Measurement 56 (2014) 203–214

Contents lists available at ScienceDirect

Measurement

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

Comparison of approaches to time-synchronous sampling in wireless sensor networks

Jürgen Funck *, Clemens Gühmann

TU Berlin, Chair of Electronic Measurement and Diagnostic Technology, Sekr. EN 13, Einsteinufer 17, D-10587 Berlin, Germany

A R T I C L E I N F O

Article history:
Received 18 November 2013
Received in revised form 11 June 2014
Accepted 7 July 2014
Available online 12 July 2014

Keywords:
Wireless sensor network
Time synchronization
Sampling
Signal distortions

A B S T R A C T

Time synchronization is required in many data acquisition applications for sensor networks. A large variety of time synchronization protocols for wireless sensor networks has been suggested. Yet, setting up a sensor network for synchronized data acquisition based on those algorithms is not a trivial task. This paper outlines different approaches to time synchronous sampling in wireless sensor networks. The approaches are implemented on a wireless sensor network. Experiments are done to compare the signal quality of the signals acquired with the different approaches in terms of frequency error, signal-to-noise ratio (SNR) and total-harmonic-distortion (THD). Furthermore, the approaches are compared in terms of the effort required for communication and implementation.

1. Introduction

Time synchronization is required in many data acquisition applications for sensor networks. In general it allows for the mapping of measurements made by different sensor nodes onto a common timescale, thus enabling the ordering of events and detection of causalities [1]. This is especially important when identifying the dynamics of a physical system, e.g. the relationship between temperature, rotational speed and torque at an automotive clutch [2]. Other applications where synchronized data acquisition is important are the monitoring of power systems by synchronized phasor measurements [3], or localization by time of flight measurements [4]. Even in not very time-critical applications like environmental monitoring [5] synchronized data acquisition may be desired to get a consistent state of the area under observation.

The desired synchronization accuracy is highly dependent on the application. For dynamic system identification the maximum synchronization error should be significantly below the smallest relevant time constant of the system under investigation. This can range from the order of seconds to the order of micro-seconds. For synchronized phasor measurements synchronization requirements of the order of 10 μs have been reported [3]. In localization application the synchronization accuracy if often desired to be in the range of 1 μs or even below that [4,6]. In environmental monitoring even synchronization accuracies in the order of minutes may be sufficient in some applications.

In recent years intensive research has been done on the synchronization of clocks in wireless sensor networks. Summaries of the state-of-the-art protocols are given in [7–9]. Descriptions of individual time synchronization protocols can be found in [6,10,11]. Publications on synchronized sampling using wireless sensor networks are, however, much rarer. Yet some examples can be found [4,5,12].

When setting up a wireless sensor network for time synchronous data acquisition the design goals usually are to minimize the latency of data transmission while maximizing the networks availability as well as the quality and integrity of the data. Common challenges are the

* Corresponding author.
E-mail addresses: juergen.funck@tu-berlin.de (J. Funck), clemens.guehmann@tu-berlin.de (C. Gühmann).

http://dx.doi.org/10.1016/j.measurement.2014.07.001
0263-2241/© 2014 Elsevier Ltd. All rights reserved.

changing quality of wireless links over time [13] as well as the limited capacity of the batteries that often power the sensor nodes. Thus usually a tradeoff has to be found between the latency and data quality that benefit from frequent communication and the network availability that is mainly determined by the amount of energy used for communication and computation. In this context it is interesting to note that in a wireless sensor network the energy cost of computation is generally small compared to the cost of communication [14]. The software for wireless sensor nodes should be efficient and have a small memory footprint since wireless sensor nodes are usually equipped with microprocessors that are energy efficient but have little computational power and memory. Furthermore, the software should be reliable and easily extendible to ease its adaption to new sensing applications. This is best achieved by a modular software design with little or no interdependencies between the individual software modules.

This article is an extension of our paper [15] presented at the 19th IMEKO-TC4 Symposium. It deepens the treatment of clock errors and their effects on the acquired signals and introduces methods to measure them. The description of a practical implementation as well as experimental results and their analysis have been added. The analysis of the two presented approaches has been intensified and updated based on the experimental findings.

This article is organized as follows: Section 2 gives a brief introduction into the communication and synchronization protocols relevant for this paper. In Section 3 a wireless sensor network is modeled as a generic multi-channel sampling system and two different approaches to synchronized sampling are outlined. An overview of the effects of synchronization errors and clock updates by the synchronization protocol on the acquired signals is given in Section 4. Section 5 presents the implementations of the approaches as well as the experiments done to compare them. In Section 6 the experimental results and theoretical properties of the approaches are discussed. Finally Section 7 draws conclusions from this discussion and outlines the direction of future research on this topic.

2. Communication and synchronization protocols for wireless sensor networks

Communication within wireless sensor networks is organized by a multitude of protocols. For every task a variety of protocols has been suggested. Good overviews of the protocols for wireless sensor networks can be found in [9,16].

The Contiki operating system [17] is optimized for the use in wireless sensor networks. It includes amongst other things a complete IP-based networking stack and a cooperative multi-tasking environment. It is used as a platform for practical implementations in this paper.

This section briefly introduces the two classes of protocols most relevant to the synchronous acquisition of measurement data. Protocols for the synchronization of wireless sensor nodes are presented in Section 2.1. Section 2.2 discusses the most relevant medium access (MAC) protocols.

2.1. Synchronization protocols

Synchronization protocols aim to synchronize the clocks of sensor nodes which differ in offset and drift. The offset is defined as the difference ΔT between two clocks i and j at the same point of time [8]:

$$\Delta T = t_i(t) - t_j(t) \tag{1}$$

The drift ρ is defined as the difference between a clock's rate f and the ideal rate of 1 [8]:

$$\rho = f - 1 \tag{2}$$

Many synchronization protocols only correct the clock offset. In this case the synchronization has to be repeated periodically to keep the synchronization error within a desired interval. The minimum length of the synchronization period can be calculated from the clock's drift [8,9].

In literature [8,9] two classes of synchronization protocols are described: a priori and a posteriori. A priori synchronization protocols continuously synchronize the clocks of the nodes in a network, so that every node always has an estimate of the network's global time. A posteriori protocols only synchronize the clocks once an event occurs, i.e. the estimate of the event's time of occurrence on the network's global timescale is calculated only after the event has occurred. An important limitation of a posteriori synchronization protocols is that they cannot be used to coordinate actions between the nodes, e.g. trigger actions on different nodes at the same time [10].

Well known examples of a priori synchronization protocols are the Flooding Time Synchronization Protocol (FTSP) [6] and the Timing-sync Protocol for Sensor Networks (TPSN) [11]. The Contiki operating system implements a time synchronization protocol very similar to FTSP named `timesynch`. One reference node broadcasts its current time in regular intervals. Based on the received timestamp and their local time at reception the receiving nodes calculate the offset ΔT_i between their local clock $t_{local,i}$ and that of the reference node $t_{local,0} = t_{global}$. Thus they obtain an estimate of the synchronized network time.

$$t'_{global,i} = t_{local,i} + \Delta T_i \tag{3}$$

Once a node is synchronized it starts broadcasting the synchronized time as well. This way synchronization over multiple-hops can be achieved.

An example of an a posteriori synchronization protocol is the Routing Integrated Time Synchronization (RITS) [10]. It is based on the Elapsed Time on Arrival (ETA) primitive introduced in [10]. Its key idea is to transmit the time that has passed since an event instead of the event's timestamp. Thus the receiving node can calculate the timestamp of the event on its local timescale by subtracting the elapsed time from its local reception timestamp (see Fig. 1).

Within RITS this procedure is repeated on every node that a packet is routed through.

A recommendation made for a priori [6] as well as a posteriori [10] algorithms is, that to achieve a high synchronization accuracy all timestamps should be generated as close as possible to the sending and receiving operations, preferably in the radio driver. The reason for this is