Int. J. Electron. Commun. (AEÜ) 82 (2017) 446–457

Contents lists available at ScienceDirect

Int. J. Electron. Commun. (AEÜ)

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

Regular paper

Clock synchronization in wireless sensor networks using least common multiple

Nikhath Tabassum*, Geetha D. Devanagavi, Rajashekhar C. Biradar

REVA University, Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Bangalore 560 064, India

A R T I C L E I N F O

Keywords:
Wireless sensor network
Clock synchronization
Clock skew
Clock offset
Cluster

A B S T R A C T

In this research, we propose the Clock Synchronization by Least Common Multiple (CSLCM) method to remove the clock offset and clock skew among the sensor nodes. The proposed CSLCM enables the nodes to reach a network synchronization time by calculating the least common multiple of their Clock Time Period (CTP). The network is organized into clusters and every node reaches the network synchronization time using its own CTP. Simulation results show that, the CSLCM algorithm is more efficient compared to the Average Time Synchronization with Pairwise messages (ATSP) in terms of accuracy, communication overhead, and computation overhead.

1. Introduction

The capability of WSN to track, monitor, detect an event, and aggregate data, makes it very suitable for realizing various functionalities in a smart city. To name a few applications of the WSN in the smart city, WSN is used for monitoring the structural health of the monuments, bridges and skyscrapers [1], for smart management of solid waste [2], for detecting air pollution levels [3], for smart vehicle login in the smart city, traffic control and smart parking [4–6], and for smart metering to reduce power consumption in every household of the smart city [7]. The basic component of the WSN is the sensor node, which collects the desired data from its environment periodically or during an event. It processes the data and communicates with other nodes in the network, for executing a process. The data aggregation from the sensor nodes can be done efficiently if public transportation vehicles are used [8] to carry the mobile sinks or by using the shortest path between the sensor node and the mobile sink [9], where the latter is used to collect data from the sensor nodes. For most of the applications, the clock time is used to know, (a) The exact moment of time, the event had occurred in the network, (b) The duration between two distinct events and (c) The sequence of the events that occurred in the network.

Each sensor node has an inbuilt clock, that is set to a common time before the deployment. After the deployment of the sensor nodes, their clock time might change during the operation due to different climatic conditions like temperature and humidity, as the sensors may not experience the same conditions, when they are deployed over a larger geographical area. There may be variations in the crystal oscillators due to manufacturing process and variations may result due to aging as well. Therefore for collision free data transfer and for real time data monitoring,the sensor nodes should be well synchronized in time with their inbuilt clocks aligned to same time.

The various protocols for clock synchronization are [10–28]. Some of the protocols have an external reference clock to which all the nodes synchronize [10–14]. In these protocols, either the hardware clock or virtual clock is modified to synchronize with the network synchronization time. Each time the clock value is updated at a node, there is an inherent error, which is passed on to other nodes as this propagates, resulting in greater accumulative error by the time the network synchronization is reached. Some protocols [22–28] modify the clock recursively to achieve synchronization. These protocols converge to the network synchronization time with greater accuracy and lesser accumulative effect but with higher computation overhead. Another set of protocols that do not use external or internal reference time are the consensus based protocols [15–21]. The network synchronization time is set, based on the consensus between the nodes. The network synchronization time could be set to either the fastest clock of the network or to the average time of the nodes in a network. One such protocol, Average Time Synchronization with pairwise message exchange (ATSP) is compared with CSLCM. The consensus based protocols are efficient, but have greater communication overhead.

In this paper, we propose the Clock Synchronization using Least Common multiple (CSLCM) protocol to synchronize the sensor node clock time period CTP. In our proposed method, we do not modify the hardware clock time and do not have a reference clock to which the network synchronizes, so the accumulative error is nil. The formation of clusters and the application of CSLCM, reduces the communication

* Corresponding author.
E-mail addresses: nikhath005@gmail.com (N. Tabassum), dgeetha@reva.edu.in (G.D. Devanagavi), dir.ece@reva.edu.in (R.C. Biradar).

http://dx.doi.org/10.1016/j.aeue.2017.10.014
Received 19 June 2017; Accepted 9 October 2017

overhead to a great extent. The main objective of our protocol is to remove the clock offset and clock skew among the nodes using the CSLCM. In CSLCM, every node calculates the Least Common Multiple (LCM) of all the nodes in a network. The LCM value is the network synchronization time. The step size for every node is its own CTP. The synchronization is carried at two levels, Intracluster synchronization and Intercluster synchronization as described in Sections 3.2.2 and 3.2.3 respectively. Every node takes finite, integral number of steps to reach the network synchronization time. So all the nodes if they start at the same time to take their respective steps, they will converge to a common time (LCM) at the same instant.

1.1. Clock preliminaries in WSN

Consider the clock of a sensor node V_i having a clock skew of α_i and clock offset of β_i from the initial clock $x_i(0)$ given by Eq. (1). The Eq. (1) is derived from local hardware clock equation in [19]. The $x_i(t)$ is the CTP of a node V_i.

$$x_i(t) = \alpha_i x_i(0) + \beta_i \quad (1)$$

The clock offset is the constant difference in time between the clocks. Let us assume that all the clocks were initially set to a common CTP $x(0)$ at $x = 0$. Due to change in temperature, humidity and manufacturing process the clocks drift in frequency and phase. The constant difference in phase of the clocks is the clock offset β and the difference in frequency is clock skew α. In Fig. 1 we have two nodes V_1 and V_2 with their CTPs denoted by $x_1(t)$ and $x_2(t)$. Let us assume that $x_1(t)$ and $x_2(t)$ have same frequencies

$$\alpha_1 x_1(0) = \alpha_2 x_2(0)$$

but different clock offset as shown in Fig. 1. Therefore the clock offset between the two nodes is given in Eq. (2).

$$Clock\ offset = |x_1(t) - x_2(t)| = |\beta_1 - \beta_2| \quad (2)$$

If the clock offset $\beta_1 = \beta_2$ is the same for the two nodes then the clock skew is given by

$$Clock\ skew = |x_1(t) - x_2(t)| = |\alpha_1 x_1(0) - \alpha_2 x_2(0)| \quad (3)$$

The clock skew is depicted in Fig. 2.

The organization of the paper is as follows. The related literature work is discussed in Section 2, the system model and the synchronization is discussed in Section 3, CSLCM for different topologies in Section 4, simulation and results in Section 5 and conclusion in Section 6.

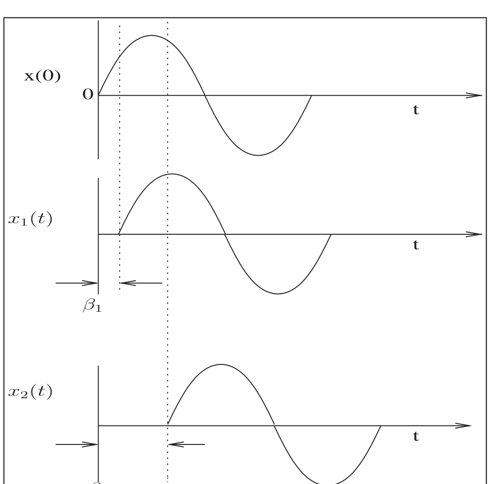

Fig. 1. Clock offset between the nodes.

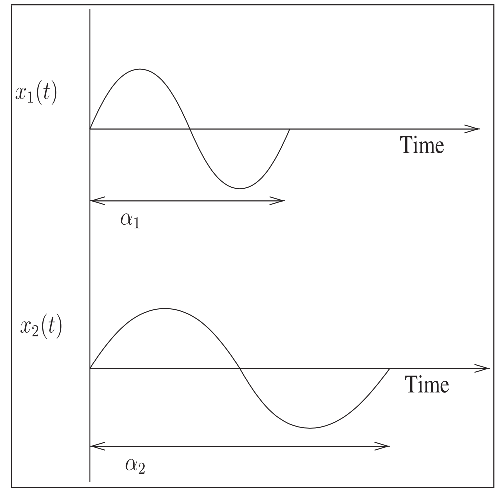

Fig. 2. Clock skew between the nodes.

2. Related work

Some of the clock synchronization protocols are discussed in this section. We have categorized the protocols based on

(a) The Clock value propagation
(b) The network model they follow
(c) The type of participation of the nodes.
(a) The synchronization based on clock value propagation can be classified into *Progressive synchronization* and *Consensus Synchronization*. In the latter, all the nodes agree to synchronize to a particular value. All the nodes may align to the fastest clock among the nodes or to the average of the clocks in the network. Whereas the former method of synchronization has the clock value passed on among the nodes without alteration. A reference node initiates the synchronization process and the reference node's clock value is propagated to other nodes.
The Progressive synchronization can be further classified into *Centralized progressive method* wherein the network topology is in the form of a tree with all the nodes having hierarchy. In *Decentralized progressive method* the topology is either a chain or a mesh network and there is no hierarchy.
LCHP [10] is a *Decentralized progressive method* type of protocol. In this protocol, time synchronization takes place using the fact that the time between two transmitting pulses from a transmitting node is same as the time between the two receiving pulses at a receiving node, assuming constant propagation delay between the nodes. If the propagation delay is constant, due to accumulative effect the protocol is accurate up to 9 hops. Therefore it is not suitable for larger networks. For larger networks, the 2LTSP [11] is suitable. In this protocol, a reference node floods the network with the synchronization packet. The neighbors follow this reference clock time and synchronize with their neighbors using 3 time stamps and 2 flags. This protocol is robust to node failure. The hardware clock is taken to follow the temperature based drift model, the synchronization period should not be large so as to change the temperature above 1 °C. The communication overhead is larger for this protocol. In AOTSP [12], only a 'time transformation function' is used to map the local time of the node to the global time of the reference node. This mapping parameters are captured in *eta* (elapsed time on arrival) which is the time difference between the global time and the node's local time stamp value and *tt* field which measures the time taken by the node to transmit the *eta* field to its neighboring node. Though it has low communication overhead, the scalability is poor due to high accumulative effect.