Automatica 59 (2015) 60–72

Contents lists available at ScienceDirect

Automatica

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

Clock synchronization protocol for wireless sensor networks with bounded communication delays ☆

Emanuele Garone [a], Andrea Gasparri [b], Francesco Lamonaca [c]

[a] Faculty of Applied Science, Control and Systems Analysis Department, Université Libre de Bruxelles, 1050 Brussels, Belgium

[b] Department of Engineering, Roma Tre University, Via della Vasca Navale 79, 00146 Rome, Italy

[c] Department of Computer Science, Modeling, Electronic and Systems, University of Calabria, Ponte P. Bucci, 87040, Arcavacata di Rende, Italy

ARTICLE INFO

Article history:
Received 26 May 2014
Received in revised form
2 February 2015
Accepted 21 May 2015

Keywords:
Asynchronous consensus algorithm
Clock synchronization
Noise robustness
Delay measurements
Wireless sensor networks

ABSTRACT

In this paper, we address the clock synchronization problem for wireless sensor networks. In particular, we consider a wireless sensor network where nodes are equipped with a local clock and communicate in order to achieve a common sense of time. The proposed approach consists of two asynchronous consensus algorithms, the first of which synchronizes the clocks frequency and the second of which synchronizes the clocks offset. This work advances the state of the art by providing robustness against bounded communication delays. A theoretical characterization of the algorithm properties is provided. Simulations and experimental results are presented to corroborate the theoretical findings and show the effectiveness of the proposed algorithm.

1. Introduction

A Wireless Sensor Network (WSN) consists of a collection of nodes deployed within an environment to perform a given task. Each node is typically equipped with a radio transceiver, a microcontroller and a set of sensors. Nodes collaborate in order to reach a common goal. WSNs are at the forefront of emerging technologies due to the recent advances in Microelectromechanical Systems (MEMSs). The inherent multidisciplinary nature of WSNs has attracted scientists coming from different research areas, from networking to robotics. Their application ranges from surveillance and coverage (Gasparri, Krishnamachari, & Sukhatme, 2008; Ghataoura, Mitchell, & Matich, 2011), structural health monitoring (Harms, Sedigh, & Bastianini, 2010; Torfs et al., 2013), and industrial process control (Hancke, 2013; Ramamurthy, Prabhu, Gadh, & Madni, 2007) to emergency response (Fischer & Gellersen, 2010; Lorincz et al., 2004) and mobile target tracking (Olfati-Saber & Jalalkamali, 2012; Xu, Ding, & Dasgupta, 2013). Most of these applications require basic services such as self-localization (Gasparri & Pascucci, 2010; Song, Shin, & Jeon, 2012), time synchronization (Lamonaca, Gasparri, Garone, & Grimaldi, 2014; Schenato & Fiorentin, 2011), and topology control (Aziz, Sekercioglu, Fitzpatrick, & Ivanovich, 2013; Williams, Gasparri, & Krishnamachari, 2014). However, the distributed nature and the limited hardware capabilities of WSNs make the development of these applications and related services particularly challenging.

In this work, we advance the state of the art by providing a solution to the clock synchronization problem in the presence of bounded communication delays. Inspired by the Average TimeSynch Protocol introduced in Schenato and Fiorentin (2011), we propose a novel synchronization protocol, denoted as Robust Average TimeSynch (RoATS), to adjust both the nodes' clock frequency and clock offset in a robust way with respect to bounded communication delays. A preliminary version of the RoATS was presented in Garone, Gasparri, and Lamonaca (2013). This paper considerably improves that preliminary results in several directions. In particular, the drift and offset compensation are now considered as two independent and asynchronous processes. As a result, the convergence analysis has been revised and more meaningful theoretical bounds have been obtained. Finally, experiments have been carried out to corroborate the theoretical finding in a real-world environment.

☆ This work was partially supported by the Italian grant FIRB "Futuro in Ricerca", project NECTAR, number RBFR08QWUV, funded by the Italian Ministry of Research and Education (MIUR) and by the Italian Ministry of Economic Development by means of the RIDITT: "Italian Network for Innovation and Technology Transfer to Enterprises", under the project "DI.TR.IM.MIS". The material in this paper was partially presented at the 2013 American Control Conference, June 17–19, 2013, Washington, DC, USA. This paper was recommended for publication in revised form by Associate Editor Jun-ichi Imura under the direction of Editor Toshiharu Sugie.

E-mail addresses: egarone@ulb.ac.be (E. Garone), gasparri@dia.uniroma3.it (A. Gasparri), flamonaca@deis.unical.it (F. Lamonaca).

http://dx.doi.org/10.1016/j.automatica.2015.06.014

2. Related work

Clock synchronization is an important problem in the context of distributed systems. This problem has become particularly relevant with the introduction of the Internet, as large networks of connected computers became more and more common. In this context, the most famous protocol is the Network Time Protocol (NTP) introduced in Mills (1991). NTP was designed for large-scale networks with a rather static topology (such as the Internet). In NTP nodes are externally synchronized to a global reference time that is injected into the network at many places via a set of master nodes. Master nodes are synchronized out of band, for example via GPS, and diffuse the notion of time to the other nodes by following a hierarchical scheme. Major limitations concerning the application of NTP to WSNs are the highly dynamic nature of the network topology, the limited bandwidth, and the necessity of a completely decentralized architecture to ensure robustness and flexibility. In the past years, several algorithms have been designed to deal specifically with the typical WSNs requirements, such as low energy consumption, bandwidth constraints and long-term operation. Several surveys concerning the clock synchronization problem in WSNs can be found in the literature. Among the others, it is worth mentioning (Sundararaman, Buy, & Kshemkalyani, 2005), where a comparison of different synchronization protocols is carried out on the basis of a palette of performance indexes, e.g., precision, accuracy, cost, and complexity. A more recent survey can be found in Wu, Chaudhari, and Serpedin (2011), where the latest advances in the field of clock synchronization of WSNs are reported from a signal processing perspective. Apart from seminal approaches such as the Reference-Broadcast Synchronization (RBS) (Elson, Girod, & Estrin, 2002) and the Timing-sync Protocol for Sensor Networks (TPSN) (Ganeriwal, Kumar, & Srivastava, 2003), relatively few protocols taking into account noisy measurements and delays have been proposed. In Carli, Chiuso, Schenato, and Zampieri (2011), the authors introduce a protocol to synchronize a network of controlled discrete-time double integrators which are nonidentical, with unknown model parameters and subject to additive measurement and process noise. In Liao and Barooah (2013), by assuming the dynamic nature of the network to be modeled as a Markov chain, the authors propose a distributed algorithm for the estimation of scalar parameters from noisy relative measurements. In particular, they prove that the estimates are mean square convergent under fairly weak assumptions. In Chaudhari, Serpedin, and Qaraqe (2008), the joint Maximum-Likelihood Estimation (MLE) of the clock offset and skew is introduced assuming an exponential delay model. In Leng and Wu (2010), the authors derive three clock-synchronization algorithms for WSNs under unknown delays. In Kim, Lee, Serpedin, and Qaraqe (2011), a clock synchronization algorithm, called the Iterative Gaussian mixture Kalman particle filter (IGMKPF) is introduced. Briefly, this combines the Gaussian mixture Kalman particle filter (GMKPF) with an iterative noise density estimation procedure to achieve robust performance in the presence of unknown network delay distributions.

Recently, the design of completely decentralized synchronization algorithms based on the consensus approach has gained momentum. Along this line, in Carli, Chiuso, Schenato, and Zampieri (2008) an algorithm based on a PI-like consensus protocol has been introduced, where the proportional (P) part compensates the different clock frequencies while the integral part (I) eliminates the different clock offsets. In Schenato and Fiorentin (2011), a distributed clock synchronization protocol, referred to as Average TimeSync (ATS), has been introduced. This protocol, which will be detailed in Section 4, is based on the cascade of two consensus algorithms aiming at synchronizing the clock drift and offset, respectively. In Maggs, O'Keefe, and Thiel (2012), a consensus-based protocol aiming at reducing the clock error between geographically closely located nodes has been proposed. In Lamonaca et al. (2014), the clock synchronization problem for event-driven measurement applications is addressed. In particular, the authors propose a consensus-based protocol which allows to achieve high accuracy in the area where an event is detected and ensure long network lifetime.

Compared to the state of the art, this paper provides a novel robust protocol, the RoATS, with provable guarantees on the synchronization accuracy against bounded communication delays. This protocol consists of two asynchronous consensus-based algorithms for the synchronization of the clocks frequency and offset. Their convergence properties are theoretically characterized and experimentally validated by means of a WSN composed of TelosB nodes (Crossbow technology, 2004).

3. Problem statement

Consider a wireless sensor network composed of N nodes and assume that the network topology is described by means of an undirected connected graph $\mathcal{G} = (V, E)$, where $V = \{1, \ldots, N\}$ is the set of vertices representing the sensor nodes and $E = \{(i, j)\}$ is the set of edges describing the point-to-point channel availability. Namely, an edge (i, j) exists if node i can transmit a packet to node j. Note that, since the network topology is assumed undirected the existence of an edge (i, j) implies the existence of the edge (j, i). Communication between pairs of nodes is assumed to be asynchronous.

Each node i is equipped with a local *hardware* clock τ_i defined as:

$$\tau_i(t) = \alpha_i t + \beta_i, \tag{1}$$

where $\alpha_i \in [\alpha_{\min}, \alpha_{\max}]$ is the local clock frequency and β_i is the local clock offset. Coefficients (α_i, β_i) may differ for each node due to construction imperfections and different operational conditions, e.g., different temperatures for the quartz oscillators. Thus, in the absence of corrections, the notion of time among the nodes may quickly diverge. To address this issue, each node is provided with a tunable *software* clock $\hat{\tau}_i(t)$ defined as:

$$\hat{\tau}_i(t) = \hat{\alpha}_i(t)\, \tau_i(t) + \hat{o}_i(t), \tag{2}$$

where $\hat{\alpha}_i(t)$ and $\hat{o}_i(t)$ are scalar parameters that can be used to adjust the ith clock frequency and offset, respectively.

The objective of the synchronization problem is to adjust these parameters to eventually achieve a common sense of time in the software clock of the nodes, that is:

$$\lim_{t\to\infty} [\hat{\tau}_i(t) - \hat{\tau}_j(t)] = 0, \quad \forall i, j \in V. \tag{3}$$

Clearly, perfect synchronization is achievable only in the ideal case of instantaneous transmission. Differently, in a more realistic scenario where (random) bounded transmission delays may occur, the synchronization objective is to ensure that the difference between the nodes' clock remains bounded.

4. The ATS protocol

In this section, for the reader's convenience and for the sake of comparison, the main aspects of the ATS protocol proposed in Schenato and Fiorentin (2011) are briefly reviewed. In the ATS, for each update k only one directed arc $e_k = (j, i)$ of $\mathcal{G}$ is involved. This implies that a single packet from node j to node i is sent. This packet contains the tuple $(\mathrm{id}_j, \hat{\alpha}_j, \hat{o}_j, \tau_j)$, where id_j is the ID of the jth node, $\hat{\alpha}_j = \hat{\alpha}_j(k-1) = \hat{\alpha}_j(t_k)$ and $\hat{o}_j = \hat{o}_j(k-1) = \hat{o}_j(t_k)$ are the parameters of its local software clock at time t_k, and $\tau_j = \tau_j(t_k)$ is the hardware timestamp of the packet, i.e. the value of the hardware clock of node j at the moment the message is sent. Upon packet reception, node i performs a timestamp of its