Available online at www.sciencedirect.com
ScienceDirect
IFAC Papers Online
CONFERENCE PAPER ARCHIVE
IFAC-PapersOnLine 49-30 (2016) 355–360

Self-Correcting Time Synchronization Support for Wireless Sensor Networks Targeting Applications on Internet of Things

Leandro T. Bruscato*. Edison P. Freitas**
Tales Heimfarth***

** Federal University of Rio Grande do Sul, Porto Alegre, RS Brazil;*
(e-mail: ltbruscato@inf.br).
*** Federal University of Rio Grande do Sul School of Engineering and Informatics InstitutePorto Alegre, RS – Brazil*
(e-mail: edison.pignaton@ufrgs.br)
**** Federal University of Lavras Computer Science Department Lavras, MG – Brazil*
(e-mail:tales@dcc.ufla.br)

Abstract: The Internet of Things is a collection of devices that communicate by exchanging a variety of data among them, in which time synchronization is needed for meaningful information creation and transmission. The robustness of the data transmissions becomes an issue, since most of these devices use wireless communication. This paper focuses in proposing and implementing a time synchronization service for low-power wireless sensor networks using low frequency real-time clocks in each node. This work presents the design, implementation and test of an adaptive algorithm, making the timing of the clocks converge as quickly as possible and after this convergence, keeping them most similar as possible. The goal is to achieve the best method that ensures right timing and still having low energy consumption. Experimental results provide evidence of the success in meeting this goal.

Keywords: **Time synchronization, Wireless sensor networks, Low-power, Latency, Self-Correcting, Clock prediction, Real-time and Internet of things.**

1. INTRODUCTION

Advances in microelectromechanical technology (MEMS) in recent decades made possible the development of small, low cost and low-power devices used as sensors nodes in wireless sensor networks (WSN). WSN based systems are very scalable and flexible, which makes them suitable to observe and monitor various aspects of the physical world in different domains, such as agriculture, military, environment, civilian, health, home (Sundararaman et al. 2005), security surveillance, assisted navigation (Li et al. 2013), Internet of Things (IoT)(Atzori et al. 2010), and many other applications. In order to perform their activities, WSN must communicate and effective communication requires time synchronization among the sensor nodes.

The IoT is rapidly gaining ground in the scenario of modern wireless telecommunications (Atzori et al. 2010). This environment is common to have many devices where the need for a time synchronization method is fundamental. The time synchronization helps to improve energy efficient and reduce collisions. Low power in sensor networks is necessary to make the system scalable. However, communication is a well know power hungry task, which means that it must be efficient to avoid unnecessary retransmissions, which would imply in waste of energy resources. Time synchronization will also help processing and analysis of the data to predict the future system behaviour, which is useful even in the application layer, supporting services such as data fusion. Observing just the aspect of communication among low-power WSN devices, time synchronization is an essential feature. Time coordination allows the schedule of transmissions and synchronization of nodes' duty cycles. This coordination further enhances the energy savings of the entire network (Sundararaman et al. 2005). Currently the main low-power wireless communication protocols, like WirelessHART, ISA100.11a and Zigbee employ duty cycle scheduling switching between active and idling modes (Nixon 2012) (Skrzypczak et al. 2009).

Observing the importance of time synchronization for WSN, thus to IoT, this work studies the main causes of desynchronization of the nodes in WSN and the existing methods to keep time synchronization. Taking these methods as basis, three different synchronization approaches propose combining these methods. The methods are tested using commercial off-the-shelf (COTS) wireless sensor nodes. The performed experiments show that the most complete of the three proposed approaches, the one that combines both self-correction and clock prediction methods having an active (dynamic) timing adjustment, is the one that achieves better results.

The remainder of this paper is organized as follows: Section 2 presents related work. Section 3 discusses the synchronization problem and the existing basic method to handle it. Section 4 presents the three proposed approaches. Section 5 presents the performed experiments while Section 6 the acquired results. Finally, Section 7 concludes the paper providing directions for future work.

2. RELATED WORK

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.11.161

In order to facilitate the study of time synchronization methods, Fengyuan Ren (Ren et al. 2008) divides the proposals into two types: unidirectional broadcast synchronization, the Reference Broadcast Synchronization (RBS) and the Flooding Time Synchronization Protocol (FTSP). In the first group, it is possible to mention the bidirectional pair-wise synchronization, the Timing-Sync Protocol for Sensor Networks (TPSN) and the Lightweight Time Synchronization (LTS). For the second group it is possible to highlight two protocols called TSync and Tiny-Sync and Mini-Sync (TS/MS). These four proposals are detailed described in (Sundararaman et al. 2005) (Ren et al. 2008).

The work of Zhengbao Li (Li et al. 2013) and Jie Zhang (Zhang et al. 2013) focus on the time synchronization in underwater acoustic mobile sensor networks (UAMSNs). They developed and RF-based algorithm (E2DTS) that is bidirectional. In the first phase estimate the clock, skew with the set of reference. Then, in the second phase, it is estimated the delay between unsynchronized nodes. This algorithm implements a method of prediction designing propagation delays. The work of Su Ping (Ping 2003) makes a comparison between the methods Delay Measurement Time Synchronization (DMTS) and RBS and concludes that DMTS presents the best energy consumption performance. The study of Yildirim (Yildirim & Kantarci 2014) proposed an algorithm concerned about network flooding, thus developed a timing method that does not overload the network. It is scalable and assists in energy consumption.

The Diedrichs' and Mirabella,'s work (Diedrichs et al. 2014) (Mirabella et al. n.d.) makes the timing of WSN based on IEEE 802.15.4 with power consumption control. It takes advantage of Carrier sense multiple access with collision avoidance (CSMA-CA). The work of Qu analyzed production logistics synchronization of system under smart cloud manufacturing for IoT based in real-time, focus in Cloud manufacturing. Since the work of Dressler focuses on the timing of co-located network for IoT. Both works used IEEE 802.15.4, RBS and TDMA (Qu et al. 2015) (Dressler & Nordin 2015).

3. TIME SYNCHRONIZATION PROBLEM BACKGROUND

3.1. Models of Clock Synchronization

The main types of clock synchronization are:

1. Global clock: Universal Time Coordinated (UTC) is the time standard commonly used across the world. This is precision global time maintain by atomic clock (PalChaudhuri et al. 2004).
2. Relative clock: This is the relative idea of time within the sensor network. Each node synchronize with every other node (PalChaudhuri et al. 2004).
3. Relative notion of time: The time can also be between nodes, without using real time, but using logical time. This model does not need to match with physical clock (PalChaudhuri et al. 2004).
4. Physical ordering: This model use to order the events of process in system (PalChaudhuri et al. 2004).

Most models should consider whether errors could occur, to adapt and improve synchronism of the system.

3.2. Time Synchronization Methods

In general, sensor clocks are based on crystal oscillators, which provide a local time for each network node, and, as has already been said, there are difference between crystal clocks. For a perfect clock, the derivative dC(t)/dt should be equal to 1. The clock skew can vary, but we assume that it stays bounded and might be between the following values:

$$1 - \rho \leq \frac{dC(t)}{dt} \leq 1 + \rho \quad (1)$$

where ρ denotes the maximum skew rate(Rhee et al. 2009). If time of a clock in a sensor node A is $C_a(t)$, where $C_a(t) = t$ for a perfect clock and the clock frequency is the rate at which a clock progresses. Clock offset is the difference between the times reported by the clocks at two sensor nodes. Clock skew is defined to be the difference in the frequencies of two clocks $C_A{}^0(t)$ - $C_B{}^0(t)$ (Rhee et al. 2009). The figure 1 represent fast, slow, and perfect clocks.

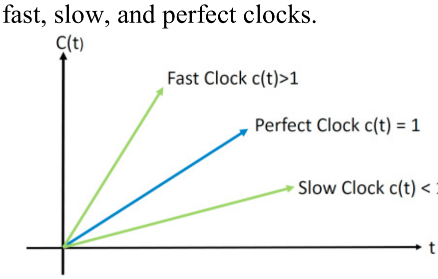

Fig. 1 Example of fast, slow, and perfect clocks

3.3. Self-Correction

The self-correction is a method in which a sensor node uses the received clock value from a time server node to approximate its own clock.

A synchronization algorithm that adapts the relationship between the clocks is faster convergence of each type of device. In Figure 2, it is possible to observer the operation of an adaptive synchronization method. In the first transmission, Msg 1, the time difference is greater than in the second transmission, Msg 2. This means that the receiving device has changed its clock to become closer to the reference one.

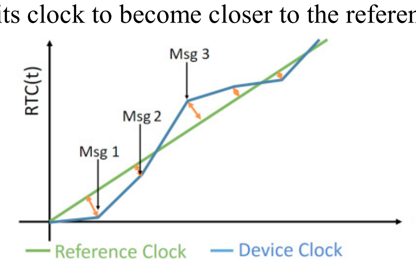

Fig. 2 Reference clock and Device's clock

In addition to this method, there is also an auto correction along the time for the equation 2 if a fine-tuning method enable. The result of this ΔClock cannot exceed a certain value. The positive or negative signal result depends on whether the device is late or early in relation to the main one. Thus, the idea is to keep the curves as most asymptotic as possible, diminishing the angle between the lines representing the