Computer Communications 37 (2014) 77–91

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Long term and large scale time synchronization in wireless sensor networks

Ge Huang [a], Albert Y. Zomaya [a], Flávia C. Delicato [b,*], Paulo F. Pires [b]

[a] Centre for Distributed and High Performance Computing, School of Information Technologies, The University of Sydney, NSW 2006, Australia
[b] UFRJ, Federal University of Rio de Janeiro, Brazil

ARTICLE INFO

Article history:
Received 16 September 2011
Received in revised form 24 June 2013
Accepted 7 October 2013
Available online 17 October 2013

Keywords:
Quadratic Taylor synchronization
Time synchronization protocols
Wireless sensor networks

ABSTRACT

Time synchronization is very important in wireless sensor networks (WSNs). Many applications, for example natural disaster monitoring and structural health monitoring of huge buildings, require a highly accurate, long-term and large-scale time synchronization among the sensor nodes that compose the network. In this paper, we propose a new time synchronization protocol, named *2LTSP* (long term and large scale time synchronization protocol), which aims at addressing such requirements. Theoretical analysis and simulation results show that when the synchronization period is less than 100 s, the error of *2LTSP* is within 0.6 ms, no matter how large the size of the network is. Besides, when the required synchronization error limit is 9 ms, the communication cost of *2LTSP* is less than 3 packets per hour in networks of any size. Therefore, *2LTSP* is highly accurate and energy-efficient even for large-scale and long-term running networks.

1. Introduction

For most WSN applications, a scalable time synchronization service responsible for maintaining a common notion of time among the network nodes is often required to enable sleep scheduling, localization, data fusion, in-network signal processing and nodes' coordination, among others WSN essential functions.

Each node in the WSN is equipped with a hardware clock used as reference for controlling its internal signal processing functions. On top of the hardware clock, each node has a logical clock whose value and frequency can be changed by time synchronization algorithms. The rates of the hardware clocks of sensor nodes depend on conditions like the temperature, the supply voltage, or crystal quartz aging, all changing over time. Even under perfectly stable and identical environmental conditions, the hardware clocks of different nodes will run at different speeds. Moreover, each node has different initial hardware clock value. On one hand, the initial hardware clock value is set to be different during manufacturing. On the other hand, in order to save energy, the node goes through a cycle of sleep and wake up based on externally detected events. Thus, hardware clock error is inevitable, and time synchronization is essential.

Time synchronization algorithms are essentially functions to synchronize local clocks (clocks at individual sensor nodes) with the WSN global clock. These algorithms are based on exchanging messages time-stamped with the values of the hardware clock among nodes. The goal of a time synchronization algorithm is to ensure, by adjusting local clocks to the global clock at regular intervals, that logical clocks of any two nodes in the WSN are as closely synchronized as possible at all times. Each time synchronization algorithm will change, in its own way, the logical clocks of the synchronized nodes in its synchronization process. Different approaches for synchronizing nodes are prone to different drawbacks and errors. Moreover, most of current time synchronization algorithms for WSNs meet requirements of specific application domains. Our proposed algorithm aims at covering a broad range of application requirements while trying to overcome several limitations of existing approaches. In the next paragraphs, we briefly discuss the current main issues and challenges for time synchronization algorithms in WSNs. First, we discuss the issue of spatial cumulative error in WSN time synchronization algorithms. Second, we analyze the accuracy provided by such algorithms. Finally, we discuss the different approaches in the light of energy consumption.

If a synchronization algorithm can only achieve its goal by performing multiple steps (i.e. multiple exchanges of messages among nodes), we call it an *accumulative one*, otherwise a *non-accumulative one*. Since WSN usually span many hops, the synchronization error introduced in *accumulative* algorithms by instability of the clock frequency and variable delays of message exchange between pair wise synchronizing nodes accumulates as the path length grows [30]. In other words, as we prove in Section 2.3 and demonstrate in our simulation experiments, the error of any accumulative

* Corresponding author.
E-mail address: fdelicato@gmail.com (F.C. Delicato).

0140-3664/$ - see front matter Crown Copyright © 2013 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.10.003

synchronization algorithm has spatial accumulative effect. Designing of non-accumulative synchronization algorithms is a challenging work. The authors in [1] have shown that even popular clock synchronization algorithms have several problems in this context. For instance, the state-of-the-art clock synchronization algorithm FTSP [2] exhibits an error that grows exponentially with the size of the network. To the best of our knowledge, the PulseSync synchronization algorithm [1] is the only non-accumulative algorithm in the literature. Since the error of PulseSync has no spatial accumulative effect, it is suitable for large-scale networks. However, the PulseSync is not suitable for long-term running networks due to its low accuracy (refer to Section 3).

For each node in a WSN, there exists, in theory, a unique function that maps, at any physical time, the local (hardware) clock to the global clock (refer to Section 4.3), which we call the *clock transfer function*. Time synchronization algorithms can be classified into two categories according to the type of *clock transfer function*: *linear* [1–7] and *quadratic* [8], which respectively use linear polynomials and quadratic polynomials to compute the clock transfer function. Moreover, a time synchronization algorithm is called a *quadratic Taylor one* if it uses the quadratic Taylor expansions to compute the clock transfer function. Quadratic Taylor synchronization algorithms are much better, in term of synchronization accuracy, than linear synchronization algorithms (refer to Section 2.2). Most existing synchronization algorithms have a low accuracy because they approximate the global clock using linear polynomials rather than quadratic Taylor expansions. To satisfy the application's demand for synchronization accuracy, synchronization algorithms with low accuracy must be executed frequently. For example, FTSP [2] has to resynchronize the nodes in the network every minute in order to achieve 90 ms synchronization error (an acceptable synchronization error limit for several applications). Consequently, synchronization algorithms with a low accuracy are only useful for short-term applications such as surveillance and object tracking and are unsuitable for efficient duty cycling and other applications that require continuous time synchronization such as synchronized sampling, because they spend a lot of energy due to their frequent resynchronization [8]. Despite the higher accuracy achieved with quadratic synchronization algorithms, their design is a challenging work. To the best of our knowledge, the synchronization algorithm proposed in [8] is the only quadratic algorithm currently reported in the literature. Its major shortcoming is to be unsuitable for large-scale networks (refer to Section 3).

When a synchronization process ends, the logical clock of each synchronized node is close to the global clock. However, over time, the logical clocks of the synchronized nodes gradually differ from the global clock. To obtain acceptable synchronization accuracy, the synchronization process has to be started at regular intervals. If the synchronization process is started once every T seconds, we say the *synchronization period* is T seconds. It is clear that a shorter synchronization period leads to higher synchronization accuracy. However, shorter synchronization periods also lead to larger energy consumption in the network nodes. Since energy saving is a crucial requirement in WSNs, the balance between synchronization accuracy and energy consumption needs to be maintained. For a given time interval T (unit: second) and time synchronization algorithm f, let's assume that $E_T(f)$ (unit: microsecond) represents the f's synchronization error for a synchronization period equal to T. For a given synchronization error limit ε (unit: microsecond) and time synchronization algorithm f, we use $T_\epsilon(f)$ (unit: second) to denote the maximal time interval T such that $E_T(f) \leqslant \varepsilon$, that is, $T_\varepsilon(f) = \max_T\{T|E_T(f) \leqslant \varepsilon\}$. We call $T_\varepsilon(f)$ the f's *immanent period* in the condition that the synchronization error limit is equal to ε (for short, ε-*period*). The longer the immanent period of a time synchronization algorithm is, the more energy-efficient it should be. The immanent period is used to compute the communication cost of time synchronization algorithms (refer to Section 2.4). The most representative time synchronization algorithms existing in the literature have all a short immanent period. For example, the immanent period considering $\varepsilon = 9\mu s$-periods of FTSP [2], RSP [4] and PulseSync [1] are, respectively, 100 s, 300 s and 100 s. Therefore, they are not suitable for long-term running networks.

Considering all the aforementioned issues and challenges, in this paper we present a new non-accumulative, quadratic Taylor time synchronization algorithm called *2LTSP*. Since *2LTSP* is non-accumulative, unlike most existing time synchronization algorithms, its error has no spatial accumulative effect as the size of the network increases. Therefore, it is suitable for large-scale networks. Moreover, as *2LTSP* approximates the global clock using a quadratic Taylor expansion rather than a linear polynomial as most existing schemes do, it is highly accurate. Theoretical analysis and simulation results show that when the synchronization period is less than 100 s, the error of *2LTSP* is within 0.6 ms, no matter how large the size of the network is. Moreover, when the required synchronization error limit (ε) is 9 microseconds, the immanent period of *2LTSP* is larger than or equal to 1300 s in networks of any size. Since only a packet is flooded to the entire network in a synchronization process of *2LTSP* (see Section 4) and its immanent period is long, then it is an energy-efficient protocol. This means *2LTSP* is suitable for long-term running networks.

The rest of this paper is organized as follows. In Section 2, we will give some definitions and assumptions that are important to understand the formulation of our proposal. In Section 3, we review the existing clock synchronization protocols that are directly related to our approach. The proposed synchronization scheme is presented in Section 4. Error analysis is given in Section 5. Section describes the performed simulation experiments. Section 7 describes the implemented prototype of the proposed algorithm. Finally, Section 8 draws the conclusion. In addition, some proofs were included in Appendixes A and B.

2. Concepts and assumptions

This section is organized as follows. In Subsection 1, we present the WSN model and communication frame format adopted in this work. In Subsection 2, we discuss the Quadratic Taylor synchronization. In Subsection 3, we discuss the spatial accumulative effect in WSN. In Subsection 4, we define communication cost model used in this work.

2.1. WSN model and communication frame format

In this work, a WSN is modeled as a graph. The *network graph* $G(N)$ of a WSN N is defined as follows: (1) the vertex set of $G(N)$ is the set of all nodes in network N; (2) for any two vertices A and B, the sufficient and necessary condition for the existence of an edge connecting them in $G(N)$ is that they are within radio range of each other. Hereafter, we will use the terms from graph theory without explaining them. For any two nodes A and B, the distance between them in the network graph is called the *hop distance*. In this paper, we suppose that there exists a unique node (referred to as the *reference node*) such that its hardware clock is the global clock of the network and its logical clock is always synchronized with its hardware clock. The *network radius* is defined as the hop distance between the reference node and the farthest node from it. A network whose radius is equal to *one* is called a *one-hop network*, while a network whose radius is larger than *one* is called a *multi-hop network*. A time synchronization in a one-hop network