A new classification of clock Synchronization in Wireless Sensor Networks

Habib Aissaoua[1], Makhlouf Aliouat[1], Abdelkader Laouid[3,4], and Farid Lalem[2]

[1]Department of Computer Science, University of Ferhat Abbas Setif 1, Setif, Algeria
[3]Department of Computer Science, University of Abderrahmane Mira, Bejaia, Algeria
[2]Lab-STICC CNRS laboratory, University of Western Brittany, Brest, France
[4]Department of Computer Science, University of El Oued, El Oued, Algeria

Abstract. Since the birth of computer networks and distributed systems, the clock synchronization issue has been considered as a major challenge. In fact, the importance to have clocks synchronized resides in the fact that a wide range of communication protocols and applications necessitate a common notion of time for the overall network nodes to operate perfectly. MAC protocols and node sleep scheduling are protocol examples based on synchronized clocks. The design of synchronization protocols is challenged by the uncertainty arising from message latency and clock drifting, and also by resource constrained nodes. In this work, we first exhibit the basic concepts of the clock synchronization issue, and we provide a thorough classification based on the features that have to be considered when designing synchronization protocols in such networks.

Keywords: Wireless Sensor Networks, Clock synchronization, Time drift, Message delay uncertainty, Accuracy

1 Introduction

Wireless sensor networks (WSNs) are widely studied both by academic and industry researchers in recent years [1]. There are various applications to derive benefit from WSNs' services: environmental monitoring, health monitoring, inventory location monitoring, objects tracking, and transportation network monitoring, etc. Unlike classical networks, which aim to provide high quality service without power restriction, sensor network protocols' primarily purpose is to economize as much as possible the energy in order to enhance the network's lifetime. Indeed, synchronization is a critical component for any WSN. Examples are given by data fusion, coordination of wake-up and sleeping times, scheduling algorithms such as TDMA, localization schemes, etc. [4, 10, 12, 2]. During the last decades, clock synchronization has been taken as a decisive key for many wired network applications; and thus attracted intensive interest by researchers with the result that numerous synchronization schemes have been designed for wired networks. However, these schemes cannot be directly used in WSNs due to their specific requirements. In fact, the design of clock synchronization protocols for WSNs is challenging because it is typified by various network constraints. Detailed discussions on clock synchronization challenges in WSNs can be found in [11, 5, 4].

In this work, we propose a new straightforward classification that includes most of the existing clock synchronization schemes for WSNs. Indeed, features of clock synchronization schemes that we have proposed have been used in some research works, but we have introduced other features such as terrestrial vs. underwater network synchronization, sttatic vs. dynamic skew compensation, receivers to receivers vs. one-receiver to receivers synchronization, have not been involved in any works.

This paper is organized as follows. In Section 2, some basic concepts related to clock synchronization and the problem definition are presented. In Section 3, a classification for clock synchronization protocols is provided. Section 4 concludes the paper.

2 Basic concepts and problem definition

2.1 Basic concepts:

Before going into the details of our work, we first define the clock model that will be used in this paper. Each sensor node i has its own local hardware clock consisting of an oscillator frequency that defines the rate f at which the clock progresses. As it is widely adopted, we associate with each node i a local clock $\tau_i(t)$ in order to implement an approximation of the hardware clock. Consequently, the local clock of a node i can be modelled as follows:

$$\tau_i(t) = a_i t + b_i \tag{1}$$

where a_i is the clock *speed (rate)*, and b_i is the local clock *offset* at real time t. Assuming that $\tau_i(t)$ and $\tau_j(t)$ are the local clocks of nodes i and j, respectively, we can define the relative clock between them as:

$$\tau_i(t) = \hat{a}_{ij}\tau_j(t) + \hat{b}_{ij} \tag{2}$$

The parameters $\hat{a}_{ij}$ and $\hat{b}_{ij}$ represent the relative rate and the relative offset between the clocks of node i and j, respectively.

2.2 Problem definition:

WSNs consist of spatially separated nodes, of wireless communication links, and of clocks. Indeed, sharing a common notion of time among network nodes is one of the most basic services in any such network. To achieve this purpose, each network node has to request the time information from its neighbor nodes or from a reference node that provide the accurate time. The procedure applied to assist the entire node clocks to agree on common time is well known as *clock synchronization.* Ideally, the purpose of any clock synchronization algorithm is to ensure the *exact* time between all network nodes. In order to be practical, the *exactness* term has been relaxed by researchers, meaning that the clock synchronization purpose is to maintain the nodes' clock values quite closely synchronized during the network's lifetime. Obviously, two clocks C_i and C_j are declared to be δ-synchronized if their offset value at a time instance t is bounded by some specific constant δ [6]:

$$|C_i(t) - C_j(t)| \leq \delta \qquad (4)$$

Inequality (4) states that at any real time, network node clocks may differ by at most δ, which is the goal of clock synchronization algorithms.

3 Clock Synchronization classification

The clock synchronization in WSNs has been extensively studied in recent years and many protocols have been proposed. Our paper suggests a helpful classification introducing new features that are not covered by these works. For that purpose, we adopt the following classification to categorize the clock synchronization protocols proposed in the literature:

- *Network features*
- *Objective features*
- *Communication technique features*
- *Clock behavior features*

3.1 Network features:

We have divided these features into three categories:

Terrestrial vs. underwater network synchronization:

In general, as terrestrial wireless sensor motes work on high speed radio communication, most of the clock synchronization protocols have been designed with the assumption that the propagation delays among sensor nodes is negligible. However, these protocols are unsuitable for underwater sensor networks (UWSNs) due to the slow propagation speed that characterizes the underwater environment [7]. Consequently, time propagation is the main challenge when designing synchronization schemes for underwater network.

Static vs. dynamic network synchronization:

Nodes in a static network don't move, and topology remains unchanged once the nodes are deployed. However, nodes in a dynamic network have the ability to move, which leads to frequent topology changes. For that, performing clock synchronization in dynamic WSNs is quite challenging compared to static WSNs.

Single hop vs. multi-hop network synchronization:

In single hop networks, nodes can communicate directly between them since nodes are neighbors of each other. Thus, a simple message exchange between nodes would facilitate all nodes to synchronize. However, the multi-hop transmission is one basic feature of WSNs due to the low power transmission level used by nodes, and due to their potential large size. Thus, nodes are not required to be in communication distance with the reference node. Otherwise, each node can use its neighboring nodes as relays to reach the reference node.

3.2 Objective features:

We can divide these features as follows:

Reference-based vs. distributed synchronization:

In a reference-based synchronization protocol, one node needs to be elected as a reference node which acts as reference clock. Then, all non reference nodes will be synchronized to the reference node by adjusting their own clocks based on the clock information received from their reference node. By contrast, nodes in distributed synchronization protocols perform exactly the same algorithm, aiming to converge their local clocks to a common global time value.

Internal vs. external synchronization

The external synchronization objective is to synchronize all node clocks within the network to a real world time provided from outside the network (e.g., UTC, GPS). On the other hand, an internal synchronization purpose is to minimize the error among the local clocks of the nodes in the network. Here, the actual real time is not required, meaning that consistency among the network node clocks is sufficient.

Proactive vs. reactive synchronization:

In proactive synchronization, time synchronization protocols strive to keep the node clocks synchronized at all times. To this end, synchronization should be performed continuously on all nodes, and thus, whenever an event of interest occurs, these clocks can instantly be consulted. In reactive synchronization schemes, a node clock runs unsynchronized, and the synchronization process is started only after an event of interest has been detected [8].

3.3 Communication technique features:

There are three renowned timing message approaches for clock synchronization in WSNs which are:

Round Trip Synchronization (RTS)

In order to communicate the time information between network nodes, the RTS mechanism has been widely used in the literature. To understand this approach, let us suppose that node A needs to synchronize with node B. As depicted in *Figure* 1, during each synchronization round i, a synchronization message containing the sending time t_i^1 is sent to node B by node A. Next, node B records the reception of the message using its local clock at t_i^2 and then replies to Node A at t_i^3 . The reply message holds the timestamp t_i^2 and t_i^3. At last, node A receives the reply message at t_i^4. Note that t_i^1 and t_i^4 are the timestamp recorded by the local clock of node A, while t_i^2 and t_i^3 are the timestamp recorded by the local clock of node B. If we consider only the clock offset between two adjacent nodes, the above procedure can be mathematically modelled as:

$$t_i^2 = t_i^1 + d + \theta \qquad (5)$$

$$t_i^3 = t_i^4 - d + \theta \qquad (6)$$

where θ represent the clock offset and d is the transmission delay. Then node A can calculate the clock offset and transmission delay as below, and synchronize itself to B.

$$\theta = \frac{(t_i^2 - t_i^1) - (t_i^4 - t_i^3)}{2} \qquad (7)$$

$$d = \frac{(t_i^2 - t_i^1) + (t_i^4 - t_i^3)}{2} \qquad (8)$$

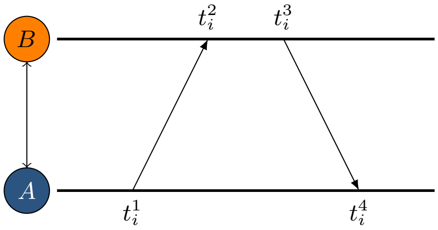

Fig. 1: **Round Trip Synchronization between two nodes**

Unidirectional Time Dissemination (UTD):

In this approach, a sender node disseminates a synchronization message including its local clock to its neighbor nodes while the others receive it and record its arrival times, as shown in *Figure* 2. Based on sender's and receiver's timestamp, receivers update their clock. According to *Equation*(2) the relationship between the time at the sender A and the time at the receiver B can be expressed as follows:

$$t_i^b = \alpha_{ab}.t_i^a + \beta_{ab} \qquad (9)$$

where β_{ab} and α_{ab} stand for relative clock offset and skew between node A and node B, respectively.

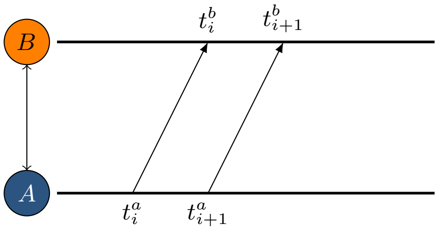

Fig. 2: **Unidirectional Time Dissemination**

Usually, in order to reduce the error introduced by the communication delay uncertainty, protocols that use the UTD technique timestamp the synchronization messages at the MAC layer.

Inter-Receivers Synchronization (IRS):

the IRS approach assumes that all neighboring nodes receive the same wireless signal at approximately the same time. Generally, there are two variations of IRS:

- *Receivers to Receivers Synchronization (RRS):* As depicted in *Figure* 3, node A acts as a *beacon* node and broadcasts a synchronization beacon message without any information about its local clock to its neighbors. The receivers B and C consider the arrival times t_i^b and t_i^c, respectively, as a reference point for comparing their clocks. Therefore, nodes B and C have to exchange their arrival times to each other. Finally, nodes A and B can easily compute the difference between their reception times to synchronize their local clocks.

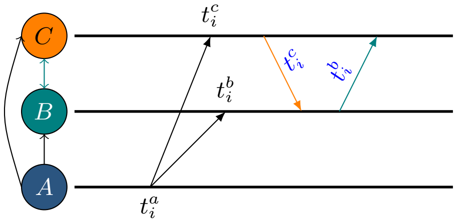

Fig. 3: Receivers to Receivers Synchronization

- *One-Receiver to Receivers Synchronization (ORRS):* Unlike the RRS technique, neighbor nodes in the ORRS technique can synchronize their local clocks by only receiving synchronization messages. To this end, *one receiver* should be elected to act as a reference point and the other *receivers* can be synchronized by comparing their local reading clocks with it. As shown in *Figure* 4, node A starts a synchronization round i by broadcasting a synchronization pulse to its neighbor nodes. Upon receipt, nodes B, C and D record the pulse arrival times using their local clocks. Next, node B, selected to be a reference point, broadcasts a packet containing its arrival time t_i^b. Once nodes C and D receive this time information they estimate the offset with respect to the reference point B by computing the difference between their reception times t_i^c and t_i^d and the received information t_i^b.

3.4 Clock behavior features

The main clock behavior features are described as follows:

Clock correction vs. clock transformation:

Typically, a synchronization process ends by estimating the relative offset and skew parameters with respect to either reference node or neighbor nodes. In order to make all clocks display the same time, nodes have to correct their clocks by compensating their local clock values using the estimated parameters. The most

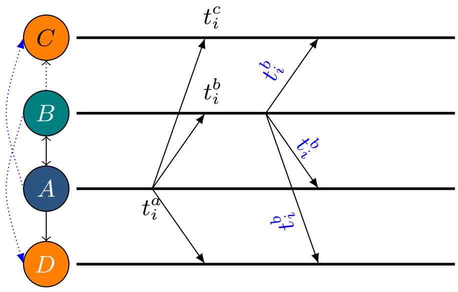

Fig. 4: **One-Receiver to Receivers Synchronization**

straightforward approach to achieve a common notion of time without clock correction is to perform time transformation. For instance, RBS [3] keeps locally both offset and skew of every other node with respect to its local clock. Once a time information is received, it translates the received clock value to its local time.

Offset vs. skew compensation:

In order to achieve synchronization, each node has to adjust (compensate) its local clock values using the estimated parameters. In fact, *offset* and *skew compensations* are used by nodes to adapt their local clock values to each other or to the reference(s) node(s). As shown in *Figure* 5, in order to adjust the local time, offset compensations arise at times t_1, and t_2. However, in *Figure* 6, the skew compensations arise at time t_1 to adjust the clock rate among nodes. Performing only the instantaneous clocks, offset compensation has the advantage of simplicity, but the clocks will deviate away due to the clock drift. In fact, to achieve more stable synchronizations without any frequent resynchronizations, clock skew compensation should be performed.

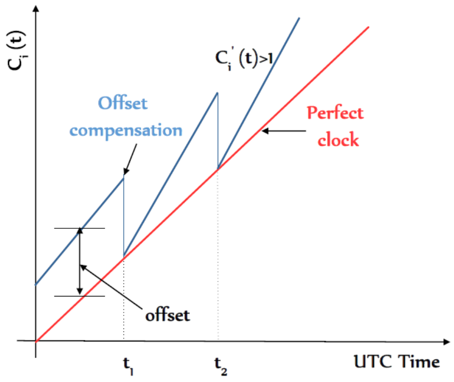

Fig. 5: **Offset compensations**

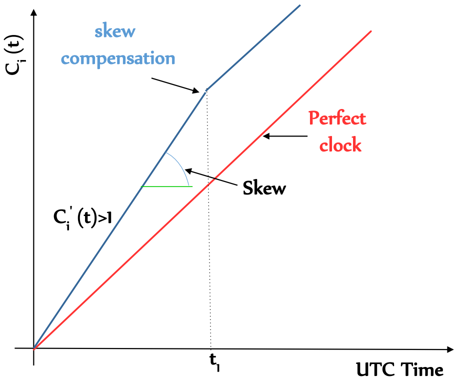

Fig. 6: Skew compensation

Static vs. dynamic skew compensation:

Usually, as soon as the clock synchronization process has been achieved, nodes should schedule the next synchronization round to compensate the potential clock drifting. Most of clock synchronization schemes force nodes to perform synchronization at a *static* interval to maintain their clocks synchronized. However, a short resynchronization interval could be expensive in terms of energy as the exchanged messages will be increased. To deal with that problem, a new class of synchronization schemes has been proposed recently (e.g, [9, 13]) that could considerably augment the resynchronization interval. In fact, the key idea is to perform the skew compensation *dynamically* according to the ambient condition changes.

4 Conclusion

Clock synchronization is a crucial issue for any distributed system and even more in WSNs, because many applications and communication protocols rely on it. Therefore, clock synchronization has been extensively studied in wired networks; unfortunately, the proposed solutions are often inappropriate for WSNs because they are typified by various constraints. Otherwise, achieving and maintaining clock synchronization at high accuracy with minimum energy cost is very challenging in such resource constrained networks. Indeed, this issue has driven researchers to design new techniques which fit to the restricted resources typifying WSNs. This paper has proposed a thorough classification including the previous ones and based on the features that should be taken into account when designing a new synchronization protocol in such networks. We believe that our work can give a valuable insight into most and recent work on clock synchronization techniques used by designers and also motivate and encourage new research.

Bibliography

[1] Ian F Akyildiz, Tommaso Melodia, and Kaushik R Chowdhury. A survey on wireless multimedia sensor networks. *Computer networks*, 51(4):921–960, 2007.

[2] Djamel Djenouri and Miloud Bagaa. Synchronization protocols and implementation issues in wireless sensor networks: A review. *Systems Journal, IEEE*, 2014.

[3] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, 2002.

[4] Jeremy Elson and Kay Römer. Wireless sensor networks: A new regime for time synchronization. *ACM SIGCOMM Computer Communication Review*, 33(1):149–154, 2003.

[5] Ya R Faizulkhakov. Time synchronization methods for wireless sensor networks: A survey. *Programming and Computer Software*, 33(4):214–226, 2007.

[6] Leslie Lamport and P Michael Melliar-Smith. Synchronizing clocks in the presence of faults. *Journal of the ACM*, 32(1):52–78, 1985.

[7] Jun Liu, Zhong Zhou, Zheng Peng, Jun-Hong Cui, Michael Zuba, and Lance Fiondella. Mobi-sync: efficient time synchronization for mobile underwater sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 24(2):406–416, 2013.

[8] Kay Römer, Philipp Blum, and Lennart Meier. Time synchronization and calibration in wireless sensor networks. *Handbook of sensor networks: Algorithms and architectures*, 49:199, 2005.

[9] Thomas Schmid, Zainul Charbiwala, Roy Shea, and Mani B Srivastava. Temperature compensated time synchronization. *Embedded Systems Letters, IEEE*, 1(2):37–41, 2009.

[10] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *Network, IEEE*, 18(4):45–50, July 2004.

[11] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 2005.

[12] Yik-Chung Wu, Qasim Chaudhari, and Erchin Serpedin. Clock synchronization of wireless sensor networks. *Signal Processing Magazine, IEEE*, 28(1):124–138, 2011.

[13] Zhe Yang, Lin Cai, Yu Liu, and Jianping Pan. Environment-aware clock skew estimation and synchronization for wireless sensor networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1017–1025. IEEE, 2012.