon)$, which is half that of sensor clock 1 running at a frequency of $2(1+\varepsilon)$. A single signal enables both sensor clocks. Hence, they have the same offset, which is assumed as $\theta$.

The proposed scheme makes more equations than before available to evaluate multiple unknown parameters with only one-way time transfer. To determine the clock skew and offset via a Sync message, receiving the $i$-th Type-0 Sync message yields

$$T_{lj} = (1+\varepsilon)(T_{lj} + d + X_i) + \theta + \phi_{lj}$$

(2)

and

$$T_j' = 2(1+\varepsilon)(T_{lj} + d + X_i) + \theta + \phi_{lj}$$

(3)

where $\phi_{lj}$ is a random jitter of sensor clocks 1 and 2 when receiving the $i$-th Type-0 Sync message. The random jitter $\phi_{lj}$ is assumed to be an independent Gaussian random variable (RV) with zero mean and variance $\sigma^2$. Due to the synchronous design, sensor clocks 1 and 2 have the same clock jitter. Likewise, when receiving the $i$-th Type-1 Sync message,

$$T_j'' = (1+\varepsilon)(T_{lj} + d + X_i + \xi) + \theta + \phi_{lj}'$$

(4)

and

$$T_j''' = 2(1+\varepsilon)(T_{lj} + d + X_i + \xi) + \theta + \phi_{lj}'$$

(5)

where $\phi_{lj}'$ is a random jitter with zero mean and variance $\sigma'^2$ of sensor clocks 1 and 2 when receiving the $i$-th Type-1 Sync message.

Now, with (2)–(5), the ML estimation of clock parameters can be obtained. To determine the clock skew without disturbance of the fixed propagation delay and the random access delay from the anchor node, subtracting (2) from (4) yields

$$T_{lj} - T_{lj} = (1+\varepsilon)\xi + \phi_{lj}' - \phi_{lj}$$

(6)

Since $\phi_{lj} \overset{\Delta}{=} \phi_{lj}' - \phi_{lj}$ is also a Gaussian RV with zero mean and variance $\sigma'^2 = 2\sigma^2$, assuming the observations $\{\phi_{lj}'\}_{1}^{N}$ are independently and normally distributed with the same mean.

---

1 In this paper, the packet is the same as the message unless otherwise stated.
and variance, after \( N \) (Type-0 and Type-1) Sync messages are received, the likelihood function of \( \epsilon \) based on the observation of \( \{ \phi_i \}_{i=1}^{N} \) is given by

\[
L(\epsilon) = \left( 2\pi \sigma^2 \right)^{-N} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} \left( T_i - \phi_i \right)^2} .
\]  

(7)

Differentiating the log-likelihood function gives

\[
\frac{\partial \ln L(\epsilon)}{\partial \epsilon} = \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left( T_i - \phi_i \right) \left( T_i - \phi_i \right)^{-1} - 1.
\]

Hence, the ML estimate of \( \epsilon \) is given by,

\[
\hat{\epsilon} = \arg \max_{\epsilon} \left[ \ln L(\epsilon) \right] = \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left( T_i - \phi_i \right)^{-1} - 1.
\]

(9)

Similarly, to determine clock offset, multiplying (2) by 2 and subtracting (3) from the result yield

\[
2T_i - T_i' = \theta + \phi_i .
\]

(10)

The likelihood function of \( \theta \) based on the observation of \( \{ \phi_i \}_{i=1}^{N} \) is given by

\[
L(\theta) = \left( 2\pi \sigma^2 \right)^{-N} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^{N} \left[ 2T_i - T_i' - \theta \right]^2} .
\]

(11)

Therefore, the ML estimate of \( \theta \) is given by,

\[
\hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} \left( 2T_i - T_i' \right) .
\]

(12)

Finally, subtracting (2) from (3) and (4) from (5) respectively yield

\[
T_i' - T_i = \left( 1 + \epsilon \right) \left( T_i - d + X_i \right)
\]

(13)

and

\[
T_i^\sigma - T_i = \left( 1 + \epsilon \right) \left( T_i - d + X_i + \xi_i \right).
\]

(14)

Dividing (13) by (14), and after some manipulation, yield

\[
\hat{Z}_i = \frac{\left( T_i' - T_i \right)}{\left( T_i^\sigma - T_i \right)} .
\]

(15)

where total delay is \( Z_i = d + X_i \).

Remarkably, the estimate of \( Z \) is not affected by the random clock jitter, while the estimate of \( \theta \) is not affected by \( \xi_i \). Additionally, the proposed scheme needs simple operations, as in (9), (12), and (15). Figures 6 show the hardware requirement for generating two sensor clocks in a receiver, which is very low. With the proposed scheme, the clock skew and offset estimation problem has been transformed into a problem independent of random delay \( X_i \) and propagation delay \( d \), and no assumption on the distribution of \( X_i \) is made for the estimation of clock skew and offset.

Due to page limitations, only the final result of Cramér-Rao lower bounds (CRLBs) on the estimated parameters are shown below

\[
\text{var} \left[ \epsilon \right] \geq -E \left[ \frac{\partial^2 \ln L(\epsilon)}{\partial \epsilon^2} \right]^{-1} = \frac{2\sigma^2}{N M_\epsilon}.
\]

(16)

and

\[
\text{var} \left[ \theta \right] \geq \frac{\sigma^2}{N}.
\]

(17)

where \( M_\epsilon \) is the second moment of \( \xi_i \), and depends on the transmission protocol of the medium access control (MAC) layer. For example, if the MAC layer has a kind of radio reservation mechanism which reserves a time slot (for the time-division multiple access) with a fixed delay after successfully sending Type-0 packet, then \( \xi_i \) could be a constant. Thus, the information of \( \xi_i \) in Type-1 packets could be omitted or not transmitted; otherwise, \( \xi_i \) should be informed to sensor nodes because the exact transmission time of Type-1 packet is not known before its successful transmission.

The obtainable resolution of the clock skew estimate depends on the length of the delay, i.e., \( \xi_i \). We assume clock 2 nominal frequency is 25 MHz. To derive a clock skew resolution as low as 1 ppm, the intended delay should not be too small because 25 MHz × 1 ppm = 25 Hz which imposes the constraint that \( \xi_i \) must be \( \geq 40 \text{ ms} \).

V. SIMULATIONS

The performances of the proposed method, the maximum-likelihood estimator for exponential delay model (EMLLE) [6], the maximum-likelihood estimator for Gaussian delay model (GMLLE) [6], and the algorithm 5 (ALG5) [7] are evaluated by simulations. Because EMLLE, GMLLE, and ALG5 are based on two-way timing messages, the scenario of two-way timing messages is constructed in simulations; however, only one-way messages are utilized by the proposed method. The assumed clock offset is 50s. The fixed link propagation delay is 3.33 \( \mu \text{s} \). Random delay \( X_i \) is generated by the exponential RV with a mean of 3.33 \( \mu \text{s} \). The clock frequencies of sensor clock 1 and 2 in the proposed approach are assumed to be 50 MHz and 25 MHz, respectively. The anchor clock frequency is 25 MHz. The standard deviation \( \sigma \) of the random jitter is \( \pm 60 \text{ ps} \). To obtain a clock skew resolution of 1 ppm resolution, the intended delay \( \xi_i \) of Sync messages should be larger than 40 ms. The assumed interval of Sync messages is 1s. Without loss of generality, the Type-1 Sync message is assumed to have a successful transmission probability of 0.5 in each packet transmission. If undergoing a
The proposed estimator achieves a smaller MSE than the other estimators because clock skew and offset are accurately obtained independent of random delay $X_i$ and propagation delay $d$.

Figure 7 plots the mean squared error (MSE) of the estimated clocks using different methods as a function of the number of observations ($N$) under a typical clock skew of 40 ppm. The proposed estimator achieves a smaller MSE than the other estimators because clock skew and offset are accurately obtained independent of random delay $X_i$ and propagation delay $d$.

Figure 8(a) and (b) respectively plot the MSE of the estimated clock skew and clock offset of the proposed method as a function of $N$, clock skew = 40 ppm.

VI. CONCLUSION

This work demonstrates that the proposed method performs well in practical conditions. The improvement is due to the accurate evaluation of parameters, clock skew and offset, which enables correction of sensor clocks during the Sync Interval. In summary, the advantages of the proposed method are as follows: 1) applicability in energy-constrained WSNs due to the necessity of only one-way messages; 2) low complexity requiring only simple operations and hardware; 3) scalability for large sensor networks; and 4) ability to obtain a precise sensor clock by the proposed scheme. Another advantage is that the proposed method is not impacted by the asymmetrical ratio (defined by total delay from the forward direction to the reverse direction) because of the employed one-way messages.

ACKNOWLEDGEMENT

This work is supported in part by the grants NSC 98-2218-E-006-009, Taiwan.

REFERENCES