A Beaconless Asymmetric Energy-Efficient Time Synchronization Scheme for Resource-Constrained Multi-Hop Wireless Sensor Networks

Xintao Huan, Kyeong Soo Kim, *Senior Member, IEEE*, Sanghyuk Lee, Eng Gee Lim, *Senior Member, IEEE*, and Alan Marshall, *Senior Member, IEEE*

***Abstract*—The ever-increasing number of WSN deployments based on a large number of battery-powered, low-cost sensor nodes, which are limited in their computing and power resources, puts the focus of WSN time synchronization research on three major aspects of accuracy, energy consumption, and computational complexity. In the literature, the latter two aspects haven't received much attention compared to the accuracy of WSN time synchronization. Especially in multi-hop WSNs, intermediate gateway nodes are overloaded with tasks for not only relaying messages but also a variety of computations for their offspring nodes as well as themselves. Therefore, not only minimizing the energy consumption but also lowering the computational complexity while maintaining the synchronization accuracy is crucial to the design of time synchronization schemes for resource-constrained sensor nodes. In this paper, focusing on the three aspects of WSN time synchronization, we introduce a framework of reverse asymmetric time synchronization for resource-constrained multi-hop WSNs and propose a beaconless energy-efficient time synchronization scheme based on reverse one-way message dissemination. Experimental results with a WSN testbed based on TelosB motes running TinyOS demonstrate that the proposed scheme conserves up to 95% energy consumption compared to the flooding time synchronization protocol while achieving microsecond-level synchronization accuracy.**

***Index Terms*—Beaconless time synchronization, energy efficiency, reverse asymmetric time synchronization, wireless sensor networks.**

I. Introduction

TIME synchronization for wireless sensor networks (WSNs) has been extensively studied in the last decades as the number of WSN deployments has been gradually increasing over the period [1]–[3]. Because most of the WSN deployments are based on a large number of battery-powered, low-cost sensor nodes, which are limited in their computing and power resources, the focus of WSN time synchronization research has been shifted toward three major aspects of *accuracy*, *energy consumption*, and *computational complexity*.

In the literature, many conventional [4]–[6] and recent [7]–[9] time synchronization schemes have been proposed to address the aspect of synchronization accuracy due to its significance. Over the past few years, a few schemes [10]–[13] tried to address the issues of energy consumption and computational complexity in time synchronization together with its accuracy. As the Internet of Things (IoT) thrives, many research efforts have been concentrated on conserving the energy of IoT systems, e.g., the conventional parameter adaptation [14]–[16] and the novel energy harvesting schemes [17]; in such an environment, the time synchronization schemes—e.g., [11], [12], [18], [19]—have been focusing on energy efficiency along with other requirements such as high synchronization accuracy and low computational complexity. In general, however, the attention received for the aspects of energy consumption and computational complexity is relatively less compared to that for the time synchronization accuracy. Especially in multi-hop WSNs, intermediate gateway nodes are overloaded with tasks for not only relaying messages but also a variety of computations for their offspring nodes as well as themselves. Therefore, not only minimizing the energy consumption but also lowering the computational complexity while maintaining the synchronization accuracy is crucial to the design of time synchronization schemes for resource-constrained sensor nodes.

In [20], unlike many existing WSN time synchronization schemes like Flooding Time Synchronization Protocol (FTSP) [4] and Timing-sync Protocol for Sensor Networks (TPSN) [6], we put our focus on *asymmetric WSNs*—where a head node[1] is equipped with a powerful processor and supplied power from outlet and sensor nodes measuring data and/or detecting events with sensors are limited in processing and battery-powered—and proposed a novel energy-efficient time synchronization scheme based on the reverse two-way message exchange and demonstrated through simulation experiments that sub-microsecond-level synchronization accuracy could be achieved.

Note that the computational precision required by the scheme proposed in [20] (i.e., the precise division of the floating-point numbers) is beyond the capability of most resource-constrained sensor nodes equipped with a low-cost MicroController Unit (MCU) providing only 32-bit floating-point as discovered in [21]. Also, its multi-hop extension through intermediate gateway nodes was discussed, but its

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, S. Lee and E. G. Lim are with the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, P. R. China (e-mail: {Kyeongsoo.Kim, Sanghyuk.Lee, Enggee.Lim}@xjtlu.edu.cn).

A. Marshall is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK (e-mail: Alan.Marshall@liverpool.ac.uk).

[1]The head node and the monitoring station (i.e., a PC or a workstation for data processing) connected to it locally or remotely over the Internet are collectively called the head in this paper.

performance under multi-hop scenarios was not analyzed at all. The idea of the reverse two-way message exchange together with message bundling for synchronization and measurement data, however, could be applied to the design of more advanced energy-efficient time synchronization schemes targeting resource-constrained sensor nodes, which could greatly reduce the number of message transmissions by sensor nodes.

Based on our prior work, therefore, in this paper we present the reverse asymmetric time synchronization framework illustrated in Fig. 1 and propose the Beaconless Asymmetric energy-efficient Time Synchronization (BATS) scheme specifically based on the reverse one-way message dissemination shown in Fig. 1 (a), which can address in a more balanced way the three major challenges in WSN time synchronization on resource-constrained sensor nodes—i.e., achieving high synchronization accuracy, reducing energy consumption [12], and lowering computational complexity at sensor nodes [22]—as follows:

First, in the proposed scheme all synchronization procedures but timestamping are moved from sensor nodes, including gateway nodes, to the head as in [21] in order to improve the accuracy of time synchronization by addressing the issue of precision loss resulting from the use of 32-bit single-precision floating-point numbers at sensor nodes: Because all procedures but timestamping are carried out at the head with plenty of computing and power resources including 64-bit floating-point precision, numerical computational errors due to precision loss could be avoided, which will be discussed in detail in Sec. A.

Second, the movement of synchronization procedures to the head greatly reduces the computational complexity of sensor nodes as well. In the proposed scheme, sensor nodes are responsible for only the timestamping procedure whose computational complexity is $O(1)$; the computational complexity for sensor nodes under the conventional schemes using linear regression (e.g., FTSP [4]), for example, could be as high as $O(CN^2)$ in case the least squares method is used for N training samples and C variables [23].

Third, noting that the propagation delay compensation through the two-way message exchange is not required for microsecond-level synchronization [4], we also design BATS based on the reverse one-way message dissemination shown in Fig. 1 (a) to lower the energy consumption of battery-powered sensor nodes; due to the one-way message dissemination under the reverse asymmetric time synchronization framework, BATS does not rely on the "Beacon/Request" messages and thereby saves the energy for their receptions and transmissions at sensor nodes, the former of which often consume more energy than the latter [24]. As in [20], application messages (e.g., for reporting measurement data to the head) can also embed and carry synchronization-related data, which further reduces the number of message transmissions. This *beaconless* time synchronization of BATS is a major advantage compared to many existing time synchronization schemes relying on sensor nodes' broadcasting synchronization messages received from a root node (i.e., *flooding*) to achieve network-level time synchronization (e.g., [4], [25]), because it eliminates beacon-related computation and energy consumption imposed on the resource-constrained sensor nodes which are already loaded with tasks including medium access control (MAC) protocol, message scheduling and routing, and data measurement.

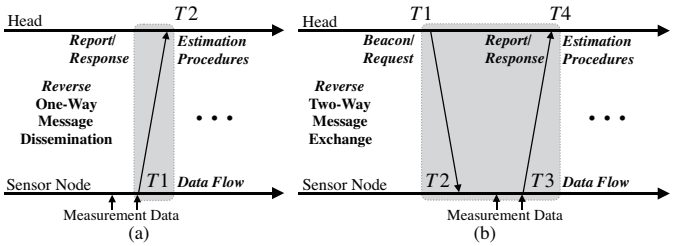

Fig. 1. Overview of the proposed reverse asymmetric time synchronization based on (a) one-way message dissemination and (b) two-way message exchange with optional bundling of measurement data.

Note that the actual energy consumption and synchronization accuracy of the proposed scheme are evaluated and analyzed through experiments on a real WSN testbed in this paper. To the best of our knowledge, this is the first work that shows the actual performance of energy efficiency of the high-precision time synchronization schemes on resource-constrained sensor nodes through real experiments.

The rest of the paper is organized as follows: In Section II, the related work including the time synchronization scheme proposed in [20] based on reverse two-way message exchange and the conventional time synchronization schemes based on one-way message dissemination is discussed. Section III presents the proposed BATS scheme based on the reverse one-way message dissemination. Section IV discusses the multi-hop extension of BATS scheme with a focus on the issue of communication overhead. The results of the performance evaluation of BATS scheme in terms of energy consumption and synchronization accuracy on a real testbed are presented in Section V. Section VI concludes our work in this paper and highlights the future work.

II. Related Work

Based on the reverse asymmetric time synchronization framework and the one-way message dissemination, the proposed BATS scheme combines in a unique way the advantages of several existing WSN time synchronization schemes and mobile computing frameworks. In this section, we review the related work in comparison to our work.

A. Reverse Asymmetric Time Synchronization

BATS is optimized for energy efficiency by reversing the flow of synchronization messages from "head→sensor nodes" to "sensor nodes→head", which is the key idea of the reverse asymmetric time synchronization framework discussed in Sec. II-A. The reverse asymmetric time synchronization framework is a generalization of the schemes we have proposed in [20] and [21] and belongs to a broader category of *reactive time synchronization protocols* [26], where sensor nodes' local clocks, which are not synchronized to a common reference, are used to timestamp events; synchronization takes place after the event has been detected.

The energy-efficient time synchronization scheme based on the asynchronous source clock frequency recovery (SCFR) and the reverse two-way message exchange shown in Fig. 1 (b)—which we proposed in [20] and is called *EE-ASCFR* in short from now on—is the first WSN time synchronization scheme exploiting the reverse asymmetric time synchronization framework in saving the energy consumptions at sensor nodes by reducing the number of message transmissions. In EE-ASCFR, the estimation of the clock offset and frequency ratio is also separated into the head and the sensor node to reduce the computational complexity of the latter where a logical clock is maintained based on the estimated clock frequency ratio. The number of message transmissions at sensor nodes is further reduced through bundling several measurement data together with synchronization data in a "Report/Response" message.

Compared to BATS, EE-ASCFR is based on the reverse two-way message exchange requiring beacon messages and parts of its time synchronization procedures—i.e., the estimation of the clock frequency ratio and the maintenance of the logical clock based on it—are still done at sensor nodes. Unlike BATS, EE-ASCFR is mainly designed for single-hop WSNs, though its multi-hop extension was briefly discussed without any performance analysis in [20].

BATS is also different from the existing reactive time synchronization protocols like Routing Integrated Time Synchronization protocol (RITS) [25] in its multi-hop extension, where, unlike RITS, there is no layer-by-layer time translation at gateway nodes along a multi-hop path; instead, the said time translation is completely moved to the head to relieve the burden of the gateway nodes.

B. Time Synchronization Based on One-Way Message Dissemination

Compared to the time synchronization schemes based on the two-way message exchange, those based on the one-way message dissemination have simpler synchronization procedures as shown in Fig. 2 (a) (i.e., conventional one) and Fig. 1 (a) (i.e., reverse one).

FTSP is a representative example of the time synchronization schemes based on the one-way message dissemination. In FTSP, the linear regression is used to estimate the clock frequency ratio and offset, i.e., the slope and the intercept of the regression line. However, due to the limited computing resources of the sensor node, FTSP limits the number of timestamp samples used for the least squares estimation of the slope and intercept: Specifically, past 8 timestamp values are employed for all different synchronization intervals (SIs) during the performance evaluation in [4]. In addition to the linear-least-squares-based clock frequency ratio and offset estimation, the timestamping procedure of FTSP records multiple timestamps—i.e., timestamps at each byte boundary after the SYNC bytes as shown in Fig. 3—in sending and receiving a synchronization message to reduce the jitter of the interrupt handling and encoding/decoding times, which is also quite demanding for resource-constrained sensor nodes [4].

To address the high computational complexity of estimating the clock skew and offset to achieve microsecond-level accuracy in FTSP, the Ratio-based time Synchronization

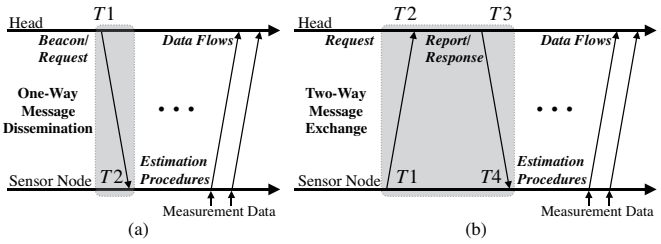

Fig. 2. Overview of the conventional time synchronization based on (a) one-way message dissemination and (b) two-way message exchange.

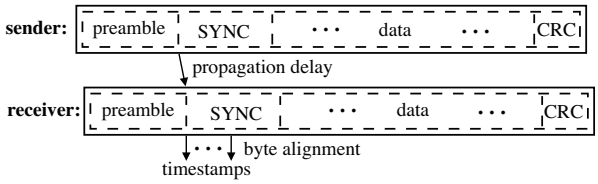

Fig. 3. MAC-layer timestamping procedure introduced in FTSP [4].

Protocol (RSP) has been proposed based on the periodical one-way synchronization message dissemination and a lightweight procedure for the estimation of clock parameters [27]. In RSP, additional thresholds and procedures are introduced to reduce the impact on the clock parameter estimation of the numerical errors by the limited floating-point precision (i.e., 32-bit floating-point numbers) of the resource-constrained sensor nodes. However, like FTSP, RSP also relies on the periodical broadcasting of the synchronization messages and, therefore, is not energy efficient in terms of message transmissions by sensor nodes.

Note that most of the time synchronization schemes based on one-way message dissemination, including FTSP and RSP, rely on flooding to achieve network-level time synchronization, where the energy consumption caused by the layer-by-layer broadcasting and the computational complexity of estimating the clock skew and offset could be high for battery-powered, low-cost sensor nodes. In case of the Reference Broadcast Synchronization (RBS) [5], because it requires not only additional reference nodes to broadcast the common clock time but also additional message exchanges between the sensor nodes to estimate their relative time differences, it is also not energy-efficient and proper for resource-constrained sensor nodes.

Being based on the one-way message dissemination, BATS can also take the advantage of simpler implementation compared to those based on the two-way message exchange. Unlike the conventional schemes based on the one-way message dissemination, however, BATS is based on the reverse asymmetric time synchronization framework, where the flow of synchronization messages is reversed from "head→sensor nodes" to "sensor nodes→head", and thereby does not rely on beacon messages.

C. Impact of Limited Precision Floating-Point Arithmetic on High-Precision Time Synchronization

Even with the relatively simpler clock parameter estimation procedures, RSP suffers from numerical errors caused by the limited precision of the low-cost MCUs (i.e., 32-bit single-precision floating-point numbers) of the resource-constrained sensor nodes: Specifically, the proposers of RSP suggest to use SI that is large enough to mitigate the impact of the numerical errors on the estimation of clock parameters.

Likewise, the actual performance of EE-ASCFR on resource-constrained sensor nodes turns out to be poorer than expected due to the precision loss resulting from the use of single-precision floating-point numbers. In this regard, we have proposed an improved version of the scheme along with its multi-hop extension based on the reverse asymmetric framework and demonstrated satisfactory time synchronization accuracy on a real WSN testbed [21].

It is worth mentioning that the use of 32-bit single-precision floating-point format is common not only for the resource-constrained sensor node platforms (e.g., TelosB [24] and MicaZ [28]) but also for the lightweight Arduino boards [29], which are frequently used as hardware platforms for IoT prototyping as discussed in [30].

D. Computation Offloading

Computation offloading transfers resource intensive computational tasks to an external platform like a cloud or an edge device (e.g., a base station) to alleviate the restrictions of limited resources in mobile systems [31]. Computation offloading, which has been studied mainly in the context of mobile cloud [32] and mobile edge [31] computing, has recently attracted wide attention from IoT community [17], [33]. The reallocation of synchronization procedures to the head in BATS can be considered as a special case of computation offloading in the different context of WSN with a specific focus on time synchronization.

III. Energy-Efficient Time Synchronization Tailored for Resource-Constrained Sensor Nodes

When the propagation delay is not significant (e.g., sub-microsecond delays for WSNs with a communication range of 300 m or less), time synchronization schemes based on the one-way message dissemination have a clear advantage over those based on the two-way message exchange in terms of the number of message transmissions at sensor nodes. Still, the schemes based on the one-way message dissemination have issues in their implementation on resource-constrained sensor nodes as discussed in Sec. II-B and more specifically in Appendix A. Here we introduce BATS—i.e., the energy-efficient time synchronization scheme based on the reverse one-way message dissemination—which addresses the implementation issues of the one-way-message-dissemination-based time synchronization schemes on resource-constrained sensor nodes.

As illustrated in Fig. 4, the proposed BATS scheme based on the reverse one-way message dissemination does not rely on beacon messages but utilizes only measurement data messages

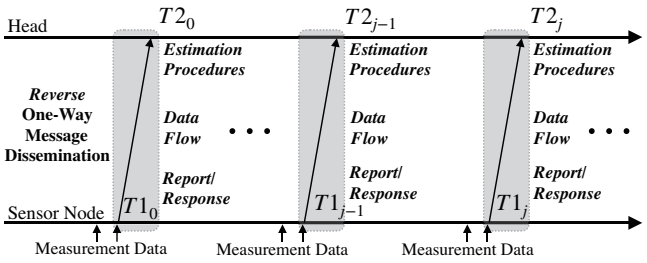

Fig. 4. Asymmetric energy-efficient time synchronization based on reverse one-way message dissemination with optional bundling of measurement data.

to carry the reversed synchronization timestamp $T1$, which results in the movement of all time synchronization procedures from sensor nodes to the head except timestamping of $T1$. As shown in Fig. 5, the timestamping of $T1$ and $T2$ is triggered by an interrupt generated by the radio chip immediately after the Start Frame Delimiter (SFD) byte has been sent and received by the MAC layer of the sender and the receiver, respectively, which is similar to that of the Recursive Time Synchronization Protocol (RTSP) [34]. This timestamping approach, which is based on a pair of timestamps generated during the transmission and reception of one message, is simpler and faster than that of FTSP [34] which reduces the jitter of interrupt handling through recording, normalizing and averaging multiple timestamps at both the sender and the receiver.

In the single-hop scenario shown in Fig. 4, the head's direct offspring nodes employ their measurement data messages to transmit their timestamps $T1_j$ during the j-th synchronization to the head for maintaining the time synchronization. Using the abundant computing and power resources including the 64-bit floating-point precision at the head, the numerical computational errors analyzed in Appendix. A could be avoided in the proposed scheme.

For modeling the linear relationship between the hardware clocks of the head and the sensor node, the first-order affine clock model [35] is employed; the affine clock model is widely used in many existing WSN time synchronization schemes as a clock model for a relatively short time period, during which the effect of environmental conditions (e.g., temperature) and voltage variation on the clock oscillator hardly changes. Specifically, the hardware clock T_i of a sensor node i for a N-node WSN (i.e., $i \in [0, 1, \ldots, N-1]$) can be described as follows:

$$T_i(t) = (1 + \epsilon_i)\, t + \theta_i, \tag{1}$$

where t is the hardware clock time of the head which is considered as a reference, and $(1+\epsilon_i) \in \mathbb{R}_+$ and $\theta_i \in \mathbb{R}$ are the clock frequency ratio and the clock offset, respectively.

In this regard, the use of the linear regression for clock parameter estimation does make sense in the context of WSNs due to its simplicity, and, in fact, most representative WSN time synchronization schemes, including FTSP and RBS, rely on this method. For a fair comparison with those schemes, we also employ the linear regression method in the proposed scheme to estimate the parameters of sensor nodes' hardware

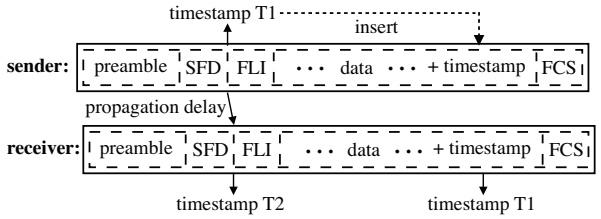

Fig. 5. MAC-layer timestamping [34] adopted in BATS.

clocks. Note that the performance of the linear regression in the proposed scheme, which is running at the head, is not constrained by the limited sample size unlike FTSP using the past 8 samples only due to the resource constraints of sensor nodes.

The following linear least squares method is employed in BATS to model the linear relationship between the hardware clocks of the sensor node i and the head based on m samples during the j-th synchronization: For $i \in [0, 1, \ldots, N-1]$ and $j = m, m+1, \ldots,$

$$\mathbf{\Phi}_i(j) = \left\{\mathbf{T2}_i(j)^\top \mathbf{T2}_i(j)\right\}^{-1} \mathbf{T2}_i(j)^\top \mathbf{T1}_i(j), \tag{2}$$

where

$$\begin{aligned} \mathbf{\Phi}_i(j) &= [1 + \hat{\epsilon}_i(j), \hat{\theta}_i(j)], \\ \mathbf{T1}_i(j) &= [T1_i(j-m+1), ..., T1_i(j)], \\ \mathbf{T2}_i(j) &= [T2_i(j-m+1), ..., T2_i(j)], \end{aligned}$$

where $\mathbf{\Phi}_i(j)$ is a vector of the estimated clock frequency ratio and offset and $(\cdot)^\top$ and $(\cdot)^{-1}$ denote vector transpose and matrix inverse, respectively.

As we will see in Section V, employing the complex linear regression solution with no limitation on the sample size for solving the linear equation of the time synchronization, could improve the performance of time synchronization compared with the conventional simpler solutions described in Section II. Furthermore, thanks to the reverse asymmetric framework that BATS is based on, we can use a more complex estimation technique with higher computational complexity (e.g., machine learning [36]) as all the estimation procedures now run at the head with abundant computing capability instead of the resource-constrained sensor nodes.

As exhibited in Fig. 4, the direction of the conventional one-way message dissemination is reversed, where the estimation of the hardware clocks between sensor node and the head is also reversed compared to the conventional one. On the one hand, when timestamps along with measurement data from sensor nodes are gathered in the head, the corresponding hardware clock timestamps of the sensor nodes should be translated based on the head's reference clock to estimate the exact time for the measurement events. On the other hand, when the head issues commands to notify sensor nodes to perform collaborative operations (e.g., sleep and wake-up for energy-efficient MAC protocols and application of coherent sampling), the timestamps carried in the commands have to be translated to the target sensor nodes' hardware clock time. Using the estimated frequency ratio $1+\epsilon_i$ and offset θ_i between a sensor node i and the head, in the single-hop scenario,

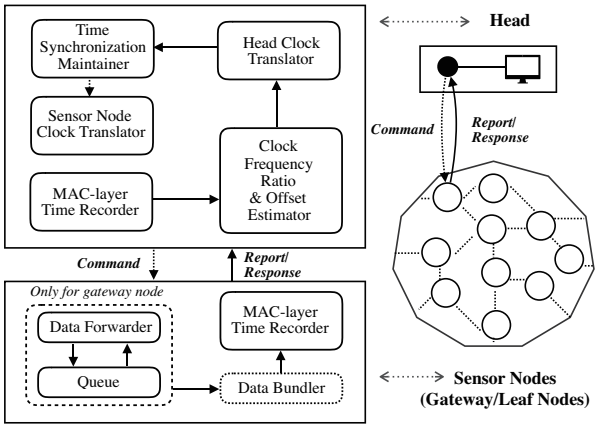

Fig. 6. A system architecture of the proposed time synchronization scheme.

the translation between their hardware clock times is done as follows:

$$t = \frac{T_i(t) - \theta_i}{1 + \epsilon_i} \iff T_i(t) = (1 + \epsilon_i)t + \theta_i, \tag{3}$$

where the translation from sensor node's clock time $T_i(t)$ to head's clock time t involved the floating-point division, which is similar to the clock time translation in EE-ASCFR. As mentioned in [37], such calculation involved with floating-point division and microsecond-level timestamp should be carefully handled in the resource-constrained sensor nodes, but those are not major issues at all in BATS.

IV. Extension of BATS to Multi-Hop WSNs

Independent of the network topologies which are typically established through routing protocols (e.g., Collection Tree Protocol (CTP) [38]), the proposed BATS scheme in principle could be extended to any existing multi-hop routing protocols with the addition of the required timestamps in the application messages. There are several issues to consider, however, in its extension to the multi-hop scenario.

In the multi-hop scenario, gateway sensor nodes located in intermediate layers process not only their own data as regular sensor nodes but also the data from their offspring sensor nodes. The presence of gateway nodes makes it complicated for the head to directly handle all timestamps required for time synchronization in multi-hop WSNs. Note that in practice, the role of a sensor node is not fixed but can be changed to either a gateway node or a leaf node—i.e., a node without relaying packets from other nodes—depending on the topology and the arrangement by the routing protocol.

A. Multi-Hop Extension of The Time Synchronization Based On Reverse One-Way Message Dissemination

As described in [20], two possible approaches for the multi-hop extension of WSN time synchronization schemes based on the reverse two-way message exchange (including EE-ASCFR) are those of *time-relaying* and *time-translation* at the gateway nodes. Of the two approaches, the time-relaying one could introduce more random delays, including queueing

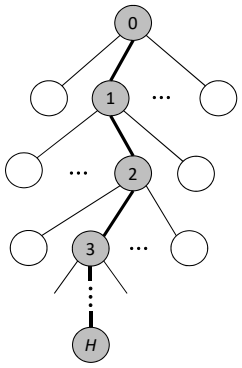

Fig. 7. H-hop end-to-end path in a multi-hop WSN.

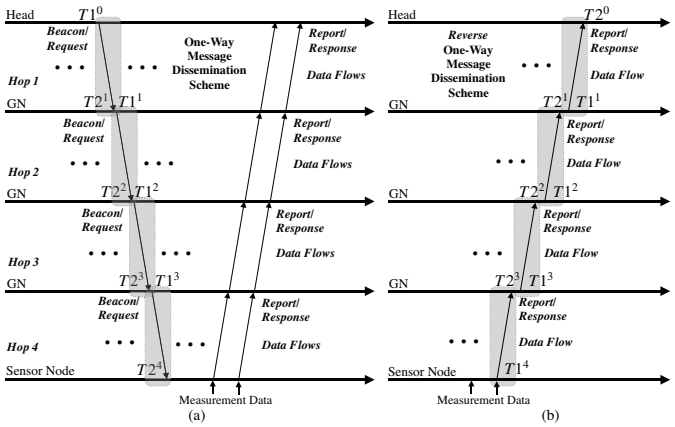

Fig. 8. Multi-hop extension of (a) a conventional (e.g., FTSP) and (b) a reverse asymmetric time synchronization scheme based on the one-way message dissemination with all-data bundling procedure.

delays, as the messages from the sensor nodes are being forwarded to the head through multiple gateway nodes, which are not properly compensated for unlike the time-translation one. To maximize the advantage of MAC-layer timestamping and avoid random queueing delays cumulated through multiple gateway nodes, therefore, we use *per-hop time synchronization* for the multi-hop extension of BATS, which is similar to the aforementioned time-translation approach in which the hardware clock time of a sensor node goes through layer-by-layer translation in order to estimate its time with respect to the reference clock of the head. The fundamental difference of the per-hop time synchronization employed in BATS compared to the time-translation approach, however, is that all layer-by-layer translations are again moved from the gateway nodes to the head to relieve the burden of the gateway nodes under the original time-translation approach. The system architecture is illustrated in Fig. 6, where the "command" is optional and the "sensor nodes" refer to both gateway and leaf sensor nodes.

In the following, we focus on one end-to-end path shown in Fig. 7 for ease of description and notation simplicity; in this case, the index of a node becomes its layer index as well, with 0 indicating both the head and layer 0 it belongs to.

As in (1), we use the affine clock model to describe the relation between the hardware clocks of two neighbor nodes: For i=1, 2, . . . , H,

$$T_i(t) = \left(1 + \epsilon_{i,i-1}\right) T_{i-1}(t) + \theta_{i,i-1}, \tag{4}$$

where $\left(1+\epsilon_{i,i-1}\right) \in \mathbb{R}_+$ and $\theta_{i,i-1} \in \mathbb{R}$ are the clock frequency ratio and the clock offset of $T_i(t)$ with respect to $T_{i-1}(t)$, respectively, and $T_0(t)$=t. Applying the linear least squares method of (2) to (4), we can estimate clock frequency ratios and offsets recursively.

To translate a hardware clock time to the reference clock time, we can use the following equation, again recursively: For i=$H, H-1 \ldots, 1$,

$$T_{i-1}(t) = \frac{T_i(t) - \theta_{i,i-1}}{1 + \epsilon_{i,i-1}}. \tag{5}$$

Note that, in the proposed per-hop time synchronization, timestamps required for the synchronization of a sensor node based on the reverse one-way message dissemination are recorded at the sensor node and its gateway node (working as a reference) as shown in Fig. 4. Unlike the time translation approach described in [20], however, the time translation operation is moved from the gateway to the head, and the gateway node just sends the pair of timestamps obtained from the reverse one-way message dissemination (i.e., $T1_i$ and $T2_{i-1}$) all the way to the head. Based on these pairs of timestamps from gateway nodes, the head can establish the relationships between sensor nodes and their gateway nodes and eventually translate sensor nodes hardware clock times to those based on the reference clock at the head recursively based on (4) and (5).

Using the aforementioned per-hop time synchronization, BATS could be extended to cover the multi-hop scenario as demonstrated in Fig. 8 (b), which is different from the conventional multi-hop scheme based on one-way message dissemination shown in Fig. 8 (a) where the synchronization timestamps are carried by standalone beacon messages and the measurement data are transported by the standalone measurement messages. The proposed BATS does not rely on broadcasting beacon messages and embeds the synchronization timestamps into the measurement messages as in [20]. In case of the conventional schemes, there are two separate flows of messages with different directions (i.e., one for synchronization and the other for measurement), we cannot embed synchronization timestamps into the measurement messages as in the extended BATS. Pseudocodes for the details of the event handling at a sensor node under BATS with and without bundling are provided in Appendix B.

B. Comparison to Other Multi-Hop WSN Time Synchronization Schemes

Having discussed the multi-hop extension of BATS, which does not rely on beacon messages but exploits the bundling of synchronization timestamps with the measurements, we compare the extended BATS to other multi-hop WSN time synchronization schemes.

Since the synchronization timestamps are bundled with measurement data in the same message, the major cost of the proposed scheme is the increase of the payload of the measurement message by the synchronization timestamps $T1$

and $T2$. As shown in Fig. 8 (b), for a node j hops away from the head, 2 timestamps—i.e., $T1^j$ and $T2^{j-1}$—are required for the head to achieve network-wide time synchronization. Therefore, for a flat H-hop network, $2H-1$ timestamps[2] are to be transmitted through the measurement messages to the head. Note that, though the same pair of timestamps (i.e., $T1$ and $T2$) are used in the conventional schemes based on one-way message dissemination such as FTSP, the cost of the transmissions of the standalone synchronization messages would be much higher; even considering that the timestamp $T2$ could be kept locally in the sensor node to perform the synchronization procedure, for a flat H-hop network, H synchronization messages carrying the timestamp $T1$ have to be broadcasted for achieving network-wide time synchronization.

Using the self-data and all-data bundling procedures detailed in Appendix B with the payload and data structure of a message shown in Fig. 9, for a flat H-hop network with each node generating M measurements, the number of message transmissions and receptions for conventional (N_{msg}^{conv}) and proposed (N_{msg}^{prop}) scheme can be obtained as follows:

$$N_{msg}^{conv} = 2(H-1) + 1 + M\sum_{i=1}^{H}\Big(2(i-1)+1\Big), \tag{6}$$

$$N_{msg}^{prop} = \begin{cases} \sum_{i=1}^{H}(2(i-1)+1), & \text{self-data bundling,} \\ 2(H-1)+1, & \text{all-data bundling.} \end{cases} \tag{7}$$

Note that N_{msg}^{conv} includes not only synchronization messages but also measurement messages, while N_{msg}^{prop} just includes measurement messages with synchronization timestamps embedded. For a flat 4-hop network generating 2 measurements per node, the number of message transmissions and receptions for the conventional scheme and the proposed scheme with self-data and all-data bundling procedures are 39, 16 and 7, respectively. These results suggest that the proposed scheme with all-data bundling could reduce more than 80% message transmissions and receptions compared to the conventional one. Consequently, the proposed scheme could achieve much higher energy efficiency.

Compared to the multi-hop extension of the time synchronization scheme based on reverse two-way message exchange proposed in [21], the multi-hop extension of BATS based on one-way message dissemination can greatly lower communication overheads by reducing the number of synchronization messages required for network-wide synchronization, which is a significant advantage when used for large-scale WSNs. If the end-to-end communication range of a multi-hop WSN is over a kilometer, however, the propagation delay cannot be ignored any longer, and the time synchronization schemes based on reverse two-way messages exchange would be a better option in such a case.

C. Discussions

Though the proposed scheme could save the energy consumption by reducing the number of message transmissions and receptions and significantly lower the computational complexity of sensor nodes, those advantages come at the expense of the increase in the computational complexity of the head. The increase in the computational complexity of the head, however, is intentional because, as discussed in Sec. I, the head in an asymmetric WSN is assumed to have plenty of computing and power resources. Note that the computation offloading for sensor nodes in the proposed framework could be extended to the head as well by exploiting mobile edge and fog computing as discussed in [31].

One major issue encountered in large-scale deployments is the additional message transmissions in bundling: Because of the reassignment of the time synchronization procedure from sensor nodes to the head, timestamps for time synchronization have to be delivered to the head. As the number of hops and sensor nodes increases, the number of timestamps to be delivered to the head also increases; the increasing number of timestamps, by the way, does not directly result in more message transmissions thanks to the proposed bundling procedure. Because the maximum number of bundled data is limited by the maximum payload size of a message, however, a gateway node has to generate additional messages when the measurement and/or synchronization data either generated by themselves or received from its offspring nodes cannot be bundled in one message, which reduces the energy efficiency of the proposed time synchronization scheme in large-scale deployments.

Due to the use of computation offloading, BATS also faces the well-known trade-off between energy consumption and execution delay [31]: Bundling a large number of data would increase the end-to-end delay while saving the energy by reducing the number of message transmissions. In this regard, we have proposed an optimal bundling scheme to address this trade-off [39].

V. Experimental Results

The proposed BATS scheme is implemented on a real WSN testbed consisting of one head and six sensor nodes as exhibited in Fig. 10, all of which are based on TelosB motes running TinyOS [40]. Note that the timer resolution of TelosB motes running TinyOS is 1 µs since the resolutions of its hardware clock running on a 32-kHz crystal oscillator and software timer are 30.5 µs and 1 µs, respectively. The accuracy of time synchronization schemes tested on the testbed, therefore, is limited to a microsecond level.

A. Energy Efficiency

1) The Number of Message Transmissions/Receptions: First, we count the numbers of message transmissions and receptions of the conventional and reverse asymmetric time synchronization schemes based on both one-way message dissemination and two-way message exchange to indirectly compare their energy consumptions. A single-hop scenario is considered, where the head and one sensor node are directly connected to each other. We assume that there are 100 measurements in total at the sensor node over the period of 3600 s, which are reported to the head through measurement

[2]Because $T2^0$ is recorded at the head, there is no need of transmission.

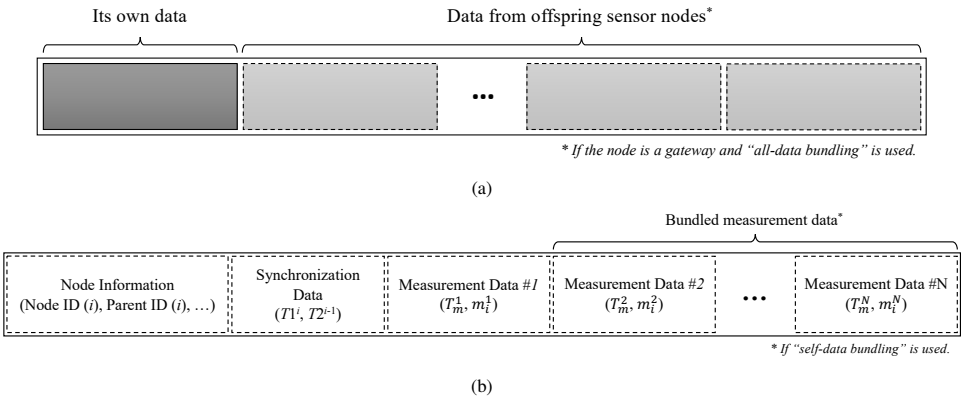

Fig. 9. Payload and data structure of a message generated at sensor node i: (a) Payload with optional "all-data bundling" and (b) data structure with optional "self-data bundling".

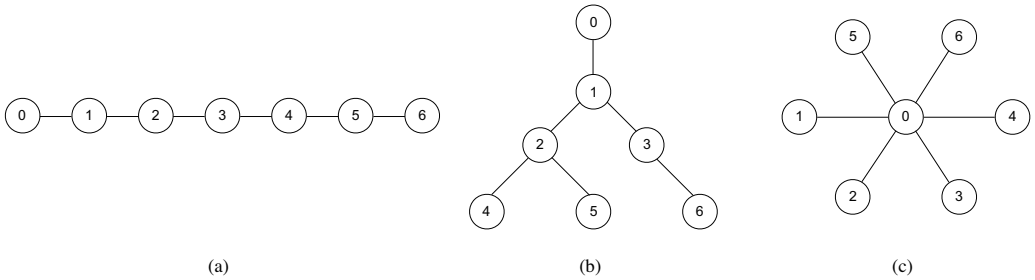

Fig. 10. Multi-hop and single-hop topologies with one head and six sensor nodes: (a) multi-hop chain topology, (b) multi-hop tree topology and (c) single-hop star topology.

TABLE I
THE NUMBERS OF MESSAGE TRANSMISSIONS AND RECEPTIONS AT SENSOR NODE FOR DIFFERENT SIS DURING THE PERIOD OF 3600 S

Type of Synchronization Scheme		N_{TX}[1]	N_{RX}
Conventional Two-way	SI = 100 s	136	36
	SI = 10 s	460	360
	SI = 1 s	3700	3600
Conventional One-way	SI = 100 s	100	36
	SI = 10 s	100	360
	SI = 1 s	100	3600
Reverse Two-way	SI = 100 s	100	36
	SI = 10 s	100	360
	SI = 1 s	100	3600
Reverse One-way	SI = 100 s	100	0
	SI = 10 s	100	0
	SI = 1 s	100	0

[1] Both synchronization and measurement messages are counted.

messages without measurement bundling, and the SI is set to 1 s. The resulting number of messages—i.e., synchronization and measurement messages—transmitted (N_{TX}) and received (N_{RX}) by the sensor node are summarized in Table I. Note that the types listed in Table I categorize *the ways of message exchange in time synchronization* and are not specific to certain time synchronization schemes; examples of the "Conventional Two-way", "Conventional One-way", and "Reverse Two-way" types are TPSN, FTSP, and EE-ASCFR, respectively, while BATS belongs to the "Reverse One-way" type.

From the comparison, we observe that the reverse one-way scheme could save the energy consumed by both transmissions and receptions of the synchronization messages; this is because the reverse one-way scheme is free from transmissions and receptions of beacon messages and synchronization timestamps are embedded into measurement messages.

In the reverse two-way scheme, the "Request" synchronization message including timestamp $T1$ can be embedded into beacon messages, but still sensor nodes consume energy to receive the beacon messages. On the other hand, the conventional two-way scheme requires the most message transmissions due to the sending the "Request" messages from the sensor node to the head. In addition, the conventional one-way scheme requires the same number of synchronization message receptions as in the reverse two-way scheme.

2) Direct Measurement of Energy Consumption: To measure the actual power consumption of a time synchronization scheme on the resource-constrained sensor nodes, we employ a stabilized voltage supply and a digital storage oscilloscope (DSO) to power the sensor node which synchronizes with the head and log the actual power consumption as shown in Fig. 11. In the experiment setup, one 1 Ω resistor is connected in series between the power supply and the sensor node. The voltage of the power supply is set to 3.3 V, a slightly higher voltage than that received from the battery holder of the TelosB sensor board [24], to provide the sensor node sufficient power. The two pins of the resistor are connected

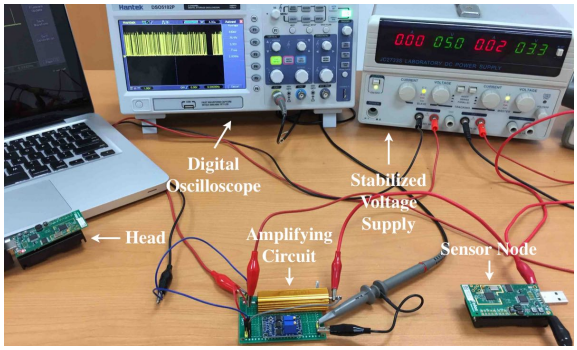

Fig. 11. Experiment setup for the measurement of the energy consumption on a WSN sensor node.

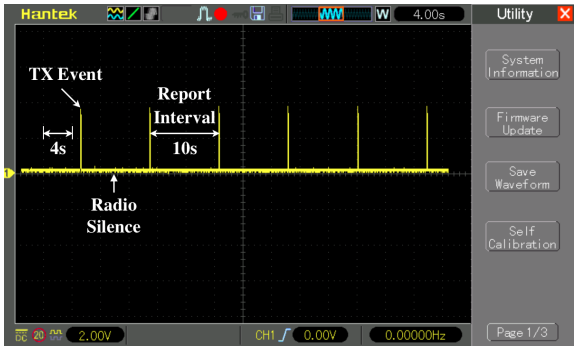

Fig. 12. Measuring and logging the energy consumption using DSO.

to the inputs of the amplifying circuit[3] which used to enlarge the mA-level current input value with 150 times. The outputs of the amplifying circuit are connected to the DSO for logging the power consumption. The power consumption logged in the DSO is illustrated in Fig.12. Because the voltage is stabilized in the measurement experiment which is $3.3V$ constantly, so the actual power consumption could be compared through comparing the logged current value sets.

During the experiments, the measurement is generated every 2 s and the bundling procedure with bundling 5 measurements is employed in both FTSP and BATS for fair comparison. FTSP broadcasts and receives the synchronization beacon messages per second (i.e., default setting) and reports the bundling messages every 10 s, BATS reports the bundling messages with the measurements and synchronization data every 10 s without broadcasting or receiving—i.e., radio listening deactivated—the synchronization beacons. The power and energy consumptions are computed as follows:

$$P(t) = V(t) \times I(t), \tag{8}$$

$$E = \int_{t_s}^{t_e} P(t)dt, \tag{9}$$

where $P(t)$, $V(t)$ and $I(t)$ are the instant power, voltage and current for the sensor node at time t, respectively, and E is the energy consumed by the sensor node over the time period of $[t_s, t_e]$. Because the voltage $V(t)$ is fixed to 3.3 V during the experiments, we only measure the current $I(t)$ over the two

[3]The amplifying circuit is based on the analog devices designer guidebook [41, chapter 4].

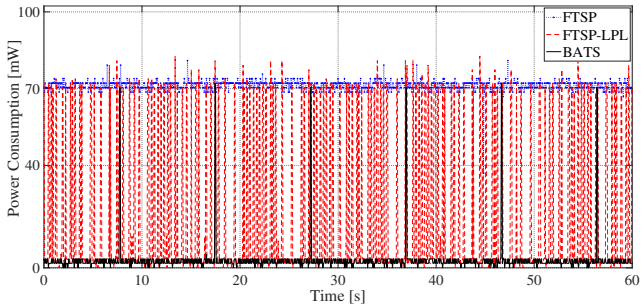

Fig. 13. Power consumptions of different time synchronization schemes over 60 s.

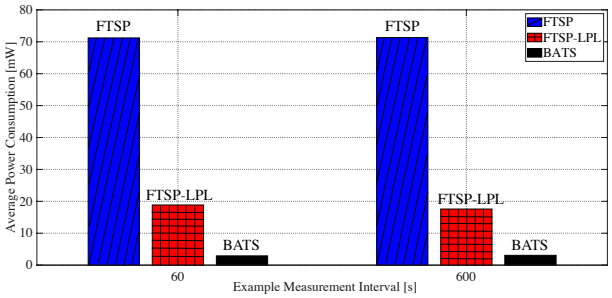

Fig. 14. Average power consumptions of time synchronization schemes over the measurement interval of 60 s and 600 s.

different time intervals of 60 s and 600 s (i.e., $t_e - t_s$ in (9)) to obtain power and energy consumptions; Fig. 13 shows the power consumption for different time synchronization schemes over the period of 60 s. Since the broadcasting of beacon messages and the reporting of the (bundled) measurement message are periodic, so the two aforementioned example measurement intervals could represent the long-term experiments. As illustrated in Fig. 14, the average power consumption for BATS is the lowest, while the original FTSP—i.e., without using low-power mode—consumes the most power. Specifically, the FTSP employing low-power mode consumes less than one-third power of the original FTSP but more than BATS, which consumes less than 5% and 16% power of the original and the low-power FTSP, respectively.

B. Time Synchronization Accuracy

1) Single-Hop Scenario: We first evaluate the time synchronization accuracy of the proposed BATS scheme in comparison to that of FTSP with a single-hop scenario such as the star topology illustrated in Fig. 10 (c). For the experiments, we assume that the sensor node periodically sends measurement data to the head via measurement messages, where 5 measurements are bundled in each measurement message for both FTSP and BATS together with synchronization data in case of the latter. We run experiments for 3600 s.

Note that BATS uses the measurement messages to carry both measurement and synchronization data, while conventional one-way time synchronization schemes like FTSP rely on beacon messages for synchronization data and use measurement messages only for measurement data. For a fair comparison between BATS and FTSP, therefore, we define

SI as the interval between two consecutive measurement messages carrying synchronization data for BATS and the interval between two consecutive beacon messages for FTSP, respectively, and use the same SI value for both schemes for each experiment.

Before carrying out a comparative analysis with FTSP, we need to decide the values of BATS system parameters affecting time synchronization performance. During the preliminary experiments, we found that the time synchronization accuracy of BATS is affected by the sample size for the linear regression for clock frequency ratio estimation. Note that, for a reasonable comparison, to find the best value of the sample sizes for different SIs, we implemented the offline re-running program in which the same raw data trace produced in the online experiment could be reused to run different offline experiments with different sample sizes. With this program, the possible sample sizes (e.g., 2, ..., 10, ..., 100, ..., all) for different values of SI are evaluated. In particular, the performance of the experiment based on all samples does not outperform the one using the most recent timestamps with a certain sample size value. This may result from, as time goes on, due to the aggravating clock drift, the very past sample data—i.e., very past timestamps—do not have positive contributions for the estimations of the timestamps in the most current synchronization interval. Instead, the most recent samples could provide relatively better contributions to the estimation of the most recent timestamps.

Fig. 15 shows the effect of sample size on the mean square error (MSE) and the mean absolute error (MAE) of measurement time estimation of BATS with different values of SI. From the results, we find that the MSE and MAE of measurement time estimation are minimal when the sample size is 19 and 5 for SI of 1 s and 10 s, respectively. As for the experiments with SI of 100 s, the size of total samples is quite limited (i.e., only 36 samples from the experiment over 3600 s), which means, it is quite difficult to apply large value for the aforementioned sample size. Anyhow, we still tuned the value of sample size in the range of 2–30, and the results showed that, 2 is the best sample size for the experiment with SI of 100 s.

With the best empirical values of the sample sizes for various values of SI, the performance of our proposed scheme embedding the linear regression method is demonstrated. The performance of the method 2 of linear regression with the best parameter value outperformed the conventional methods such as the method 1 based on the calculation of the cumulative frequency ratio proposed in [27] as shown in Fig. 16. In this figure, the measurement time estimation errors of our proposed time synchronization scheme based on both traditional ratio-based method and linear regression method are demonstrated, in which the linear regression method outperforms the ratio-based method in all three report intervals at different levels. In addition, Fig. 16 demonstrated that our proposed reverse asymmetric framework could be employed on different conventional time synchronization schemes with diverse estimation methods.

Table II summarizes the MAE and MSE of measurement time estimation from the experiments during the period of

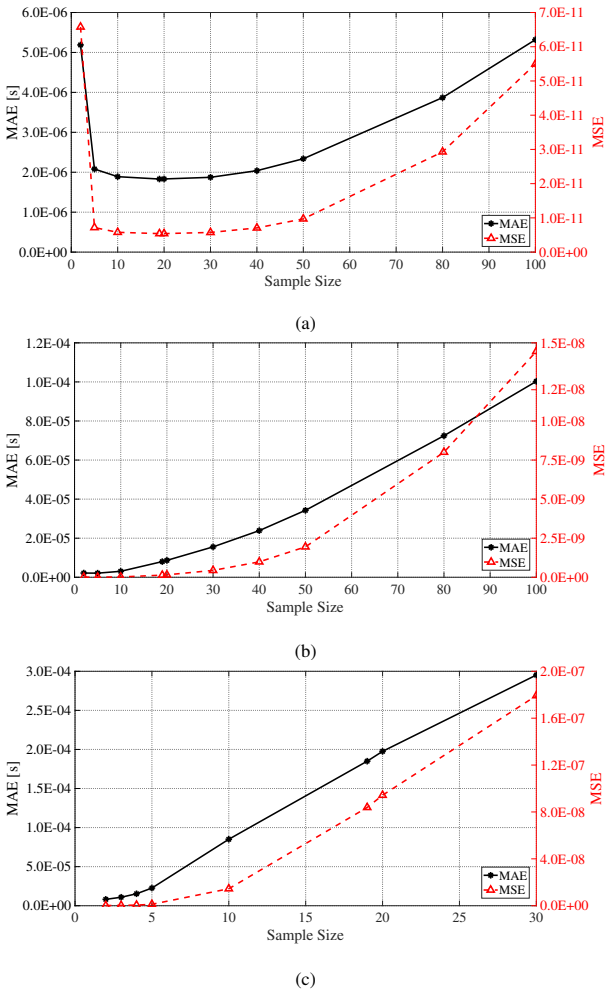

Fig. 15. The effect of sample size on the measurement time estimation of BATS: (a) SI=1 s; (b) SI=10 s and (c) SI=100 s.

3600 s, where "method 1" and "method 2" denote the ratio-based and the linear regression estimation methods, respectively, as in Fig. 16. In the single-hop scenario, one sensor node is synchronized to one head in the experiments. Note that, as the standard FTSP implementation provided in TinyOS library is employed in our experiments, so the synchronization accuracy of FTSP is limited to millisecond-level [42]. The MAE of measurement time estimation of all three different SIs show that the proposed scheme with two proposed methods provides satisfactory precision with minimum of 1.8299 µs which is competitive with consideration to the results presented in other papers—e.g., conventional two-way scheme TPSN and one-way schemes FTSP and RSP—which are also evaluated through real testbeds, however, with drastically fewer message transmissions as shown in Table II.

In addition, Fig. 16 and Table II illustrate that a relatively smaller SI results in better performance with smaller MSE, which means, the performance of the proposed scheme with both methods is related to the value of report intervals. This may result from the drifting of the low-cost crystal oscillator with up to tens of ppm of clock skew in the resource-constrained sensor node, in which the drifting range

TABLE II
MAE AND MSE OF MEASUREMENT TIME ESTIMATION OF FTSP AND BATS FOR THE SINGLE-HOP SCENARIO

Synchronization Scheme		MAE[1][s]	MSE[1]	N_{TX}	N_{RX}
FTSP[2]	SI = 100 s	0.2892E-03	0.3614E-06	36	36
	SI = 10 s	0.3164E-03	0.3983E-06	360	360
	SI = 1 s	0.3173E-03	0.4038E-06	3600	3600
BATS with the method 1	SI = 100 s	2.4837E-05	2.1194E-09	36	0
	SI = 10 s	3.2770E-06	1.7492E-11	360	0
	SI = 1 s	2.4903E-06	1.0120E-11	3600	0
BATS with the method 2	SI = 100 s	8.1524E-06	1.5805E-10	36	0
	SI = 10 s	2.1016E-06	7.3933E-12	360	0
	SI = 1 s	1.8299E-06	5.4018E-12	3600	0

[1] Based on the measurement time estimation obtained from 3600 s such that the actual performance in real deployment is represented.
[2] The standard FTSP implementation provided in TinyOS library offers limited millisecond-level time synchronization [42].

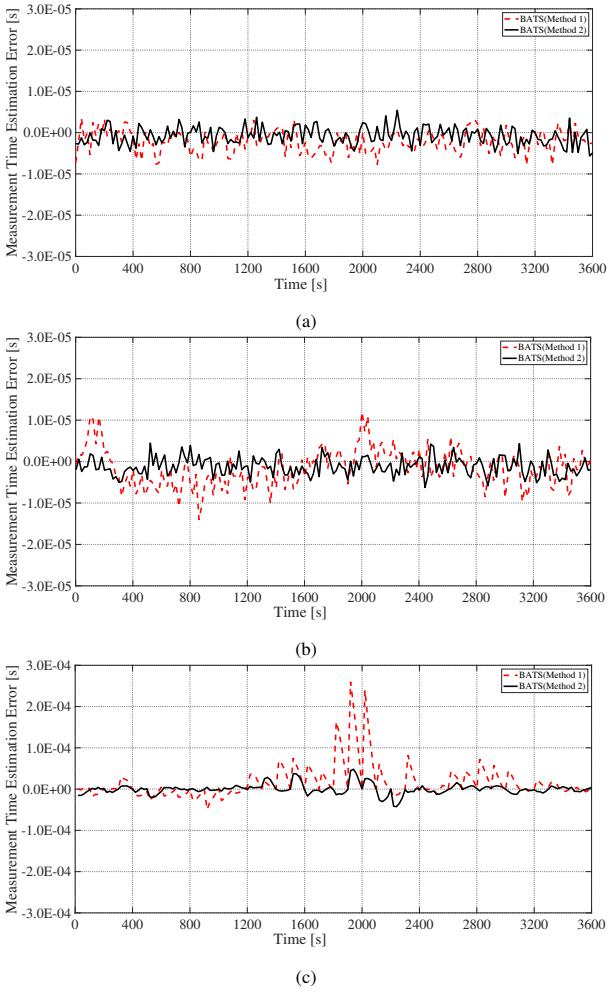

Fig. 16. Measurement time estimation errors of BATS with ratio-based and linear regression methods with SI of (a) 1 s, (b) 10 s, and (c) 100 s.

is typically larger in a longer interval.

2) Multi-Hop Scenario: Here we investigate the effect of the number of hops on time synchronization with a 6-hop chain topology as exhibited in Fig. 10 (a). We set the SI to 1 s and employ the optimal sample size of 19 for the experiment. Note that we apply the self-data bundling only in order to mainly focus on the effect of the number of hops, rather than that of bundling, and thereby make the results more consistent with

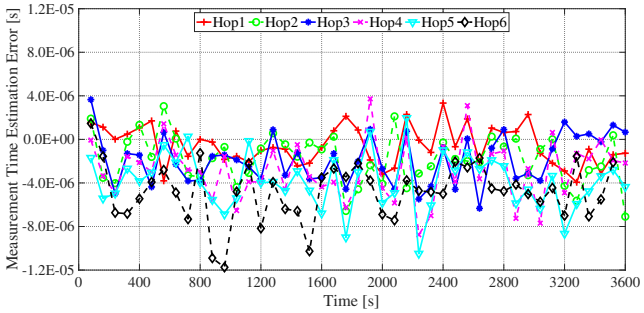

Fig. 17. Measurement time estimation errors of BATS for the multi-hop scenario.

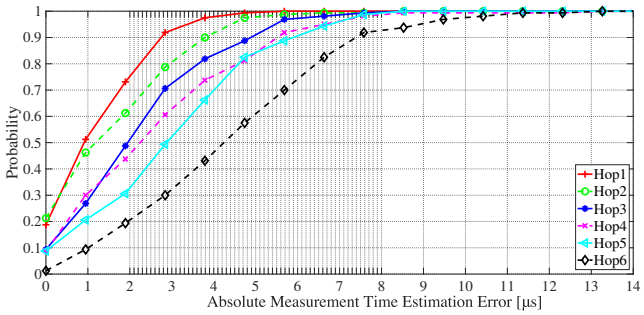

Fig. 18. Cumulative distribution functions of the absolute measurement time estimation errors of BATS for the multi-hop scenario.

those of conventional schemes reported in the literature.

As shown in Fig. 17, the level of fluctuations of the measurement time estimation errors are roughly proportional to the hop counts; for instance, the measurement time estimation errors of the node 6 hops away from the head show the highest fluctuations, while those of the node 1 hop away from the head show the least fluctuations. This is due to the per-hop synchronization strategy employed in BATS, where the estimation of the hardware clock time of a sensor node with respect to the reference clock relies on those of its upper-layer sensor nodes.

Fig. 18 shows the effect of the number of hops on time synchronization in a clearer way through the cumulative distribution functions (CDFs) of *absolute measurement time estimation errors*. 90th-percentile absolute measurement time estimation errors for sensor nodes 1 to 6 hops away from the head are 2.8 μs, 3.8 μs, 4.9 μs, 5.5 μs, 5.9 μs and 7.4 μs,

TABLE III
MAE AND MSE OF MEASUREMENT TIME ESTIMATION OF BATS FOR THE MULTI-HOP SCENARIO

Sensor Node		MAE(Chain)[1]	MSE(Chain)[1]	MAE(Tree)[1]	MSE(Tree)[1]
ID	6	4.2580E-06	2.4586E-11	2.6301E-06	1.0573E-11
	5	3.6149E-06	1.8405E-11	2.5140E-06	1.0240E-11
	4	3.1341E-06	1.4519E-11	2.7873E-06	1.1414E-11
	3	2.4847E-06	9.4813E-12	2.0985E-06	7.0036E-12
	2	1.9455E-06	6.0164E-12	2.1368E-06	7.1055E-12
	1	1.6764E-06	4.4735E-12	1.6932E-06	4.5415E-12

[1] Based on the measurement time estimation obtained from 3600 s such that the actual performance in real deployment is represented.

respectively. The MAEs and MSEs of measurement time estimation are also summarized as MAE(Chain) and MSE(Chain) in Table III. The results of Fig. 18 and Table III demonstrate that the proposed scheme can provide microsecond-level time synchronization accuracy for all the sensor nodes in the 6-hop WSN, even though the time synchronization error is cumulative over the hop count. In addition, we established one comparison experiment with the tree topology illustrated in Fig. 10 (b) to demonstrate the stable performance of BATS under various topologies. The results are summarized as MAE(Tree) and MSE(Tree) in Table III.

The practical evaluation results in Section V-A and Section V-B have jointly proved that the proposed scheme could drastically conserve more energy consumptions while maintaining the same level of synchronization accuracy compared to the conventional schemes.

VI. Concluding Remarks

We have proposed BATS, i.e., an energy-efficient time synchronization scheme based on the framework of reverse asymmetric time synchronization and the reverse one-way message dissemination; BATS can simultaneously address the major challenges in WSN time synchronization, i.e., lowering energy consumption and computational complexity while achieving high time synchronization accuracy.

The major contribution of our work in this paper is threefold: First, the reverse asymmetric time synchronization framework is presented. This framework reassigns the clock parameter estimation procedures from resource-constrained sensor nodes to the head equipped with abundant computing and power resources, which leaves only timestamping procedure at the sensor nodes. This reassignment of time synchronization can not only reduce the computational errors caused by the limited precision floating-point arithmetic at the sensor nodes but also bring further potential to use more complex estimation methods at the head, e.g., those based on machine learning techniques such neural networks as demonstrated in [36]; note that, the proposed framework could be employed by the conventional time synchronization schemes based on both one-way message dissemination and two-way message exchange, too.

Second, based on the reverse asymmetric time synchronization framework, BATS is proposed and extended to multi-hop WSNs, which significantly reduces the energy consumption by eliminating the need of the extra synchronization-related message—i.e., beacon message and standalone synchronization message—transmissions.

Third, the actual energy consumption and time synchronization accuracy of BATS are evaluated through extensive experiments on the real WSN testbed. The results demonstrate that BATS conserves up to 95% of the energy consumption by FTSP and provides 1.8299 µs synchronization accuracy in the single-hop scenario. In case of the multi-hop scenario, the synchronization accuracy for 1-hop and 6-hop sensor nodes are 1.6764 µs and 4.2580 µs respectively, which results in 0.5163 µs per-hop synchronization error in average.

As part of BATS, we have also outlined the message bundling procedure, the full investigation of which would require the consideration of its effect on the end-to-end delay of the measurement data as well as the time synchronization performance. In the follow-up work, therefore, we will carry out a systematic investigation of the effects of the bundling procedure on both synchronization and delay performance in order to identify potential issues and address them through more advanced bundling procedures.

Note that the asymmetric time synchronization framework presented in this paper fits the asymmetric IoT deployment which typically consists of a powerful server and numerous resource-constrained IoT devices [43].

Appendix A
Impact of Precision Loss on the Performance of One-Way-Message-Dissemination-Based Time Synchronization Schemes

Taking RSP as an example, which was proposed to simplify the estimation of the clock skew and offset in FTSP, we analyze the impact of the precision loss on the performance of the one-way-message-dissemination-based time synchronization schemes.

In case of the head and the sensor node shown in Fig. 2 (a), their hardware clock times (i.e., t for the head and $T_i(t)$ for the sensor node) can be modeled as follows:

$$t = \alpha_i T_i(t) + \beta_i, \tag{10}$$

where α_i and β_i are the clock frequency ratio and the clock offset of the reference clock with respect to the sensor node hardware clock, respectively. Let $T1_k$ and $T2_k$ (k=1, ...) be the timestamps corresponding to the latest Beacon/Request message received at the sensor node. Then, the logical clock $\mathscr{T}_i^{\mathrm{RSP}}$ at the sensor node i—i.e., the estimated reference time corresponding to the sensor node's hardware clock $T_i(t)$—is obtained as follows [27]:

$$\mathscr{T}_i^{\mathrm{RSP}}\Big(T_i(t)\Big) = \hat{\alpha}_{i,k} T_i(t) + \hat{\beta}_{i,k}, \tag{11}$$

where $\hat{\alpha}_{i,k}$ and $\hat{\beta}_{i,k}$ are the clock frequency ratio and the clock offset estimated based on the linear interpolation using four timestamps, i.e.,

$$\hat{\alpha}_{i,k} = \frac{T1_k - T1_{k-1}}{T2_k - T2_{k-1}}, \tag{12}$$

$$\hat{\beta}_{i,k} = \frac{T1_{k-1} \cdot T2_k - T2_{k-1} \cdot T1_k}{T2_k - T2_{k-1}}. \tag{13}$$

The impact of the precision loss on the estimated reference clock of RSP in (11) can be analyzed in a similar way to that for EE-ASCFR [21]: Let ϵ_α and ϵ_β be the precision loss in the estimation of the clock frequency ratio and the clock offset, i.e.,

$$\epsilon_\alpha \triangleq \hat{\alpha}_{i,k} - \hat{\alpha}^{\mathrm{LP}}_{i,k}, \tag{14}$$

$$\epsilon_\beta \triangleq \hat{\beta}_{i,k} - \hat{\beta}^{\mathrm{LP}}_{i,k}, \tag{15}$$

where $\hat{\alpha}^{\mathrm{LP}}_{i,k}$ and $\hat{\beta}^{\mathrm{LP}}_{i,k}$ are the value of the clock frequency ratio and the clock offset from the practical implementation that are affected by the limited-precision floating-point arithmetic in (12) and (13). The computational error $\Psi(t)$ in the estimated reference clock at the sensor node at time t, therefore, can be derived as follows:

$$\begin{aligned} \Psi(t) &= \left(\hat{\alpha}_{i,k} T_i(t) + \hat{\beta}_{i,k}\right) - \left(\hat{\alpha}^{\mathrm{LP}}_{i,k} T_i(t) + \hat{\beta}^{\mathrm{LP}}_{i,k}\right), \\ &= \epsilon_\alpha T_i(t) + \epsilon_\beta. \end{aligned} \tag{16}$$

(16) shows that the computational error consists of two components, the first of which is proportional to the current time of the sensor node's hardware clock (i.e., $T_i(t)$).

According to the IEEE standard for floating-point arithmetic [44], a non-zero floating-point number x can be represented in the binary format as follows:

$$x = \sigma \cdot \bar{b} \cdot 2^e, \tag{17}$$

where σ is the *sign* taking the value of $+1$ or -1, $\bar{b}$ is the *binary fraction* whose value is within the range of $[1, 2)$, and e is the *integer exponent*. Because the IEEE 32-bit single-precision floating-point format assigns 1 bit to σ, 8 bits to e and 23 bits to $\bar{b}$, the *machine epsilon* [45] becomes 2^{-23}. If the rounding arithmetic is chopping (i.e., rounding towards 0), the precision loss ϵ_α is within the range of $[-2^{-23}, 0]$, and the maximum absolute precision loss in this case is $2^{-23} \approx 1.19 \times 10^{-7}$. This implies that the first component of (16) alone could result in about 0.1 µs and 1 µs computational errors for $T_i(t)$ of 1 s and 10 s, respectively, in the worst case.

Note that the analysis of the computational error above is based on the worst-case scenario for simplicity. As discussed in [27], however, in reality the precision loss ϵ_α itself is inversely proportional to SI (i.e., the time difference between two consecutive beacon messages), which could more or less relax the dependency of the computational error $\Psi(t)$ on $T_i(t)$. In fact, setting an optimal value of SI is quite complicated because the value of SI not only affects the computational error but also determines the impact of the sensor node's hardware clock drift due to the changes in the ambient temperature and the battery voltage.

Data: The node maintains the following data and variables:
- e: Event object including a timestamp;
- $mac\text{-}layer\_timestamping$: Boolean variable controlling MAC-layer timestamping
- $node\_status$: Variable indicating the status of node (i.e., GATEWAY or LEAF);
- p: Packet object (optionally) including timestamps from MAC-layer timestamping;
- Q_M: FIFO queue for measurement data;
- Q_P: FIFO queue for packets;
- Q_{T2}: FIFO queue for timestamp $T2$.

```
On detecting an event e:
switch e.type do
    case MEASUREMENT do // its own measurement
        d ← Q_M.dequeue() // measurement data from the queue
        ts ← e.getTimestamp()  // for measurement, not for synchronization
        p ← Packet(d, ts)  // create a packet object
        send(p, mac-layer_timestamping=ON)  // for T1
    case PACKET do
        if p.getDestAddress()≠HEAD then // packet received from other sensor nodes
            if node_status==GATEWAY then
                p ← Q_P.dequeue()  // packet from the queue
                if p.getT2()==NULL then // direct offspring
                    T2 ← Q_T2.dequeue()  // from mac-layer timestamping
                    p.setT2(T2)
                send(p, mac-layer_timestamping=OFF)
                 // no need for another T1
            else
                // Process the packet from the head ...
    otherwise do
        // Process other event ...
```

Algorithm 1: Event handling at a sensor node under BATS without bundling.

Appendix B
Pseudocodes for Event Handling at A Sensor Node under BATS

Algorithms 1 and 2 show pseudocodes for the details of the event handling at a a sensor node—i.e., either a gateway node or a leaf node—under BATS with and without bundling.

Acknowledgment

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

References

[1] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, May 2005.

[2] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[3] Y. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, Jan. 2011.

[4] 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.

Data: The node maintains the following data and variables:
- e: Event object including a timestamp;
- $mac\text{-}layer\_timestamping$: Boolean variable controlling MAC-layer timestamping
- $node\_status$: Variable indicating the status of node (i.e., GATEWAY or LEAF);
- p: Packet object (optionally) including timestamps from MAC-layer timestamping;
- b: Bundling counter for counting the buffered bundles.
- bf: Bundling buffer for carrying packets;
- Q_M: FIFO queue for measurement data;
- Q_P: FIFO queue for packets;
- Q_{T2}: FIFO queue for timestamp $T2$.

On detecting an event e:
switch *e.type* **do**
  case *MEASUREMENT* **do** *// its own measurement*
    $d \leftarrow Q_M$.dequeue() *// measurement data from the queue*
    $ts \leftarrow e$.getTimestamp() *// for measurement, not for synchronization*
    $bf \leftarrow bf$+Packet(d, ts) *// kick packet into bundling buffer*
    $b \leftarrow b$+1 *// update the bundling number*
  case *PACKET* **do**
    if $p.getDestAddress() \neq HEAD$ **then** *// packet received from other sensor nodes*
      if $node\_status == GATEWAY$ **then**
        $p \leftarrow Q_P$.dequeue() *// packet from the queue*
        if p.getT2()==*NULL* **then** *// direct offspring*
          $T2 \leftarrow Q_{T2}$.dequeue() *// from mac-layer timestamping*
          p.setT2($T2$)
        if $SELF\_DATA\_BUNDLING == TRUE$ **then** *// in case of self-data bundling*
          $send(p, mac\text{-}layer\_timestamping = OFF)$ *// no need for another $T1$*
        else *// in case of all-data bundling*
          $bf \leftarrow bf+p$ *// kick packet into bundling buffer*
          $b \leftarrow b+p.bundles$ *// update the bundling number*
      else
        // Process the packet from the head ...
  otherwise do
    // Process other event ...
if $b >= MAX\_BUNDLING$ **then** *// number of bundled data equals or exceeds maximum bundling number*
  send$(bf, mac\text{-}layer_timestamping = ON)$ *// for $T1$*
  $bf \leftarrow NULL$ *// clear bundling buffer*

Algorithm 2: Event handling at a sensor node under BATS with bundling.

[5] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. 5th Symp. Operating System Design and Implementation*, Dec. 2002, pp. 147–163.

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

[7] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "Implementation and evaluation of the reference broadcast infrastructure synchronization protocol," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 801–811, Jun. 2015.

[8] H. Wang, L. Shao, M. Li, B. Wang, and P. Wang, "Estimation of clock skew for time synchronization based on two-way message exchange mechanism in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4755–4765, Nov. 2018.

[9] 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.*, pp. 1–1, 2019.

[10] M. Jin, T. Xing, X. Chen, X. Meng, D. Fang, and Y. He, "Dualsync: Taming clock skew variation for synchronization in low-power wireless networks," in *Proc. IEEE INFOCOM 2016*, April 2016, pp. 1–9.

[11] F. Gong and M. L. Sichitiu, "CESP: A low-power high-accuracy time synchronization protocol," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2387–2396, Apr. 2016.

[12] J. P. B. Nadas, R. D. Souza, M. E. Pellenz, G. Brante, and S. M. Braga, "Energy efficient beacon based synchronization for alarm driven wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 23, no. 3, pp. 336–340, March 2016.

[13] K. S. Yildirim and O. Gurcan, "Efficient time synchronization in a wireless sensor network by adaptive value tracking," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3650–3664, July 2014.

[14] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, S. Son, and G. J. Pappas, "ATPC: Adaptive transmission power control for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 12, no. 1, pp. 6:1–6:31, Mar. 2016.

[15] A. Pal and A. Nasipuri, "Joint power control and routing for rechargeable wireless sensor networks," *IEEE Access*, vol. 7, pp. 123 992–124 007, 2019.

[16] W. Dong, C. Chen, X. Liu, Y. He, Y. Liu, J. Bu, and X. Xu, "Dynamic packet length control in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1172–1181, Mar. 2014.

[17] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[18] A. Al-Shaikhi and A. Masoud, "Efficient, single hop time synchronization protocol for randomly connected WSNs," *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 170–173, April 2017.

[19] B. Zhou and M. C. Vuran, "Cortis: Correlation-based time synchronization in internet of things," in *Proc. 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–7.

[20] 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.

[21] 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.

[22] S. el Khediri, N. Nasri, M. Samet, A. Wei, and A. Kachouri, "Analysis study of time synchronization protocols in wireless sensor networks," *arXiv e-prints*, pp. 1–15, Jun. 2012.

[23] M. V. Barel, G. Heinig, and P. Kravanja, "A superfast method for solving toeplitz linear least squares problems," *Linear Algebra and its Applications*, vol. 366, pp. 441–457, Jun. 2003, special issue on Structured Matrices: Analysis, Algorithms and Applications.

[24] "TelosB datasheet," http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf, accessed: 21-10-2019.

[25] B. Kusý, P. Dutta, P. Levis, M. Maróti, Ákos Lédeczi, and D. Culler, "Elapsed time on arrival; a simple and versatile primitive for canonical time synchronisation services," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 1, no. 4, pp. 239–251, Jul. 2006.

[26] J. Sallai, B. Kusý, A. Lédeczi, and P. Dutta, "On the scalability of routing integrated time synchronization," in *Proc. of the Third European Conference on Wireless Sensor Networks*, ser. EWSN'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 115–131.

[27] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Ratio-based time synchronization protocol in wireless sensor networks," *Telecommunication Systems*, vol. 39, no. 1, pp. 25–35, Sep. 2008.

[28] "MicaZ datasheet," http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf, accessed: 21-10-2019.

[29] "Arduino platforms," https://www.arduino.cc/, accessed: 21-10-2019.

[30] S. K. Mani, R. Durairajan, P. Barford, and J. Sommers, "An architecture for IoT clock synchronization," in *Proc. of the 8th International Conference on the Internet of Things*, Oct. 2018, pp. 17:1–17:8.

[31] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, third quarter 2017.

[32] D. Han, B. Bai, and W. Chen, "Outage bottleneck for reliable mobile computation offloading: Transmission or computation?" in *Proc. 2016 IEEE GlobalSIP*, Dec 2016, pp. 465–469.

[33] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[34] 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.

[35] R. T. Rajan and A.-J. van der Veen, "Joint ranging and clock synchronization for a wireless network," in *Proc. CAMSAP 2011*, Dec. 2011, pp. 297–300.

[36] G. Cena, S. Scanzio, and A. Valenzano, "A neural network clock discipline algorithm for the rbis clock synchronization protocol," in *Proc. 14th IEEE International Workshop on Factory Communication Systems*, Jun. 2018, pp. 1–10.

[37] D. Djenouri and M. Bagaa, "Implementation of high precision synchronization protocols in wireless sensor networks," in *Proc. WOCC 2014*, May 2014, pp. 1–6.

[38] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis, "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 10, no. 1, pp. 16:1–16:49, Dec. 2013.

[39] X. Huan, K. S. Kim, S. Lee, and M. K. Kim, "Optimal message bundling with delay and synchronization constraints in wireless sensor networks," *Sensors*, vol. 19, no. 18, Sep. 2019.

[40] "TinyOS," https://github.com/tinyos/tinyos-main, accessed: 21-10-2019.

[41] "A Designer's Guide to Instrumentation Amplifiers, 3RD Edition, 2006," https://www.analog.com/en/education/education-library/dh-designers-guide-to-instrumentation-amps.html, accessed: 21-10-2019.

[42] J. M. Castillo-Secilla, J. M. Palomares, and J. Olivares, "Temperature-compensated clock skew adjustment," *Sensors*, vol. 13, no. 8, pp. 10 981–11 006, Aug. 2013.

[43] B. Badihi, R. Jantti, Huseyin, and Yigitler, "Time synchronization for IoT deployments: Clock discipline algorithms and protocols," https://www.researchgate.net/publication/328577639_Time_Synchronization_for_IoT_Deployments_Clock_Discipline_Algorithms_and_Protocols, 2018, accessed: 21-10-2019.

[44] IEEE Computer Society, *IEEE Std 754^{TM}-2008, IEEE Standard for floating-point arithmetic*, Std., 2008.

[45] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys*, vol. 23, no. 1, pp. 5–48, Mar. 1991.

Xintao Huan received the B.Sc. and M.Sc. degrees in computer engineering from the University of Duisburg-Essen, Germany, in 2013 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Electronics, University of Liverpool, UK, and the Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, China. He was a Research Assistant with the Networked Embedded Systems Group, University of Duisburg-Essen, from 2012 to 2016. His research interests include wireless sensor networks and Internet of Things.

Kyeong Soo Kim (S'89-M'97-SM'19) received PhD degree in Electronics Engineering from Seoul National University, Korea, in 1995, and has been working as an associate professor of Electrical and Electronic Engineering at Xi'an Jiaotong-Liverpool University, China, since 2014. From 1996 to 1997, he was engaged in the development of multi-channel asynchronous transfer mode (ATM) switching systems as a Post-Doc researcher at Washington University in St. Louis, Missouri. From 1997 to 2000, he worked with the passive optical network (PON) Systems R&D organization of Lucent Technologies and was involved with development of ATM-PON systems, which won 1999 Bell Labs President's Silver Award. From 2001 to 2007, he was with STMicroelectronics, working as a Principal Engineer; during this period, he also took the position of STMicroelectronics Researcher-in-Residence at the Stanford Networking Research Center. From 2007 to 2014, he worked at Swansea University, U.K., as an associate professor. Dr. Kim is a senior member of IEEE and a member of IET.

Sanghyuk Lee received Doctorate degree from Seoul National University, Seoul, Korea, in Electrical Engineering in 1998. His main research interests include data evaluation with similarity measure, human signal analysis, high dimensional data analysis, controller design for linear/nonlinear system, and observer design for linear/nonlinear system. Dr. Lee is currently working as an Associate Professor at the Department of Electrical and Electronic Engineering of Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China, which he joined in 2011. He has also been working as a founding director of the Centre for Smart Grid and Information Convergence (CeSGIC) in XJTLU since 2014. He has been serving as a Vice President of Korean Convergence Society (KCS) since 2012, and was appointed as an Adjunct Professor at Chiang Mai University, Chiang Mai, Thailand, in 2016. Dr. Lee organized several international conferences with KCS and was awarded multiple honors such as outstanding scholar/best paper award from KCS and Korean Fuzzy Society.

Eng Gee Lim (M'98-SM'12) received the BEng(Hons) and PhD degrees in Electrical and Electronic Engineering from the University of Northumbria, UK. Prof. Lim worked for Andrew Ltd, a leading communications systems company in the United Kingdom from 2002 to 2007. Since August 2007, Prof. Lim has been at Xian Jiaotong-Liverpool University, where he was formally the head of EEE department and University Dean of Research and Graduate studies. Now, he is School Dean of Advanced Technology, director of AI university research centre and also professor in department of Electrical and Electronic Engineering. He has published over 100 refereed international journals and conference papers. His research interests are Artificial Intelligence, robotics, AI+ Health care, international Standard (ISO/IEC) in Robotics, antennas, RF/microwave engineering, EM measurements/simulations, energy harvesting, power/energy transfer, smart-grid communication; wireless communication networks for smart and green cities. He is a charter engineer and Fellow of IET. In addition, he is also a senior member of IEEE and Senior Fellow of HEA.

Alan Marshall (M'88–SM'00) has spent more than 25 years working in the telecommunications and defense industries. He has been a Visiting Professor in network security with the University of Nice/CNRS, Nice, France, and an Adjunct Professor for Research with Sunway University, Malaysia, Subang Jaya, Malaysia. He is currently the Chair in Communications Networks with the University of Liverpool, Liverpool, U.K., where he is also the Director of the Advanced Networks Group. He has formed a successful spin-out company Traffic Observation and Management (TOM) Ltd., specializing in intrusion detection and prevention for wireless networks. He has authored more than 200 scientific papers and holds a number of joint patents in the areas of communications and network security. His research interests include network architectures and protocols, mobile and wireless networks, network security, high-speed packet switching, quality of service and experience architectures, and distributed haptics. He is a Fellow of The Institution of Engineering and Technology. He is a Section Editor (Section B: Computer and Communications Networks and Systems) for the Computer Journal of the British Computer Society, and sits on the program committees of a number of IEEE conferences.