Per-Hop Delay Compensation in Time Synchronization for Multi-Hop Wireless Sensor Networks Based on Packet-Relaying Gateways

Xintao Huan, *Student Member, IEEE*, Kyeong Soo Kim, *Senior Member, IEEE*

***Abstract*—Multi-hop time synchronization is essential to large-scale deployment of IoT and wireless sensor networks. Its performance, however, is affected by the processing delays at intermediate gateway nodes, which cause cumulative synchronization error. To address this, we propose a per-hop delay compensation scheme as a simple and effective multi-hop extension option based on packet-relaying gateway nodes for energy-efficient time synchronization in wireless sensor networks. Experimental results on a real testbed demonstrate that the per-hop delay compensation scheme significantly improves the multi-hop time synchronization performance when applied to the state-of-the-art beaconless asymmetric energy-efficient time synchronization scheme and the conventional flooding time synchronization protocol.**

***Index Terms*—Multi-hop time synchronization, per-hop delay compensation, packet relaying, wireless sensor networks, energy efficiency.**

I. Introduction

TIME synchronization in wireless sensor networks (WSNs), as a fundamental service for various applications such as environment monitoring [1], [2], event detection [3], and localization [4], has been extensively studied for decades. The large-scale deployment of WSNs through multi-hop extension enables various WSN applications to be applicable to large areas which cannot be covered by single-hop communication from the head node. Multi-hop extension is also required in relatively smaller areas in order to save transmission energy of sensor nodes and overcome obstacles preventing line-of-sight communication. Depending on the way of exchanging synchronization messages, the multi-hop time synchronization schemes can be classified into two major categories: The schemes based on *two-way message exchange* and those based on *one-way message dissemination*.

Compared to the schemes based on one-way message dissemination, those based on two-way message exchange can compensate for propagation delay at the expense of additional message transmissions at sensor nodes, which increases sensor nodes' energy consumption. Among the schemes based on two-way message exchange, timing-sync protocol for sensor networks (TPSN) [5] and recursive time synchronization protocol (RTSP) [6] initiate the two-way synchronization process from sensor nodes, while *E*nergy-*E*fficient time synchronization based on *A*synchronous *S*ource *C*lock *F*requency *R*ecovery (EE-ASCFR) [7] and asymmetric high-precision time synchronization (AHTS) [8] reverse the two-way messages exchange procedure by initiating it from the head node to save the energy consumption at sensor nodes, which are many in number and typically battery-powered. Due to the propagation delay compensation, the schemes based on two-way message exchange could achieve higher synchronization accuracy, but the energy consumption for the round-trip message exchange could be an issue in case of resource-constrained WSNs, especially under multi-hop scenarios.

Owing to their straightforward one-way forwarding synchronization messages and distributed processing, conventional schemes based on one-way message dissemination gain more attraction from the research community. Flooding time synchronization protocol (FTSP) [9], a representative one-way message dissemination based scheme, relies on *synchronized nodes* for their multi-hop extension, which further distribute "reference points" to other sensor nodes not in the broadcast radius of the head and therefore consume more energy in broadcasting their own synchronization messages. Many variations, notably rapid-flooding multiple one-way broadcast time synchronization (RMTS) [10], could greatly improve the synchronization performance of the original FTSP by reducing the accumulation of per-hop error and convergence time but at the expense of increase in the number of message transmissions and computational complexity. There are also thriving consensus-based time synchronization schemes such as [11] and [12] which could improve the convergence rate in the multi-hop and clustered scenarios. These schemes, however, mainly focus on the improvement of multi-hop time synchronization performance but hardly take into account the issue of energy efficiency.

Beaconless asymmetric energy-efficient time synchronization scheme (BATS) [13] has been recently proposed to reduce the energy consumption and computational complexity of battery-powered, low-cost sensor nodes. BATS, which is based on one-way message dissemination and time translation for its multi-hop extension, is conceptually equivalent to the extension based on multiple "reference points" of the flooding time synchronization schemes. The direction of message dissemination, however, is reversed to improve the energy efficiency of sensor nodes by reducing the number of message exchanges, and most of the distributed synchronization tasks are centralized at the head to relieve the computational burden of the gateway nodes.

In this letter, we systematically analyze the feasibility of compensating for the processing delay at gateway nodes in multi-hop WSN with a major focus on the effect of precision

This work was supported in part by the Research Development Fund (under Grant RDF-16-02-39) and the Key Programme Special Fund (under Grant KSF-E-25) of Xi'an Jiaotong-Liverpool University.

X. Huan is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, U.K., and also with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: Xintao.Huan@liverpool.ac.uk).

K. S. Kim is with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou 215123, P. R. China (e-mail: Kyeongsoo.Kim@xjtlu.edu.cn).

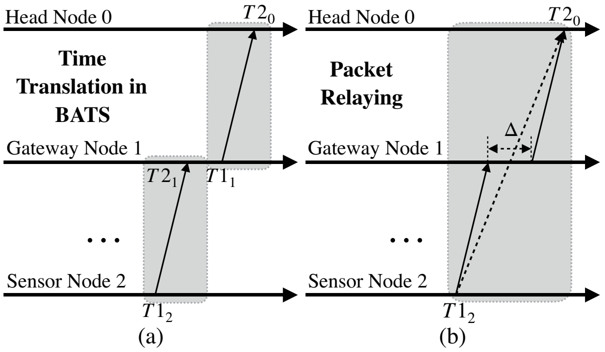

Fig. 1. Multi-hop extension based on (a) time translation proposed in BATS and (b) packet-relaying gateways.

loss and clock skew on the processing delay compensation. Based on the analysis, we propose a per-hop delay compensation (PHDC) scheme as a simpler but more effective multi-hop extension option for energy-efficient WSN time synchronization schemes leveraging one-way message dissemination and apply it to BATS and FTSP—i.e., representative WSN time synchronization schemes based on *reverse* and *conventional* one-way message dissemination—to demonstrate its feasibility and evaluate the improvement in multi-hop time synchronization performance. To the best of the authors' knowledge, this is the first time that the idea of processing delay compensation at intermediate gateways has been mathematically formulated, systematically analyzed, and implemented on a real testbed for performance evaluation in the context of WSNs since the issue of processing delays was discussed in [7].

II. Per-Hop Delay Compensation in Multi-Hop WSNs Based on Packet-Relaying Gateways

A. Multi-Hop Extension Based on Packet-Relaying Gateways

Multi-hop extension of WSN time synchronization schemes based on *time-translating* and *packet-relaying* gateways is discussed in [7]. Of the two options, multi-hop extension based on time-translating gateways is employed in the original BATS as part of its per-hop synchronization strategy, which establishes time synchronization by translating timestamps across each gateway [13]. As illustrated in Fig. 1 (a), the synchronization between the sensor node and the head node is established through translating timestamps at the gateway, which can eliminate the effect of gateway processing delays on time synchronization.

With multi-hop extension based on packet-relaying gateways, on the other hand, gateway operations are much simpler because timestamps are relayed without any translation at the gateway. The time synchronization, however, could be affected by rather large and random per-hop processing delay (i.e., Δ) resulting from queueing/scheduling and media access control (MAC) operations at each gateway as shown in Fig. 1 (b); if the per-hop delay is compensated for, the packet-relaying option could be a better alternative to the time-translating one in terms of computational complexity and energy consumption.

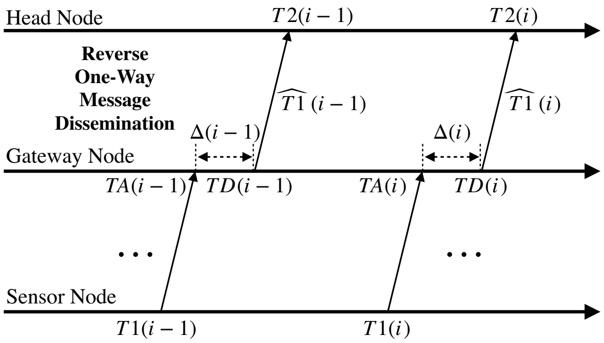

Fig. 2. A WSN with one gateway node and one sensor node.

B. Delay Compensation at a Packet-Relaying Gateway

First, we explain the basic idea of delay compensation at a packet-relaying gateway using a 2-hop WSN shown in Fig. 2 where, for ease of notation, we ignore subindexes of timestamps denoting hop number unlike Fig. 1.

The processing-delay-compensated timestamp $\widehat{T1}(i)$ for ith synchronization (i=2, 3...) can be expressed as follows:

$$\widehat{T1}(i) = T1(i) + \left\lfloor \Delta(i) \times \frac{T1(i) - T1(i{-}1)}{TA(i) - TA(i{-}1)} \right\rfloor, \tag{1}$$

where $T1(i)$, $TA(i)$, and $TD(i)$ are the MAC-layer timestamps recorded at transmission, reception and forwarding of the ith synchronization message on the sensor node and the gateway node, and $\Delta(i)$ is the processing delay defined as $TD(i){-}TA(i)$. The *floor function* (i.e., $\lfloor\cdot\rfloor$) is used to convert the processing delay scaled by the estimated clock frequency ratio between the gateway and the sensor nodes to an integer for a timestamp.

Note that the integer conversion in (1) could eliminate the effect of the precision loss in the floating-point division: Let R be the true value of the floating-point division $\frac{T1(i)-T1(i-1)}{TA(i)-TA(i-1)}$ and ε be the error resulting from the precision loss. In this case, the second term in (1) can be expressed as follows:

$$\begin{aligned}\lfloor \Delta(i) \times (R + \varepsilon) \rfloor &= \lfloor \Delta(i) R + \Delta(i)\varepsilon \rfloor \\ &= n + \lfloor \delta + \Delta(i)\varepsilon \rfloor,\end{aligned}$$

where n and δ are the integer and the fractional part of $\Delta(i)R$, respectively (i.e., $n{=}\lfloor \Delta(i)R \rfloor$ and $\delta{=}\Delta(i)R{-}n$). Therefore, if $\delta{+}\Delta(i)\varepsilon$ is less than one (i.e., $\lfloor \delta{+}\Delta(i)\varepsilon \rfloor{=}0$), the effect of the precision loss is eliminated during the integer conversion.

Note that, because $\Delta(i)$ is an integer (i.e., a difference of timestamps), if the clock frequency ratio between the gateway and the sensor nodes is close to one, the fractional part of $\Delta(i)R$ becomes negligible (i.e., $\delta{\approx}0$). In such a case, we can obtain the following upper bound for the precision loss:

$$\Delta(i)\varepsilon < 1 \Rightarrow \varepsilon < \frac{1}{\Delta(i)}. \tag{2}$$

If the precision loss is less than the upper bound given in (2), it does not affect the processing delay compensation.

1) On the Implementation without Clock Skew Compensation: When the processing delay can be managed to be lower than a certain bound, we can ignore the clock skew compensation, which would simplify PHDC at gateways. Without clock skew compensation, the processing-delay-compensated timestamp in (1) can be simplified to

$$\widehat{T1}(i) = T1(i) + \Delta(i). \tag{3}$$

Let ϵ be the clock skew between the gateway and the sensor nodes—i.e., $(1+\epsilon)$ is the clock frequency ratio between the gateway and the sensor nodes. If the estimation of the clock skew is perfect (including no precision loss in the floating-point division), the error resulting from the lack of clock skew compensation is given by[1]

$$\lfloor \Delta(i)\,(1+\epsilon) \rfloor - \Delta(i) = \lfloor \Delta(i)\epsilon \rfloor .$$

Therefore, we can obtain the following upper bound for the clock skew:

$$\Delta(i)\epsilon < 1 \Rightarrow \epsilon < \frac{1}{\Delta(i)}. \tag{4}$$

If the clock skew is less than the upper bound given in (4), it does not affect the processing delay compensation.

Alternatively, if we are given the clock tolerance specification (i.e., the maximum value of the clock skew in ppm), we can derive the upper bound for the processing delay at the gateway node, i.e.,

$$\Delta(i) < \frac{1}{\epsilon}. \tag{5}$$

A typical frequency tolerance of a crystal over the manufacturing process is ±100 ppm [14]. If we ignore the effect of temperature variation (e.g., the gateway and the sensor nodes are close to each other), the processing delay less than 10^4 does not affect the processing delay compensation. For example, if the clock resolution is 1 µs, the processing delay should be less than 10^4 µs which is 10 ms. Note that, when the processing delay exceeds this bound, the clock skew should be compensated for in order to achieve satisfactory performance.

Concerning the above analysis, we have conducted a series of comparative experiments employing two groups of sensor nodes, i.e., one meeting the bound of (4) and the other not. To evaluate the performance under a multi-hop scenario, we consider a flat 3-hop network consisting of one head, one gateway, and two sensor nodes, where the 2-hop sensor node also serves as a gateway node for the 3-hop sensor node. We carried out three experiments—i.e., BATS with packet relaying (PR), BATS with PR and delay compensation (DC) and BATS with PR, DC and clock skew compensation (SC)—for each group and show the mean absolute errors (MAEs) of the measurement time estimation in Fig. 3.

As expected, packet relaying without delay compensation results in synchronization errors of the order of milliseconds due to the uncompensated per-hop delays. With delay compensation, on the other hand, packet relaying can provide reasonable synchronization performance when the clocks of sensor and gateway nodes meet the bound of (4). When packet relaying is used with clock skew compensation as well as delay compensation, reasonable synchronization accuracy is obtained for both 2-hop and 3-hop sensor nodes even when their clocks do not meet the bound of (4).

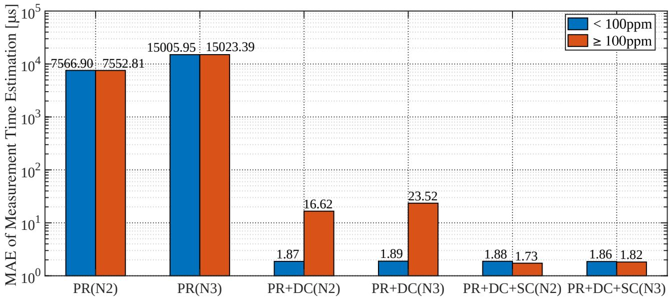

Fig. 3. MAEs of the measurement time estimation of BATS equipping packet relaying (PR), delay compensation (DC) and skew compensation (SC) evaluated on the sensor nodes with different clock skews, N2 and N3 stand for the sensor nodes located in the second hop and third hop from the head.

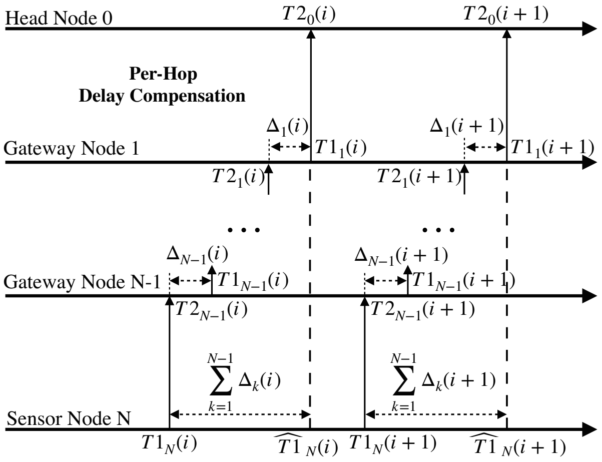

Fig. 4. PHDC in a multi-hop WSN.

Considering different temperature variations over a long distance and increasing load over multiple hops affecting the conditions of (4) and (5), packet relaying based on PHDC but without clock skew compensation would be a viable option for a small-scale WSNs or hybrid WSNs where a couple of packet-relaying gateways are used between two time-translating gateways.

C. Implementation over Multiple Gateways

When the clock frequencies of gateway and sensor nodes are synchronized to that of the reference clock at the head as in EE-ASCFR *or* the processing delays at gateways can be managed to be lower than the delay bound given in (5), we can ignore the effect of clock skew on the synchronization.

PHDC over multiple gateways, therefore, can be implemented without clock skew compensation as illustrated in Fig. 4, where we ignore the propagation delay as is the case for typical WSN communication ranges in the literature. In this case, the processing delay $\Delta_i(j)$ at gateway i during the jth synchronization can be measured as a difference between the departure time (i.e., $T1_i(j)$) and the arrival time (i.e., $T2_i(j)$) of a timestamp through MAC-layer timestamping, and the content of the received timestamp is increased by $\Delta_i(j)$ at the time of its departure as described in (3).

[1] In the following, we assume $\epsilon>0$ for simplicity.

In case the effect of clock skew cannot be ignored, a set of pairs of timestamps—i.e., $(T1, T2)$—used for the time synchronization between the two neighbor nodes could be leveraged to estimate their frequency ratio: During the jth (j=2, 3, ...) synchronization, the ratio of the clock frequency of node i to that of node $i-1$ can be estimated by[2]

$$R_{i,i-1}(j) = \frac{T1_i(j) - T1_i(j-1)}{T2_{i-1}(j) - T2_{i-1}(j-1)}. \tag{6}$$

With the clock frequency ratio, the processing delay measured at node $i-1$ can be translated to that with respect to the clock of node i, i.e.,

$$\widehat{\Delta}_{i,i-1}(j) = R_{i,i-1}(j) \times \Delta_{i-1}(j). \tag{7}$$

Based on (6) and (7), we can translate the processing delay measured at node i (i=1, ..., $N-1$) to that with respect to the clock of the originating sensor node N, i.e.,

$$\widehat{\Delta}_{N,i}(j) = \left(\prod_{k=i+1}^{N} R_{k,k-1}(j) \right) \times \Delta_i(j), \tag{8}$$

which should be added to the content of the received timestamp at its departure time. The final timestamp received by the head after a series of PHDC, therefore, is given by

$$\widehat{T1}_N(j) = T1_N(j) + \sum_{i=1}^{N-1} \widehat{\Delta}_{N,i}(j). \tag{9}$$

With this pair of $(\widehat{T1}_N(j), T2_0(j))$, the head estimates the clock parameters of the sensor node, eliminating the effect of the per-hop delays on time synchronization.

Note that for the delay compensation in (8), the product of ratios (i.e., $\prod_{k=i+1}^{N} R_{k,k-1}(j)$) should be known to each gateway. For this, we can add the product of ratios to the synchronization message at each gateway or we can move all the calculations (i.e., (6)–(8)) to the head node as in BATS, the latter of which can relieve the computational burden of the gateways at the expense of the increased communication overhead [13].

III. Experimental Results

We have set up a WSN testbed for the 6-hop linear topology shown in Fig. 5 to evaluate the improvement in multi-hop time synchronization performance made by the proposed PHDC when applied to BATS and FTSP based on packet-relaying gateways. The sensor nodes are TelosB motes running TinyOS equipped with a 32-kHz crystal oscillator (CO) having a resolution of 30.5 µs [15].

Since the state-of-the-art WSN time synchronization schemes could achieve microsecond to sub-microsecond synchronization accuracy [7], 30.5 µs resolution provided by the 32-kHz CO is insufficient. To achieve microsecond-level synchronization accuracy, we employ the sensor node's internal clock for timestamping, which is driven by a digitally-controlled oscillator (DCO) and has a minimum resolution of 1 µs. From preliminary experiments, however, we found that the accuracy of DCO-driven internal clocks does not always meet the requirement of (4) even after the calibration by the 32-kHz CO as described in [16]; in fact, the skews of 6 sensor nodes' internal clocks measured with respect to that of a reference node turn out to be up to thousands of ppms as shown in Table I. In addition to the clock skews, we also measured the processing delays and found that they are around 8 ms on the sensor node platform we employed.

TABLE I
Skews of 6 TelosB sensor nodes' internal clocks.

Node	1	2	3	4	5	6
Skew [ppm]	1080	2340	60	1460	360	37

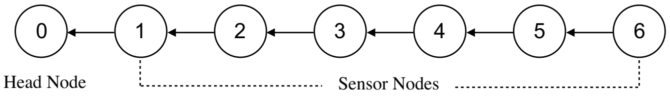

Fig. 5. A WSN with 6-hop linear topology.

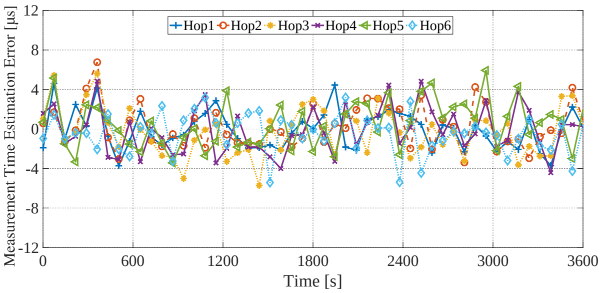

Fig. 6. Measurement time estimation errors of BATS based on packet-relaying gateways with PHDC.

Employing these sensor nodes, we have implemented the proposed PHDC with clock skew compensation on BATS with the self-data bundling option [13]—i.e., one measurement is bundled in each measurement message—for fair comparison with conventional time synchronization schemes like FTSP. Fig. 6 shows the measurement time estimation errors of BATS with PHDC over 3600 s with a synchronization interval of 1 s; during the experiments, each sensor node periodically sends measurement data to the head via measurement messages. The results show that all the sensor nodes across six hops achieve nearly the same performance of the measurement time estimation errors in the range of ±7 µs with no clear sign of the cumulation of synchronization errors over multiple hops. This indicates that, with proper compensation of both per-hop delay and clock skew at each gateway node, the multi-hop synchronization reduces to the synchronization between the head and the end node, which is not much affected by the intermediate gateway nodes.

The improvement in multi-hop synchronization performance by the proposed PHDC becomes clearer when compared to BATS with time translation (TT) as shown in Fig. 7, where we also include the results of PHDC applied to FTSP—i.e., a representative of popular flooding-based time synchronization schemes based on conventional one-way message

[2] (6) is an example, and advanced skew estimation schemes based on more samples can be applied as in [13].

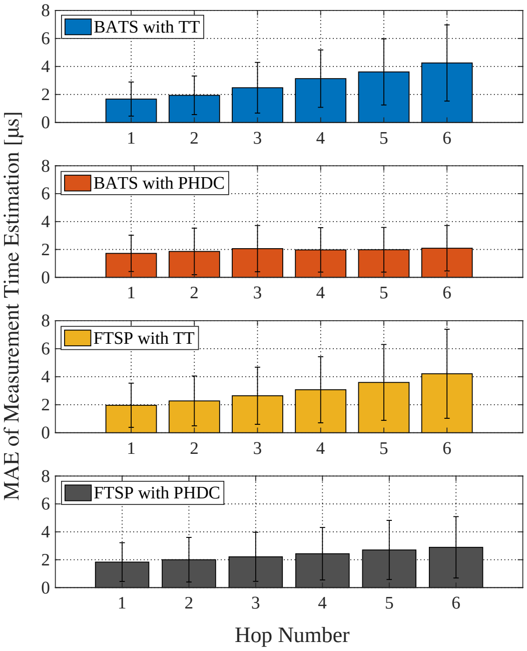

Fig. 7. MAEs of the measurement time estimation of BATS with TT, BATS with PHDC, FTSP with TT, and FTSP with PHDC.

dissemination—to demonstrate the effectiveness of the proposed PHDC for other time synchronization schemes. From Fig. 7, we observe that the hop number hardly affects MAE of the measurement time estimation and its standard deviation for BATS with PHDC, achieving the average value of 1.95 µs for MAE over 6 hops. In case of the BATS with TT and FTSP with TT, both MAEs and their standard deviations increase as the hop number increases, which indicates the cumulation of synchronization errors over multiple hops.

The difference between the multi-hop synchronization performance of BATS with PHDC and that of FTSP with PHDC, by the way, shows the importance of the accurate clock skew compensation, because BATS has advantages in compensating for clock skew over FTSP as discussed in [8] and [13]. Though the proposed PHDC applied to FTSP shows multi-hop synchronization performance not as good as that of BATS with PHDC, it could provide an alternative option for multi-hop extension of FTSP, which is simpler to implement but with more than 50% improvement in per-hop synchronization error over TT,—i.e., 0.21 µs in FTSP-PHDC vs. 0.45 µs in FTSP-TT.[3]

IV. Concluding Remarks

In this letter, we have proposed a PHDC scheme for multi-hop WSN time synchronization based on packet-relaying gateways, analyzed the effect of precision loss and clock skew on it, and discussed implementation options for the cases with and without clock skew compensation. We applied PHDC to both BATS, the state-of-the-art energy-efficient time synchronization scheme, and FTSP, the conventional flooding time synchronization scheme, in order to evaluate its effectiveness in improving multi-hop time synchronization performance. Experimental results on a real WSN testbed demonstrate that, when applied to BATS, the proposed scheme can properly address the cumulative synchronization error over multiple hops and that, in case of FTSP, it could provide a simper option for multi-hop extension with more than 50% improvement in per-hop synchronization error over TT.

[3] 0.5 µs per-hop synchronization error is reported for the original FTSP [9].

References

[1] M. Carminati, O. Kanoun, S. L. Ullo, and S. Marcuccio, "Prospects of distributed wireless sensor networks for urban environmental monitoring," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 6, pp. 44–52, Jun. 2019.

[2] T. Shu, J. Chen, V. K. Bhargava, and C. W. de Silva, "An energy-efficient dual prediction scheme using LMS filter and LSTM in wireless sensor networks for environment monitoring," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6736–6747, Aug. 2019.

[3] W. Zhu, J. Cao, and M. Raynal, "Energy-efficient composite event detection in wireless sensor networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 177–180, Jan. 2018.

[4] J. Wang, D. Fang, Z. Yang, H. Jiang, X. Chen, T. Xing, and L. Cai, "E-HIPA: An energy-efficient framework for high-precision multi-target-adaptive device-free localization," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 716–729, Mar. 2017.

[5] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. SenSys'03*, Nov. 2003, pp. 138–149.

[6] M. Akhlaq and T. R. Sheltami, "RTSP: An accurate and energy-efficient protocol for clock synchronization in WSNs," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 3, pp. 578–589, Mar. 2013.

[7] K. S. Kim, S. Lee, and E. G. Lim, "Energy-efficient time synchronization based on asynchronous source clock frequency recovery and reverse two-way message exchanges in wireless sensor networks," *IEEE Trans. Commun.*, vol. 65, no. 1, pp. 347–359, Jan. 2017.

[8] X. Huan and K. S. Kim, "On the practical implementation of propagation delay and clock skew compensated high-precision time synchronization schemes with resource-constrained sensor nodes in multi-hop wireless sensor networks," *Computer Networks*, vol. 166, p. 106959, Jan. 2020.

[9] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi, "The flooding time synchronization protocol," in *Proc. 2nd Int. Conf. SenSys*, Nov. 2004, pp. 39–49.

[10] F. Shi, X. Tuo, S. X. Yang, J. Lu, and H. Li, "Rapid-flooding time synchronization for large-scale wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1581–1590, Mar. 2020.

[11] H. Wang, D. Xiong, L. Chen, and P. Wang, "A consensus-based time synchronization scheme with low overhead for clustered wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1206–1210, Aug. 2018.

[12] F. Shi, X. Tuo, L. Ran, Z. Ren, and S. X. Yang, "Fast convergence time synchronization in wireless sensor networks based on average consensus," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1120–1129, Feb. 2020.

[13] X. Huan, K. S. Kim, S. Lee, E. G. Lim, and A. Marshall, "A beaconless asymmetric energy-efficient time synchronization scheme for resource-constrained multi-hop wireless sensor networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1716–1730, Mar. 2020.

[14] *Texas Instruments: Selection and Specification of Crystals for Texas Instruments USB 2.0 Devices*, Accessed: 2020-03-16. [Online]. Available: https://www.ti.com/lit/an/slla122/slla122.pdf

[15] *Texas Instruments: MSP430 32-kHz Crystal Oscillators*, Accessed: 2020-03-16. [Online]. Available: http://www.ti.com/lit/an/slaa322d/slaa322d.pdf

[16] *TelosB datasheet*, Accessed: 2020-03-16. [Online]. Available: http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf