EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Mobile beacon-based adaptive time synchronization for wireless sensor networks

Jingsha He[1,2], Xinggang Xuan[1*], Nafei Zhu[1], Na Huang[1] and Peng He[2]

Abstract

Time synchronization of clocks in the sensor nodes for wireless sensor networks (WSNs) is a fundamental technology for most mission-critical applications. Most of past research in time synchronization for WSNs, however, has only focused on achieving some of the goals at a time, such as accuracy, energy consumption, completion time, etc., making these solutions less capable of adapting to different application requirements. In this paper, we propose a new time synchronization algorithm named MBATS (mobile beacon-based adaptive time synchronization) in which a mobile beacon is employed to move or fly over the sensor deployment area to complete time synchronization. Moreover, MBATS is designed so that the number of sensor nodes that are synchronized by one instance of time synchronization from the mobile beacon could vary dynamically to meet application requirements on accuracy, completion time and energy consumption, making the proposed MBATS algorithm highly adaptable to different application requirements. In addition to showing the advantage of the proposed MBATS algorithm on the adaptability of time synchronization as well as on some of the main metrics of synchronization over comparable schemes for WSNs, we also present the results of our study on comparing the performance of letting the mobile beacon traverse along a designing path versus follow a random path. Such a study is important since it would allow us to learn the performance gains that we can expect to achieve with extra control effort spent on designing the path over the effortless random path strategy. Such study could provide us with some clues on how to choose a suitable time synchronization strategy to better meet application requirements, which may not necessarily be the designed path strategy due to the tradeoff between cost and performance gains.

Keywords: Wireless sensor networks, Time synchronization, Adaptiveness, Energy efficiency

1 Introduction

Wireless sensor networks (WSNs) are self-organized and distributed networks composed of many deployed sensors that are normally employed to sense and collect data about some desired environmental or physical properties and transmit these data in a multi-hop fashion to the server for integration and processing. The collected data are usually transmitted to a sink node that connects to the Internet from where they are forwarded to the server. WSNs have been applied to military, medical, industrial, and many other areas of applications rapidly [1]. Such networks have become one of the most important parts of the future information infrastructure for Internet of Things (IoTs) [2] and will also replace the wired networks that are still in wide use today.

For WSNs, the time of the collected data may be very critical for data processing and integration, making time synchronization a fundamental requirement for many applications with related requirements. This is because in a WSN, every sensor node runs independently and executes on its own clock. Due to the differences of the internal structure of sensor nodes as well as the external operating conditions, clocks in sensor nodes may drift over time. Many applications relying on the data from WSNs have a very strict requirement on the time when the data are collected, especially when integration of data is needed in the applications. Time synchronization is prerequisite to these kinds of applications, and the

* Correspondence: xgxuan@emails.bjut.edu.cn
[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
Full list of author information is available at the end of the article

accuracy of time synchronization thus has a crucial effect on the effectiveness and performance of such applications.

Time synchronization for WSNs has emerged as one of the key technologies to ensure the consistency of data, for the problem is effectively dealt with at the root. It has been recognized that time synchronization technology can ensure that all the sensor nodes in a WSN have the same reference of time [3]. However, due to resource constraints in wireless sensor nodes, time synchronization algorithms should also take into consideration of minimizing energy consumption as a basic requirement.

To achieve time synchronization, wireless sensor nodes should use a common time reference. Each and every node would make sure that its clock value, which is used as the approximate time by the node, is aligned as closely as possible to the clock of the reference node.

Uncontrolled message loss and delay are the major factors in communication that can also cause clock skew and offset, which can be difficult to measure accurately by sensor nodes. Therefore, it is very hard for any node to perfectly compensate the clock offset.

Some conventional time synchronization schemes have spent some effort on reducing the effect of message delays. In single-hop synchronization, almost all factors except those related to signal propagation, interrupt handling and context switching have been addressed. However, error accumulation in multi-hop synchronizations is still an issue that has not been well studied.

Besides the accuracy of clocks, energy efficiency in time synchronization is a critical issue for WSNs [4]. Energy consumption is directly related to the number of messages that need to be transmitted during the whole process of synchronizing the different clocks. Some work proposed to take advantage of the broadcast nature of wireless media to reduce the number of message transmission during time synchronization, thus improving energy efficiency [5]. Following such an approach, all sensor nodes would simply broadcast their time information without concerning delay. Instead, nodes would try to minimize the effect of delay by adopting MAC-layer timestamping. The disadvantages of such broadcast-based approach include severe collisions and useless message transmissions that would increase the number of message, resulting in higher energy consumption.

In this paper, we propose a new time synchronization algorithm called MBATS (mobile beacon-based adaptive time synchronization) for WSNs. In MBATS, a mobile beacon is used to broadcast time reference values to the sensor nodes as it traverses through the area covered by the WSN in contrast to the traditional approach of sender-receiver synchronization using a stationary reference source. By dynamically adjusting the number of sensor nodes that are synchronized for each instance of synchronization initiated by the mobile beacon, MBATS can adapt to different application requirements, such as accuracy, energy consumption, and completion time. To achieve a high degree of accuracy, the mobile beacon would only synchronize the sensor nodes that are one hop from it, thereby minimizing errors as well as energy consumption of the sensor nodes since no synchronization needs to be performed among the sensor nodes themselves. If the requirement on accuracy is not very strict, sensor nodes can also participate in time synchronization, reducing the amount of time for synchronizing the entire network while incurring more energy consumption of the sensor nodes. Thus, MBATS can adapt to different application requirements on synchronization accuracy, completion time, and energy consumption, making it a flexible solution for WSNs of various sizes with different application requirements.

We will also conduct performance evaluation of methods for clock synchronization for WSNs in which mobile beacon nodes are employed. Our evaluation consists of a set of experiments to show that MBATS can indeed achieve the objectives of both accuracy in time synchronization and reduction in energy consumption when compared to existing time synchronization schemes in addition to its high degree of adaptability. Our experiment considers the scenarios in which the mobile beacon traverses randomly as well as along a designed path. The latter allows us to further investigate how much performance gain could be achieved after spending the extra effort on designing a path and directing the mobile beacon to travel along the path to complete time synchronization. The results of such a study would provide us with a better understanding of the trade-off between path design and performance improvement so that a strategy can be chosen to make MBATS better suit the application requirements on time synchronization.

The remainder of this paper is organized as follows. We review some related work and describe our system model in Sections 2 and 3, respectively. We then present the MBATS algorithm in Section 4. In Section 5, we describe and analyze the experiment results. Finally, in Section 6, we conclude this paper with some discussion on future work.

2 Related work

Since Elson introduced a time synchronization mechanism for WSNs [6], more clock synchronization algorithms have been proposed. In paper [7], Elson presented reference-broadcast synchronization (RBS) algorithm in which the reference node periodically broadcasts messages to its neighboring nodes that compare the local timestamp to that in the broadcast messages to calculate time bias to realize time synchronization. Ganeriwal presented TPSN (timing-sync protocol for sensor networks) which is a two-way synchronization method completed in two stages, namely, the level discovery phase and the synchronous phase [8]. TPSN is an effective time synchronization protocol and is easy to implement, but it has the shortcoming of cumulating errors and unbalanced network synchronization accuracy [9]. It has been noticed that local clock offset would increase as the number of layers of synchronization increases and decreasing synchronization interval would improve the accuracy but increase the traffic load in the network [10].

Ping proposed the DMTS (delay measurement time synchronization) algorithm which is a one-way synchronization algorithm [11], namely, the receiving node calculates the time adjustment by accurately measuring the one-way delay from the sending node to the receiving node as well as the timestamp value of the sending node. Maroti proposed the FTSP (flooding time synchronization protocol) algorithm based on analysis of the sending and receiving process and subdivides the time delay in order to reduce the uncertainty of delay [5]. Through sending multiple signaling messages, the receiving node can use minimum variance of the linear fitting method to estimate drift and shift between its clock and that of the sending node.

Some synchronization algorithms or protocols, such as CCS (consensus clock synchronization) [12], ATS (Average TimeSync) [13], and CCTS (cluster-based consensus time synchronization) [14] can achieve time synchronization of a whole network by using local information. Some routing algorithms improve the performance of routing by taking consideration of the transmission delay [15]. In the distributed protocol R4Syn that was proposed for doing both local and multi-hop synchronization [16], the receiver-to-receiver paradigm is used, which has the advantage of reducing the time-critical-path, thus improving the accuracy compared to common sender-to-receiver protocols. The RTSP (Recursive Time Synchronization Protocol) proposed to synchronize all the nodes in a network to a global clock using multi-hop architecture in an energy-efficient way [17]. Due to the special structure of the nodes themselves, wireless sensors have limited energy, which prohibits wireless sensor nodes from running complex protocols and computations for time synchronization [18].

Although there are already a number of time synchronization methods or protocols for WSNs, the clock source of the reference node is usually location fixed. Due to limited communication range of wireless sensor nodes, even a reference node that has wide communication coverage can hardly cover the entire network within a one-hop connection. Therefore, to achieve time synchronization for the network, cluster head nodes or multiple hop links formed through other intermediate nodes have to be relied upon to transmit synchronization information to achieve time synchronization, which results in a tremendous amount of traffic and high energy consumption. At the same time, as the number of forwarding nodes increases, clock skews accumulate, resulting in lower accuracy of synchronization. It is generally expected that the performance of time synchronization would decrease as the size of a network increases.

Zhang et al. proposed a time synchronization algorithm based on the use of a mobile node in which a path is designed for the mobile node to traverse the whole network to complete time synchronization with a one-hop fashion [19]. Although the algorithm can achieve desired accuracy and maintain a minimum level of energy consumption, the amount of time that it takes to complete network-wide synchronization may be too long, especially for large networks. Boukerche et al. proposed an algorithm called LTS-MB in which a mobile node is used to perform time synchronization and node localization together [20]. Villas et al. proposed to use an unmanned aerial vehicle to do localization and synchronization [21]. However, in most applications, localization of sensor nodes only needs to run once at the time of network formation or repeats much less frequently than time synchronization which must usually be carried on periodically. It is thus not suitable to tie both localization and synchronization together for the purpose of saving energy. In addition, sensor localization usually has a much stricter requirement on the design of the mobile path than time synchronization for the mobile beacon to traverse the network, thus causing much higher overhead for time synchronization unnecessarily.

3 The system model

For the purpose of time synchronization, nodes in a WSN are classified into three types, as illustrated in Fig. 1, each taking distinctive roles and carrying out different functions.

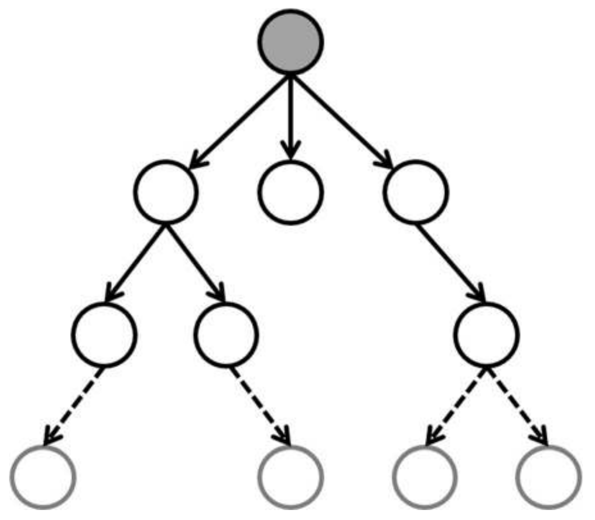

Fig. 1 Nodes in a WSN for time synchronization

The first type is the reference nodes. A reference node provides global reference time which is almost always the sink node. A reference node usually relies on an external source such as the GPS to obtain time values. In mobile beacon-based time synchronization, a mobile beacon acts as the reference node to perform time synchronization of regular sensor nodes which is expected to be equipped with GPS to make it capable of obtaining the current time. The mobile beacon is also expected to be capable of moving freely throughout the network. We assume that the mobile beacon has abundant energy supply since it can come back to the base after completing one round of time synchronization to get recharged before going out for the next round of time synchronization.

The second type is the intermediate nodes or non-leaf nodes. When an ordinary sensor node finishes time synchronization, it could get involved in helping other sensor nodes get synchronized, making itself an intermediate node. Such nodes will behave according to parameter setting in the execution of the synchronization algorithm.

The third type is the leaf nodes. When an ordinary sensor node finishes time synchronization, it would not involve in helping any other sensor nodes get synchronized, making itself a leaf node.

A schematic illustration of time synchronization using a mobile beacon is illustrated in Fig. 2.

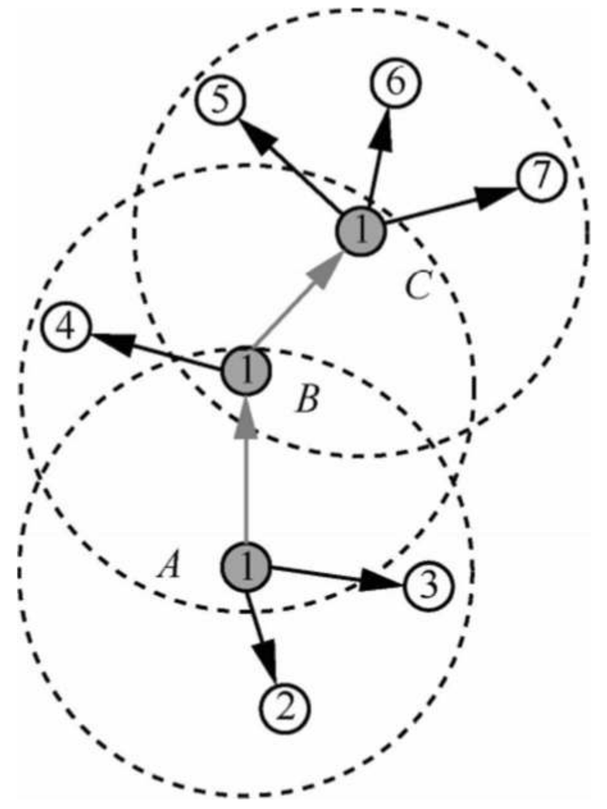

Fig. 2 Time synchronization using a mobile beacon

Suppose there are N sensor nodes that are randomly deployed in an area. The mobile beacon (an unmanned vehicle or a flying drone) traverses across the area at a certain speed and sends messages to the ordinary sensor nodes along the way to synchronize the time in their local clocks. The goal is to make the clocks arrive at a common time for the sensor nodes in a WSN through synchronizing the clocks in each and every of the sensor nodes.

In our algorithm, we assume a virtual hierarchical topology for the sensor nodes and each node is assumed to sit at a certain layer in this structure when the mobile beacon moves along a path to perform synchronization. We define nodes at level *i* if and only if they can communicate with nodes at level *i*-1, and the mobile beacon is the only node at level 0 anytime and anywhere during its traversal through the network. Different from TPSN, in our algorithm, time synchronization is performed without explicit discovering layers of the nodes. Rather, the layers of the nodes are formed dynamically. If a node at layer *i*-1 can synchronize node S, then *S* is said to be located at layer *i*. When synchronization ends, all nodes are synchronized to the root node, i.e., the mobile beacon.

Usually, the root node can be a sink node that acts as the gateway between the external world and the WSN. In our MBATS algorithm, we use the mobile beacon as the root node that is equipped with a GPS receiver so that it is capable of synchronizing all the sensor nodes to that of the physical world. For a very large WSN, if the length of time that it takes for one mobile beacon to cover the entire network is too long, the network can be divided into multiple smaller networks and a separate mobile beacon can be deployed for each of the small networks to synchronize nodes independently and in parallel. For ease of discussion, we will only consider the scenario in which the entire network has just one mobile beacon as the root node for time synchronization.

We assume that the mobile beacon has abundant energy for time synchronization, for it can be recharged between successive synchronization cycles without affecting the life of the sensor network. The mobile beacon could also consume much more energy on its mobile and GPS functionalities than on time synchronization. We also assume that each node has a unique identifier so that nodes can recognize each other using their identifiers.

In our algorithm, we define *n* as the maximum number of layers of synchronization according to the accuracy requirement. Such a number can also be determined according the requirement on completion time. We assume that each layer of synchronization will accumulate some error due to clock skew at the receiving node in relative to the sending node which is described using parameter δ. We would like to point out, however, that due to physical conditions or surrounding environments, the value of δ may vary at different points of time.

Therefore, some measurements should be performed before sensor deployment to get a proximate value or a set of values for δ in order to derive a reasonable and proximate value for the synchronization error. If the accumulative error in the requirement is defined to be no more than M, then

$$n \le \frac{M}{\delta} \tag{1}$$

We can thus set the maximum number of layers for time synchronization according to the requirement on accuracy. We can further define the average error S of the network as

$$S = \frac{1}{N}\sum\nolimits_{i=1}^{n} i^{*}L_i \tag{2}$$

where N is the total number of nodes in the network and L_i is the number of nodes that are synchronized at the ith layer. It can be seen from (2) that the more the number of nodes that can be synchronized with fewer number of layers, the smaller the average error will be.

For one instance of time synchronization between two nodes, the amount of energy consumed for sending a message and receiving a message is denoted by E_s and E_r, respectively. For a WSN of N nodes, if each node consumes E_r amount of energy, then the N nodes would consume NE_r amount of energy. In addition, if the number of intermediate nodes is m, each of which would only broadcast one message, the total amount of energy consumed is mE_r. Therefore, without taking into consideration of the loss and collision of messages, the total amount of energy consumption would be

$$Q = NE_r + mE_s \tag{3}$$

In reality, however, energy consumption should generally be higher than Q because there may be repeated synchronization effort due to various problems, such as transmission failure caused by collisions.

4 The proposed MBATS algorithm

For synchronization between two nodes A and B, node A would send a *time _ synchronize* message to node B. The message would at least contain the layer number i of A and information that is needed for completing synchronization. Node B at layer $i+1$ is considered to be synchronized after it receives the message. Here, we omit the details of message transmission as well as the specific methods that can be employed to compute the clock value using the information in a synchronization message. This simplification or abstraction would allow us to focus on the characteristics of network-wide synchronization without having to get into the details of the derivation of the clock value since there are already many algorithms that are suitable for such tasks. We also omit potential message transmission errors since techniques already exist to deal with such issues.

As the mobile beacon moves or flies around, the execution of MBATS continues. As discussed earlier, MBATS does not require that a hierarchical structure be established in advance. After being synchronized, the nodes at a layer would broadcast synchronous messages and any nodes that receive the messages will also get synchronized. The process continues until a receiving node determines that it has already been synchronized during the current cycle or the layer has reached the maximum number defined. When the mobile beacon initiates synchronization by broadcasting the synchronization message, there is no need to establish a cluster or a tree structured topology for the network. Therefore, the failure of a single node will not cause time synchronization of the network to stop or to fail unless the failed node sits at a critical point where subnetworks join at this point. Then, the unsynchronized subnetworks would have to wait for the mobile beacon to come to the respective areas to initiate time synchronization. The clock synchronization robustness is consequently increased greatly.

The steps of the MBATS algorithm are described as follows:

(1) The mobile beacon always gets its time through the GPS. So it is considered to be well synchronized with the physical world time at all times. The mobile beacon traverses the network following a predesigned path and will broadcast the synchronization message along the way. Also, mobile beacon is always assigned at layer 0.
(2) After a node is synchronized (initially the mobile beacon only), it would wait for a random amount of time and then broadcast a synchronization message which contains the layer number and its identity to its immediate neighbors. In a cycle of synchronization, nodes that have already completed synchronization will simply drop incoming synchronization messages to reduce energy consumption.
(3) Upon receiving a synchronization message, a node would assign itself a layer which is one layer higher than that found inside the synchronization message that it has just received. The node will perform synchronization of its internal clock if it has not done

so during the current synchronization cycle. This node can thus be considered to be synchronized.

(4) If the current layer at the node reaches the predefined layer number n or if the node does not actually perform synchronization, synchronization stops, i.e., no more synchronization messages will be broadcast to any other nodes. Otherwise, go back to step (2) to continue.

(5) The mobile beacon continues sending the synchronization message as it traverses the network until it returns to the base following a defined parameter. The mobile beacon would repeat time synchronization for every cycle, a parameter defined based on the size of the network and the requirement on the accuracy of clock times.

In step (4), n represents the maximum number of synchronization layers determined in advance. When $n = 1$, synchronization stops at the nodes that receive the synchronization message from the mobile beacon, resulting in the minimal average error. When $n = 2$, the network will simulate a cluster structure during synchronization and when $n>2$, the network will simulate a tree structure during synchronization. By setting the maximal number of synchronization layers, we can flexibly control network topology structure according to actual performance requirement such as synchronization accuracy, completion time, and energy consumption.

The mobile beacon traverses the network following a path that can be either random or predesigned. Different paths are expected to cause different impact on the MBATS algorithm. It is obvious that too many synchronization messages from the mobile beacon to the same nodes in a synchronous cycle may result in the waste of energy. At the same time, to guarantee the coverage of the entire network, optimal path design will help to shorten the total amount of time spent on time synchronization.

5 Experiment

It may be desirable for the mobile beacon to follow a path to ensure that all sensor nodes can be synchronized and synchronization can meet certain performance requirements in terms of accuracy, completion time, and energy consumption. Although path design can be the subject of a separate research, to evaluate the effectiveness of MBATS, we designed two moving paths in our experiment. The first follows a random moving model as shown in Fig. 3a for its simplicity and minimal overhead in choosing a path for the mobile beacon to traverse. Thus, in a synchronization cycle, it is possible that the mobile beacon pass through some regular nodes more than once while leaving some others unattended, resulting in unnecessary waste of energy and prolonging the time to cover the entire network. Nonetheless, since this could be the simplest design of a path (actually no design), we can use the corresponding results as the benchmark in the performance evaluation of other path designs to which existing schemes of time synchronization are applied. Another path design in our evaluation is shown in Fig. 3b in which the goal is to ensure that the mobile beacon cover all sensor nodes during one round of synchronization. We call this scheme the designed path throughout our discussion.

We assume in the experiment that R is the radius of the signals from the mobile beacon and from each and every of the regular sensor nodes. Thus, each circle in Fig. 3 represents an area that can be synchronized by one instance of synchronization initiated by the mobile beacon as it traverses through the WSN. In the case of the designed path in Fig. 3b, if the number of layers of synchronization is set at n, to fully cover the entire sensor deployment area, the distance W between adjacent lines in the designed path cannot be larger than $1.5nR$. In our experiment, we simply set $W = 1.5nR$.

We used the NS2 simulation platform to evaluate the performance of our proposed MBATS algorithm. The

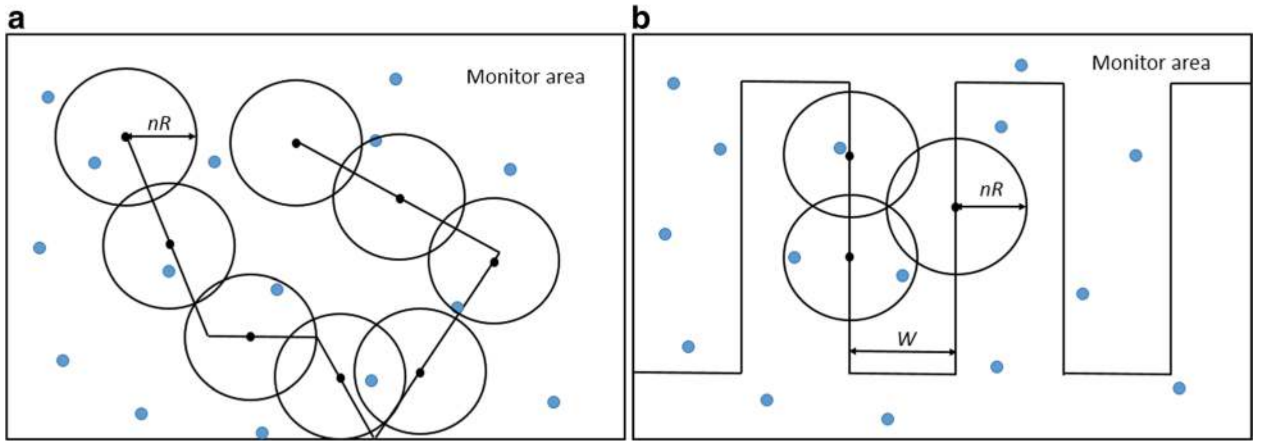

Fig. 3 Path selection for the mobile beacon in MBATS. **a** The random path. **b** The designed path

Table 1 Parameters for the experiment

Parameters	Value
Number of sensor nodes	500 nodes
Communication range (R)	15 m
Beacon node communication range (R)	15 m
Sensor deployment area	100×100 m^2/200×200 m^2
Number of mobile beacons	1
Beacon velocity	10 m/s
Synchronization error	δ(1 hop)
RSSI inaccuracy	20% of communication range
Energy consumption for message transmission	0.08 J
Energy consumption for message reception	0.02 J
Length of synchronization message	568 bits

purpose is to assess the MBATS in terms of synchronization error, energy consumption and network coverage. The parameters for the experiment are listed in Table 1. As can be seen, 500 nodes are randomly deployed in an area of 100×100 m^2 to simulate a network with dense deployment and in an area of 200×200 m^2 to simulate a network with sparse deployment, both of which are deployment scenarios that we are interested in evaluating the MBATS algorithm.

We repeated each instance of the experiment 30 times and then calculate an average value based on the 30 results. All the curves in the figures thus represent the average values of the results from the experiment.

In our comparison to other time synchronization schemes, we ran the TPSN algorithm with the reference node being placed at the center of the area. The same practice was followed for the R4Syn algorithm and the MBATS algorithm when $n = 1$. For the MRN-CS and the LTS-MB algorithms, the parameters were set according to the original configurations. For the purpose of the comparison, the mobile path that was employed in all the relevant algorithms was the same as the designed path in the MBATS algorithm. For our MBATS algorithm, the mobile beacon would move or fly through the network following the two path designs shown in Fig. 3, respectively. In all the experiments, we collected node synchronization errors relative to the reference node or the mobile beacon.

6 Results and discussion

6.1 Nodes distribution

Our first experiment intends to evaluate the percentage of nodes that get synchronized at different layers. The results are shown in Fig. 4 for densely and sparsely deployed networks, respectively, from which we can see that the percentage of nodes that are synchronized at the same layers is not much different for both path designs. This means that path design does not have much effect on synchronization accuracy and energy consumption.

6.2 Synchronization errors

For the LTS-MB and the MRN-CS algorithms, the synchronization error of every sensor node remains constant at δ since only one layer of synchronization is performed. For the MBATS algorithm, n is set between 1 and 5 in the experiment, producing five curves as the result. The experiment results are shown in Figs. 5 and 6. We can see that, when $n = 1$, the results of the MBATS algorithm are the same as those of the LTS-MB and the MRN-CS algorithms. When n>1, error would start to cumulate and the more the number of layers, the higher the amount of cumulated errors. The results of running the MBATS algorithm with the random path are shown in Fig. 6

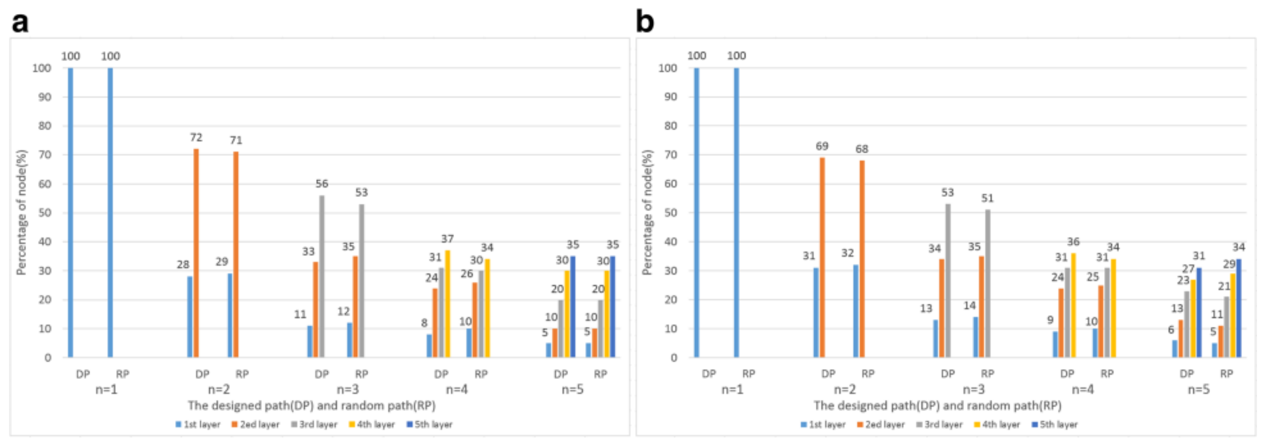

Fig. 4 Percentage of nodes synchronized at different layers. **a** The densely deployed network case. **b** The sparsely deployed network case

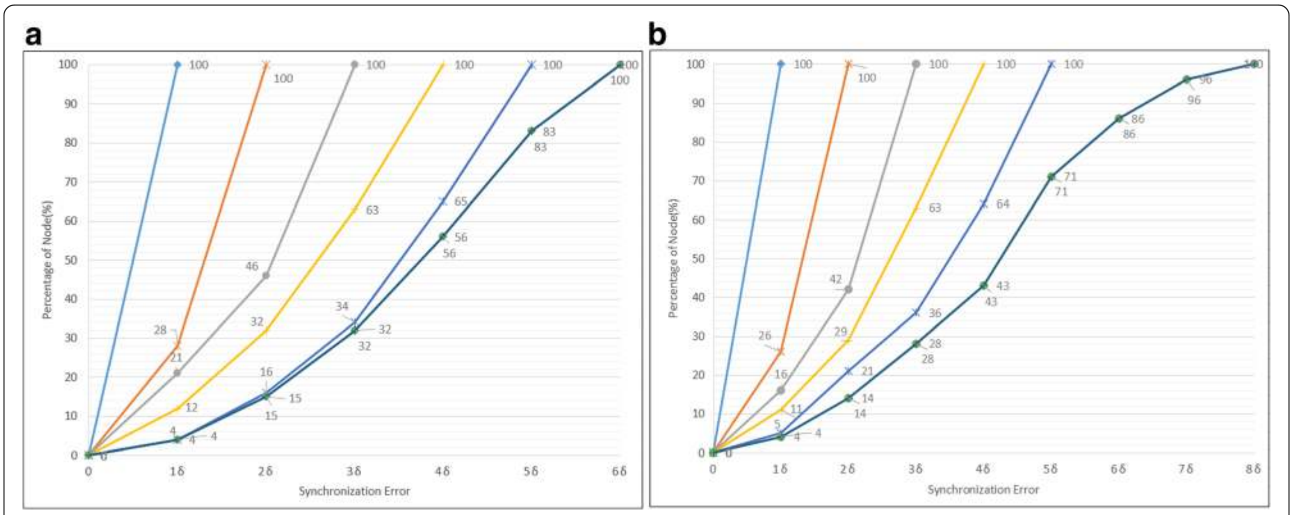

Fig. 5 Comparison of synchronization error when MBATS employs the random path. **a** The densely deployed network case. **b** The sparsely deployed network case

which are not too much different from those in Fig. 5. This shows that synchronization errors resulting from accumulation would not be impacted by the path along which the mobile beacon moves or flies. Therefore, for any applications, n should be determined using Formula (1) based mostly on the requirement on synchronization accuracy.

6.3 Energy consumption

In this experiment, we compared the proposed MBATS algorithm to the TPSN and the R4Syn algorithms since the LTS-MB and the MRN-CS algorithms are both one-hop algorithms which would result in the same energy consumption as MBATS in the case of $n = 1$ when the mobile beacon moves along the designed path. The results of the experiment are illustrated in Figs. 7 from which we can see that the MBATS algorithm consumes less energy than the TPSN and the R4Syn algorithms in all the cases. It is, however, only in the MBATS algorithm that the number of layers would affect the way how the algorithm works and thus the energy consumption. Moreover, for the MBATS algorithm, the designed path case consumes less energy than the random path case since in the latter, the mobile beacon could pass through some regular sensor nodes more than once which causes extra energy consumption for such nodes.

It should be noted that energy consumption in the TPSN algorithm is not the same for all the nodes,

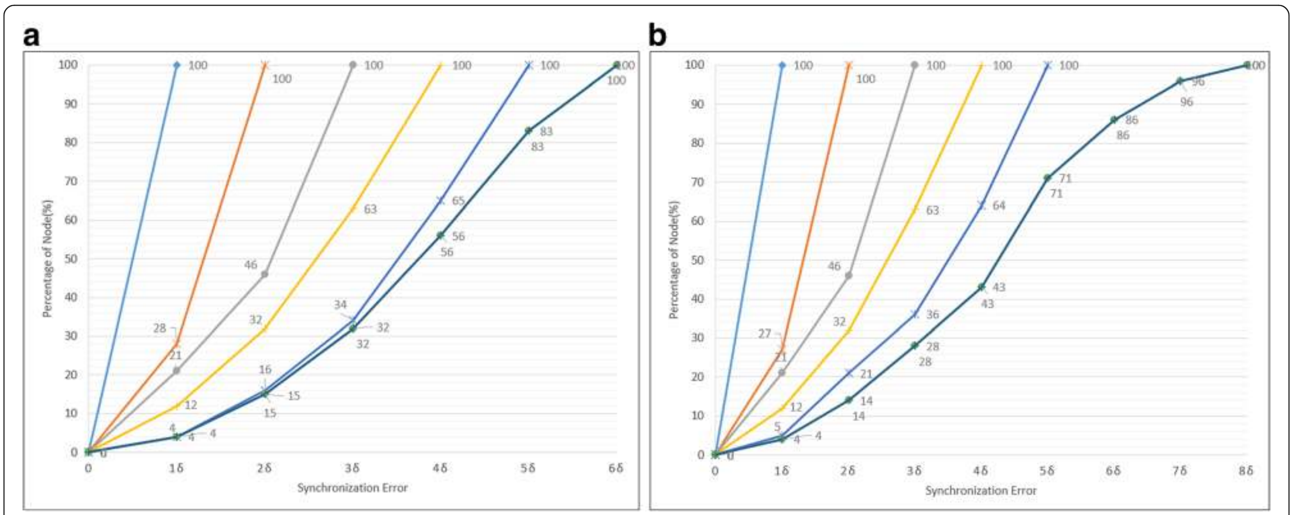

Fig. 6 Comparison of synchronization error when MBATS employs the designed path. **a** The densely deployed network case. **b** The sparsely deployed network case

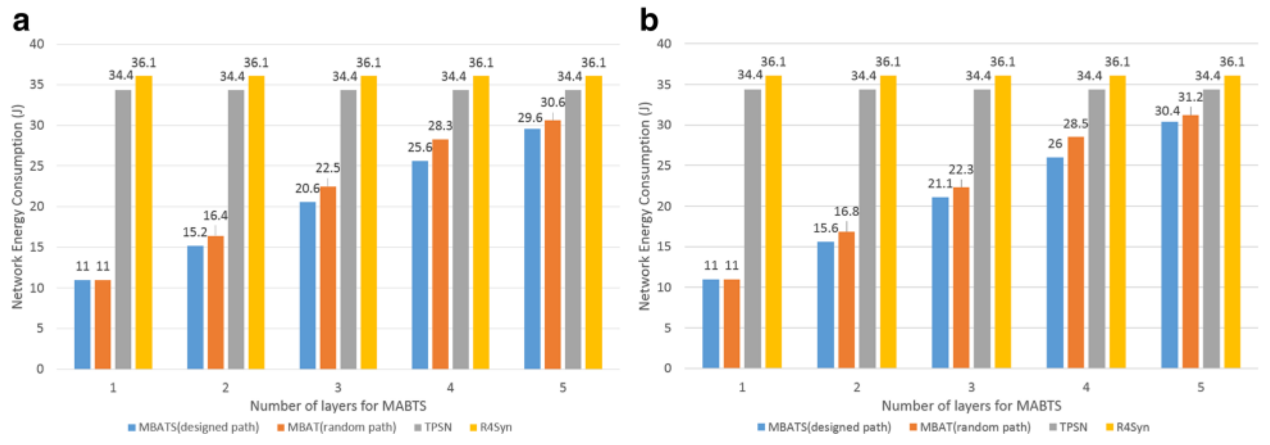

Fig. 7 Comparison of energy consumption. **a** The densely deployed network case. **b** The sparsely deployed network case

resulting in some of the nodes to exhaust energy more prematurely than some others. This is because the TPSN algorithm relies purely on message transmission to complete time synchronization for all the nodes. In a tree or cluster-like structure, leaf nodes only need to be synchronized by other nodes, thus leading to less energy consumption. However, non-leaf nodes need to synchronize other nodes and thus communicate with multiple nodes, thus increasing energy consumption that causes them to run out of energy more quickly. In the MBATS algorithm, the role of the sensor nodes in the network is not fixed in that a node could be a leaf node or a non-leaf node. This makes energy consumption spread more evenly across the nodes, thus helping to prolong the life of the network.

6.4 Efficiency of synchronization

Efficiency of synchronization is concerned about how much time it takes to complete synchronizing the entire network of sensor nodes. Figs. 8 and 9 show how the network can be covered by synchronization algorithms and at what speed (i.e., the rate of coverage). When a regular sensor node fails to be synchronized during the current cycle, we say that this node is not covered. As shown in the figures, due to the random moving pattern of the mobile beacon, some nodes may not be covered or failed to be synchronized for quite some time. In another word, it will take longer time to fully cover the entire network. Obviously, the rate of coverage relates to node density as well as to the size of the network. The greater the density and the smaller the size of the network, the better the coverage and the rate. As can be seen, the rate of coverage in the MBATS algorithm is affected by the maximal number of layers for time synchronization. The smaller the number of layers, the lower the coverage as well as the rate. When $n = 5$, synchronization can cover all the nodes fairly quickly and the rate of coverage will reach 100% faster than when the parameter assumes other lower values. Therefore, for applications with high-efficiency requirement, the design path option should be chosen.

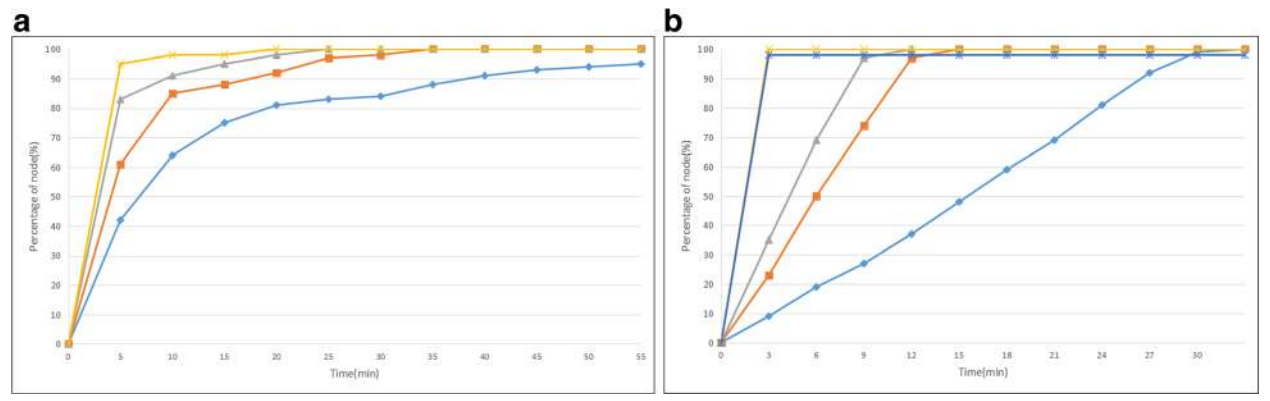

Fig. 8 Efficiency of synchronization for a densely deployed network. **a** The random path case. **b** The designed path case

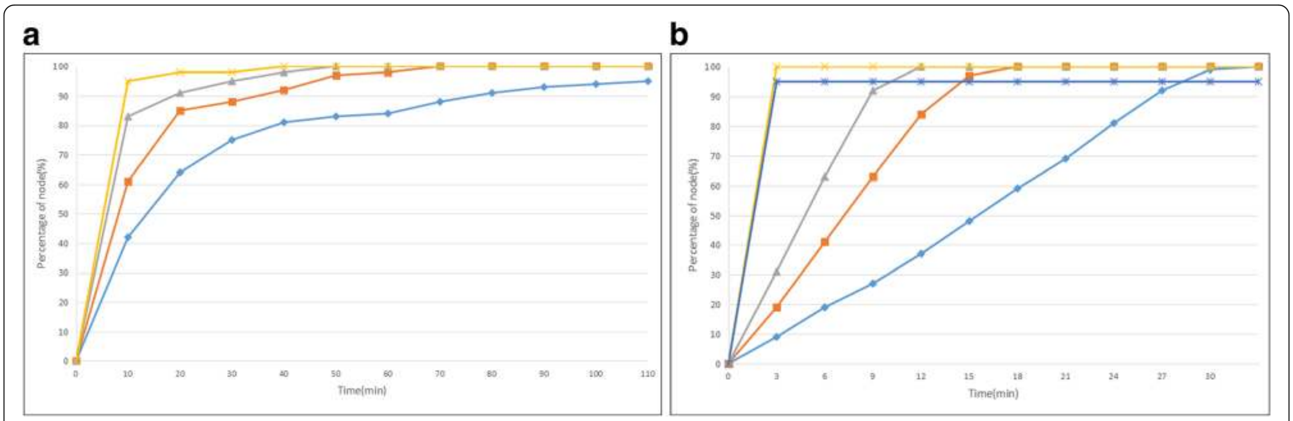

Fig. 9 Efficiency of synchronization for a sparsely deployed network. **a** The random path case. **b** The designed path case

Note that results for the TPSN algorithm is not explicitly shown in the figure since all nodes can be synchronized almost instantaneously in the algorithm, which is equivalent to the case in MBATS when the number of layers is unlimited or infinite. Also, neither the LTS-MB algorithm nor the MRN-CS algorithm considers a random moving path for the mobile beacon.

7 Conclusions

Time synchronization is an indispensable requirement for WSNs in many applications. In this paper, we proposed and evaluated a new, adaptive algorithm for time synchronization, which we referred to as the MBATS (mobile beacon-based adaptive time synchronization) algorithm. By using a mobile beacon in the algorithm, all sensor nodes can be synchronized to meet several performance requirements such as synchronization accuracy, time of completion, and energy consumption. In addition, there is no need to reconstruct the network topology when new sensor nodes join the network or existing sensor nodes leave the network since no topology of the network is assumed to run the MBATS algorithm. We claim that MBATS can offer a number of advantages over some existing algorithms such as MRN-CS, LTS-MB, TPSN, and R4Syn in lowering synchronization errors and energy consumption through experiment and analysis. We can also set the maximum number of layers in MBATS to suit networks of different sizes as well as different synchronization requirements. In the future, we will focus on the relationships between network size and the maximum number of layers to further improve the performance of time synchronization. We will also devise more optimal paths that can further improve the various performance metrics to meet different application requirements.

Acknowledgements
The authors would like to thank Yuqiang Zhang for his comments and suggestions during technical discussions.

Funding
The work in this paper has been supported by National Natural Science Foundation of China (61602456) and National High-tech R&D Program (863 Program) of China (2015AA017204). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Authors' contributions
JH and XX conceived the idea and designed the algorithm. XX, NZ, and NH performed experiment and data analysis. PH participated in technical discussion and helped in data analysis. JH and XX prepared the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. [2]College of Computer and Information Science, China Three Gorges University, Yichang 443002, Hubei, China.

Received: 20 August 2017 Accepted: 27 August 2018
Published online: 14 September 2018

References
1. S. Misra, A. Vaish, Reputation-based role assignment for role-based access control in wireless sensor networks. Comput. Commun. **34**(3), 281–294 (2011)
2. L. Atzori, A. Iera, G. Morabito, The internet of things: A survey. Comput. Netw. **54**(15), 2787–2805 (2010)
3. A.R. Swain, R.C. Hansdah, A model for the classification and survey of clock synchronization protocols in WSNs. Ad Hoc Netw. **27**, 219–241 (2015)
4. C. Wu, Y. Wang, Z. Yin, Energy-efficiency opportunistic spectrum allocation in cognitive wireless sensor network. Eurasip J Wireless Commun. Networking **2018**(1), 13 (2018)
5. M. Maróti, B. Kusy, G. Simon, Á. Lédeczi, in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. The flooding time synchronization protocol (2004), pp. 39–49
6. J. Elson, K. Römer, Wireless sensor networks: A new regime for time synchronization. ACM SIGCOMM Comput Commun Rev **33**(1), 149–154 (2003)

7. J. Elson, L. Girod, D. Estrin, Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Oper Syst Rev **36**(SI), 147–163 (2002)
8. S. Ganeriwal, R. Kumar, M.B. Srivastava, in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. Timing-sync protocol for sensor networks (2003), pp. 138–149
9. Z. Dengchang, A. Zhulin, X. Yongjun, Time synchronization in wireless sensor networks using max and average consensus protocol. Int J Distrib Sens Netw **2013**(8), 388–391 (2013)
10. S.M. Lasassmeh, J.M. Conrad, in *Proceedings of the IEEE SoutheastCon*. Time synchronization in wireless sensor networks: A survey, vol 2010 (2010), pp. 242–245
11. Ping, S. Delay Measurement Time Synchronization for Wireless Sensor Networks. (2003) Intel Research Berkeley Lab
12. M.K. Maggs, S.G. O'Keefe, D.V. Thiel, Consensus clock synchronization for wireless sensor networks. Sens. J **12**(6), 2269–2277 (2012)
13. L. Schenato, G. Gamba, in *Proceedings of the 46th IEEE Conference on Decision and Control*. A distributed consensus protocol for clock synchronization in wireless sensor network (2007), pp. 2289–2294
14. J. Wu, L. Zhang, Y. Bai, Y. Sun, Cluster-based consensus time synchronization for wireless sensor networks. Sens J **15**(3), 1404–1413 (2015)
15. K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks. Ad Hoc Netw. **3**(3), 325–349 (2005)
16. D. Djenouri, N. Merabtine, F.Z. Mekahlia, M. Doudou, Fast distributed multi-hop relative time synchronization protocol and estimators for wireless sensor networks. Ad Hoc Netw. **11**(8), 2329–2344 (2013)
17. M. Akhlaq, T.R. Sheltami, RTSP: An accurate and energy-efficient protocol for clock synchronization in WSNs. IEEE Trans. Instrum. Meas. **62**(3), 578–589 (2013)
18. F. Sivrikaya, B. Yener, Time synchronization in sensor networks: A survey. IEEE Netw. **18**(4), 45–50 (2004)
19. Y. Zhang, J. He, J. Xu, B. Zhao, F. Cai, Time synchronization method for wireless sensor networks based on mobile reference nodes. J. Commun. **36**(Z1), 171–180 (2015)
20. A. Boukerche, H.A. Oliveira, E.F. Nakamura, A.A. Loureiro, in *Proceedings of 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*. Localization in time and space for wireless sensor networks: A Mobile Beacon approach (2008), pp. 1–8
21. L.A. Villas, D.L. Guidoni, G. Maia, R.W. Pazzi, J. Ueyama, A.A. Loureiro, An energy efficient joint localization and synchronization solution for wireless sensor networks using unmanned aerial vehicle. Wirel. Netw **21**(2), 485–498 (2015)