# Clock Synchronization in Wireless Sensor Network with Selective Convergence Rate for Event Driven Measurement Applications

Francesco Lamonaca, *Member, IEEE*, Andrea Gasparri, *Member, IEEE*, Emanuele Garone, *Member, IEEE*, and Domenico Grimaldi, *Senior Member, IEEE*

***Abstract*—**
**In this work a novel Wireless Sensor Network (WSN) synchronization protocol for event-driven measurement applications is proposed. The objective is twofold: i) to provide high accuracy in the area where an event is detected, ii) to ensure a long network lifetime. The complexity of the problem arises from the fact that these two properties are usually in conflict. In fact, to increase the synchronization accuracy, nodes are required to exchange synchronization packets at higher rate, thus impacting the network lifetime. *Vice versa* to ensure a long network life time, the number of packets to be exchanged should be minimized thus impacting the synchronization accuracy. A trade-off can be achieved by observing that the packet rate should be increased only for the portion of the network surrounding the detected events as only these nodes require a higher accuracy to collect data. The proposed algorithm represents a formalization of this idea. Numerical and experimental results are provided to validate the effectiveness of the proposed algorithm.**

***Index Terms*—Wireless Sensor Network, Clock Synchronization, Consensus Algorithm, Measurement Applications.**

## I. Introduction

Clock Synchronization is a crucial issue in a wide range of applications making use of Wireless Sensor Networks (WSNs) to perform event-driven measurements. These applications include: network localization [1–3]; medical care [4]; monitoring of civil infrastructures [5], debris flow [6], environmental monitoring [7–9]. Common requirements of such applications are long life time of the WSN and high synchronization accuracy among nodes.

In this work, we propose a novel clock synchronization algorithm for measurements applications. The proposed algorithm makes a trade-off between synchronization accuracy and network life-time preservation. In particular, by starting from the observation that a higher accuracy is required only around the area where an event is detected, a synchronization protocol which selectively increase or reduce the exchange packets rate according to the event-driven measurements requirements is provided. A very preliminary implementation of this idea has been proposed in [10]. The idea is to let the WSN modify its topological properties so that a quick convergence (and consequently a good accuracy despite of external disturbances) is ensured for all these nodes deployed around the area where the event is detected, while the rest of the network maintains a slower convergence rate and, thus, a lower synchronization accuracy. Therefore, the set of WSN nodes can be logically divided in two subsets: (i) the Improved Synchronized Subset (ISS) composed by nodes detecting the event and switching in alert state, and (ii) the Default Synchronized Subset (DSS) composed by nodes in quiet state, for which default low synchronization accuracy is imposed to reduce the energy consumption and extend their life time.

F. Lamonaca and D. Grimaldi are with the Dept. of Electronic, Systems and Computer Science, University of Calabria, Ponte Pietro Bucci, 87040 Rende (CS), Phone: +39 (0984) 494712, email: {flamonaca, grimaldi}@deis.unical.it

A. Gasparri is with the Dept. of Engineering, University of “Roma Tre”, Via della Vasca Navale, 79, Roma 00146, Phone: +39 (06) 5733-3206 , email: gasparri@dia.uniroma3.it

E. Garone is with Faculty of Applied Science, Control and Systems Analysis Department, Université Libre de Bruxelles, 1050 Brussels, Belgium. Phone: +32-(0)2-650-2686 , email: egarone@ulb.ac.be

## II. Related Work

Synchronization approaches for WSN can be classified in two main families: (i) hierarchical and (ii) fully distributed.

Algorithms belonging to the former organize the network in a tree where each node takes as reference its parent to compensate both the clock drift and skew [11,12]. The major limitation of this approach is that, any time the root or a parent node becomes unreachable, the related subtree looses synchronization until the network is re-organized. To overcome this problem, the network can be organized as a set of clusters. Nodes synchronize with one another in the same cluster. For global synchronization, in [13] is proposed to elect a node as local master within each cluster. The local masters are synchronized to achieve a common sense of time. In the case of a failure of a local master only the related cluster is temporarily unsynchronized [13].

Algorithms belonging to the latter [14–18] are robust to node failure as a master node does not exist. On the contrary, the common sense of time is achieved through local collaboration among the nodes.

More recently, approaches based on the consensus protocol are proposed [15–19]. They can be classified on the basis of: (i) the parameters object of the estimation and compensation, and (ii) communication modalities synchronous or asynchronous. In particular, in [15] the consensus approach is used to compensate the clock offsets in sparsely populated mobile ad hoc networks, while in [16] it is used to compensate the clock drift for Phase Locked Loops. In [17] a second-order consensus algorithm is proposed to compensate both clock offsets and clock drifts, but it requires a pseudo-synchronous communication among nodes. In [18], a protocol composed of the cascade of two consensus algorithms is proposed. There the first consensus synchronizes the clock frequency of the nodes and the second synchronizes their clock offsets. In [19] a novel synchronization algorithm to ensure a good level of synchronization even in the presence of random bounded communication delays is proposed.

In [10, 20] a preliminary algorithm has been introduced where consensus algorithms are used in an energetic efficient way so as to obtain accurate synchronization only where it is needed, while preserving the global convergence property.

Compared with existing synchronization schemes, the novel contributions of this paper are:

1. Improvement of the framework presented in [10], by introducing a new communication policy avoiding the accuracy of the ISS to be degraded due to the interaction with the DSS.
2. Improvement of the algorithm presented in [18] with a correction term to avoid overcompensation of the software time anytime the frequency correction is performed, while preserving the same convergence properties.
3. Extension of the single event scenario described in [20] to the cases of multiple events to be monitored. To this aim, a proof-of-concept algorithm to build a connected "accurate synchronization" area (ISS) is presented.

## III. Problem Statement

Consider a WSN composed of $N$ nodes with topology described by an undirected graph $\mathcal{G} = \{V, E\}$, with $V = \{1, \ldots, N\}$ the set of nodes representing the sensors and $E = \{(i, j)\}$ the set of edges describing the point-to-point channel availability, i.e., an edge $(i, j)$ exists if node $i$ can transmit to node $j$. Note that since the network topology is undirected the existence of an edge $(i, j)$ implies the existence of the edge $(j, i)$. Let the neighborhood $V_i$ of a node $i$ be the set $V_i = \{j : (i, j) \in E\}$, with $|V_i|$ its cardinality. Furthermore, denote with $t_k$ the instant when the $k$-th communication on the WSN happens and $\mathcal{G}(t_k) = \{V, E(t_k)\}$ the possibly directed graph that describes the communication at time $t_k$, i.e. $(i, j) \in E(t_k)$ if the node $i$ sends data to the node $j$ at time $t_k$. Clearly, $E(t_k) \subseteq E$ at each time step $t_k$. Denote with $\mathbb{G}(t_k, \Delta t) = \bigcup_{i=0}^{\Delta t} G(t_{k+i})$ the graph obtained by the union of all the links activation in the time window $[t_k, t_{k+\Delta t}]$. Finally denote as rooted graph a graph in which there exists at least one node for which a path with any other node can be established.

Each node $i$ is equipped with a (local) *hardware* clock $\tau_i$ defined as:

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

where $\alpha_i$ is the local clock frequency and $\beta_i$ is the offset. Notably, the coefficients $(\alpha_i, \beta_i)$ differ for each node due to actual hardware components. Therefore, a synchronization algorithm must be provided to keep a common notion of time, otherwise clocks might diverge with respect to the others.

To this end, 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 by which a common sense of time can be achieved, i.e. synchronization of the software clocks.

***Problem 1:*** Consider an undirected connected graph $\mathcal{G} = \{V, E\}$ describing the WSN topology. Design a distributed algorithm based only on local interactions among nodes such that all the *software* clocks converge to a common sense of time, that is:

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

Notably the majority of clock synchronization approaches proposed for WSNs ( [14, 16–18]) are based on the idea that all nodes over the network exchange regularly information with their neighbors at a given frequency. However, as shown in Section IV-D, the synchronization accuracy closely depends on the rate at which packets are exchanged.

Based on this observation, the following more general problem is addressed in this work:

***Problem 2:*** Consider an undirected graph $\mathcal{G} = \{V, E\}$ describing the network topology of a sensor network. Design a synchronization protocol for measurement applications which solves Problem 1 but where the synchronization accuracy can be dynamically changed according to the measurements requirements.

In the sequel, a framework to address this problem is proposed where two different synchronization accuracies might coexist over the network, according to the application requirements. Roughly speaking, the following components are necessary to build such a framework:

- **The Synchronization Protocol** - to to keep a common notion of time among node clocks.
- **Multi-Rate Policy** - to ensure the correct behavior of the network with ISS and DSS components.
- **ISS Connector algorithm** - to build a unique ISS in response to triggering events.

These aspects will be detailed in the next sections.

## IV. The Revised ATS Algorithm

The ATS algorithm introduced in [18] is here improved by the introduction of a different offset compensation policy to attenuate the overcompensation of $\hat{\tau}_i(t)$ during clock frequency compensations. The convergence of the proposed strategy will be investigated for rooted communication topologies. In the sequel, $t_k$ will denote the time when a message is exchanged over the network and $t_k^+$ will denote the time when the parameters update is carried out.

### A. The ATS Algorithm

The ATS synchronization algorithm is based on the idea that each node determines its own $\hat{\alpha}_i(t)$ and $\hat{o}_i(t)$ through local interactions with its neighbors. In particular, let $t_k$ be the time when the node $j$ sends a packet into the network.

The packet contains the tuple $(\mathrm{id}_j, \hat{\alpha}_j, \hat{o}_j, \tau_j)$ where $\mathrm{id}_j$ is an identifier of the $j$-th node. $\hat{\alpha}_j = \hat{\alpha}_j(t_k)$ and $\hat{o}_j = \hat{o}_j(t_k)$ are the local clock corrections at time $t_k$, and $\tau_j$ is the hardware timestamp of the packet, i.e. the value of the hardware clock in the moment the message is sent $\tau_j = \tau_j(t_k)$. When the node $i$ receives the packet, it first stores the current value of its "hardware" local clock in a variable $\tau_{ij}$. Assuming the transmission and time-stamping operations instantaneous, then $\tau_{ij} = \tau_i(t_k)$.

At this point, the node $i$ executes the local synchronization procedure consisting of three steps:

*1* **Relative drift estimation:** The $i$-th node computes the relative drift estimation $\alpha_{ij}(t_k^+)$ comparing $\tau_i$ and $\tau_j$ in two different instants and evaluating the relative frequencies. To this end, each node $i$ must store two variables $\tau_j^{old}$, $\tau_{ij}^{old}$ for each neighbor $j$ in an internal structure. It follows:

$$\alpha_{ij}(t_k^+) = \frac{\alpha_j}{\alpha_i} = \frac{\tau_j - \tau_j^{old}}{\tau_{ij} - \tau_{ij}^{old}} \tag{4}$$

where $\alpha_i$, $\alpha_j$ are the real clock frequencies for the nodes $i$ and $j$, respectively. $\tau_j$ and $\tau_{ij}$ are stored into $\tau_j^{old}$ and $\tau_{ij}^{old}$ after the update.

*2* **Drift compensation**: $\hat{\alpha}_i$ is updated on the basis of $\alpha_{ij}(t_k^+)$ as follows:

$$\hat{\alpha}_i(t_k^+) = \rho_v \hat{\alpha}_i(t_k) + (1-\rho_v)\alpha_{ij}(t_k^+)\hat{\alpha}_j(t_k), \tag{5}$$

*3* **Offset compensation:** $\hat{o}_i$ is updated as follows:

$$\hat{o}_i(t_k^+) = \hat{o}_i(t_k) + (1-\rho_o)(\hat{\tau}_j(t_k) - \hat{\tau}_i(t_k)) \tag{6}$$

where $\rho_v$ and $\rho_o$ are design parameters that can be set between 0 and 1. The correction parameters are usually initialized at $\hat{\alpha}_i = 1$, $\hat{o}_i = 0$, with $i = 1, ..., N$. Note that (4) and (5) can be performed only if at least one message from node $j$ has been previously received.

The following results was proven in [18].

***Theorem 1:*** [18] Consider a WSN synchronized through the ATS Algorithm. Under the assumption $\alpha_i$ and $o_i$ are constants values $\forall i \in V$ and there are no transmission delays, if it exists an integer $\Delta t > 0$ such that for any integer $k > 0$ the graph $\mathbb{G}(t_k, \Delta t)$ is strongly connected, then all software clocks will synchronize exponentially fast:

$$\lim_{t\to\infty} \hat{\tau}_i(t) = \hat{\tau}_j(t), \forall i, j \in V. \tag{7}$$

To reduce the sensitivity of the relative frequencies estimate, in [15] the authors suggest to modify the update of $a_{ij}$ in (4) by introducing a low pass filter:

$$\alpha_{ij}(t_k^+) = (1-\rho_l)\alpha_{ij}(t_k) + \rho_l \frac{\tau_j - \tau_j^{old}}{\tau_{ij} - \tau_{ij}^{old}} \tag{8}$$

where $\rho_l$ is a design parameter between 0 and 1. Low values of $\rho_l$ are advised for high communication frequencies.

As pointed out in [18], communication among nodes in the ATS algorithm can be asynchronous, i.e., the update of $\hat{\alpha}_i(t_k^+)$ and $\hat{o}_i(t_k^+)$ can be carried out upon the reception of a new message, thus providing robustness to packets' loss. Notably, as it will become clear in the sequel, all the properties of ATS algorithm are inherited by its enhanced version proposed in our work.

### *B. The Revised ATS and its properties*

In this paper, the ATS algorithm is modified by replacing the offset compensation term (6) with the following one

$$\hat{o}_i(t_k^+) = \hat{o}_i(t_k) + (1-\rho_o)(\hat{\tau}_j(t_k) - \hat{\tau}_i(t_k)) - \Delta\hat{\alpha}_i(t_k)\tau_i(t_k), \tag{9}$$

where $\Delta\hat{\alpha}_i(t_k) = (\hat{\alpha}_i(t_k^+) - \hat{\alpha}_i(t_k))$. The importance of this improvement is that the additional correction term prevents the software clock time $\hat{\tau}_i(t)$ being overcompensated due to changes of the $\alpha_i$ values.

To understand the importance of this correction term, consider the case where a packet was sent to node $i$ by node $j$ at time $t_k$, and assume that a frequency variation is detected. Then, $\Delta\hat{\alpha}_i(t_k) \neq 0$ while, at the same time, the software time are synchronized $\hat{\tau}_i(t_k) = \hat{\tau}_j(t_k) = \tau'$. This means that, if (6) is used, $\Delta\hat{o}_i(t_k) = \hat{o}_i(t_k^+) - \hat{o}_i(t_k) = 0$. The software clock time, according to (2), is

$$\begin{aligned} \hat{\tau}_i(t_k^+) &= \hat{\alpha}_i(t_k^+)\,\tau_i(t_k) + \hat{o}_i(t_k^+) \\ &= (\hat{\alpha}_i(t_k) + \Delta\hat{\alpha}_i(t_k))\,\tau_i(t_k) + \hat{o}_i(t_k) \\ &= \tau' + \Delta\hat{\alpha}_i(t_k)\tau_i(t_k) \end{aligned} \tag{10}$$

Note that the term $\Delta\hat{\alpha}_i(t_k)\tau_i(t_k)$ introduce an overcompensation that is proportional to the value of the hardware clock $\tau_i(t_k)$ and that, as the time goes on, can be arbitrarily large in the transient. This may affect significantly the synchronization accuracy.

The proposed correction term counterbalances this effect. The following convergence property can be proven.

***Theorem 2:*** Consider a WSN synchronized through the Revisited ATS Algorithm. Under the assumption $\alpha_i$ and $o_i$ are constants values $\forall i \in V$ and there are no transmission delays, if there exists an integer $\Delta t > 0$ such that for any integer $k > 0$ the graph $\mathbb{G}(t_k, \Delta t)$ is rooted, then all the software clock will synchronize exponentially fast:

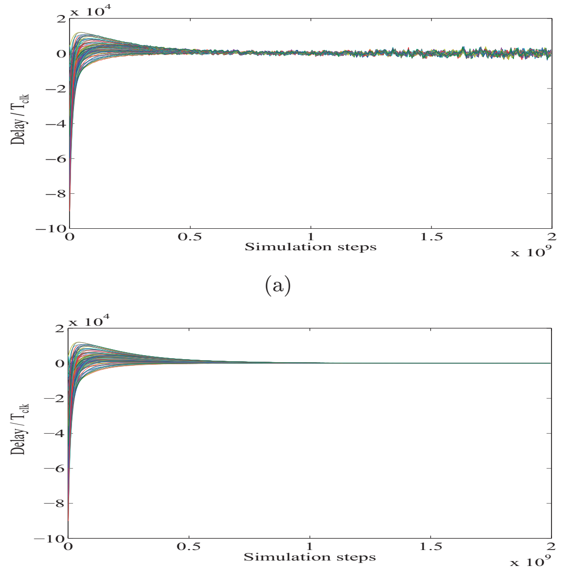
$$\lim_{t\to\infty} \hat{\tau}_i(t) = \hat{\tau}_j(t), \forall i, j \in V. \tag{11}$$

*Proof:* The proof follows the same line of Theorem 6 in [18]. The only differences are that :

- in [18] the case of strongly connected graph is considered. This can be extended to rooted graph by noticing that Theorem 6 in [18] holds true also in the case an integer $K$ such that $Q_l = P_{(l+1)K-1} \ldots P_{lK+1}P_{lK}$ has at least one column whose elements are all positive for all $l = 0, 1, \ldots$
- The presence of the term $\alpha_i(t)\tau_i - \alpha_i(t^+)\tau_i(t)$, in the $\hat{\tau}_i$ convergence proof, which being an exponentially vanishing term can be treated using the same arguments of Theorem 6 in [18].

■

Note that, the algorithm is robust against node failure as according to Theorem 2 only a routed communication graph over a certain time period is required for the convergence. Convergence might be lost only if a node failure determines a split of the network into several components. Nevertheless, it should be noticed that even in this special case Theorem 2 ensures the convergence within each residual connected component.




(a)

(b)

Fig. 1. Trend of delays of 100 nodes respect to $node\#1$ in the case the correction term $-\Delta\hat{\alpha}_i(t_k)\tau_i(t_k)$ is not used a), and is used b).

As it will be shown later on, the extension of Theorem 1 to the case of rooted graphs is a key point for the development of a synchronization framework with selective convergence rate.

Figure 1 shows the difference between the software clock of each node and node#1 taken as a reference for a WSN with $N = 100$ nodes deployed as a $10 \times 10$ lattice. Both the difference and the simulation steps are normalized respect to the mean clock period $T_{clk} = \frac{N}{\sum_{i=1}^{N} \alpha_i}$. In particular, Fig.1 a) shows the performance of the standard ATS, while Fig.1 b) shows the performance with the proposed correction term. It can be noticed that a smoother trend is experienced in the latter case.

### C. Implementation and Technological Aspects

The ATS algorithm and the proposed revised version can be implemented in a completely event-driven fashion, according to an asynchronous communication scheme. Nevertheless, as pointed out in [21], this does not represent a good engineering practice for energetic reasons. In fact, to

keep the radio in the listening mode to wait for arriving packets is a very expensive operation from the energetic standpoint. Therefore, the following transmission policy is adopted in this work: each node $i$ sends packets when the software clock is such that it exists an integer $m$ satisfying $\hat{\tau}_i(t) = \bar{t}_i + mT_i$. In other words, the node $i$ will send a packet periodically on the basis of its software clock with a synchronization period $T_i$ and with an offset $\bar{t}_i \in [0, T_i)$. In order to mitigate the occurrence of packet collisions, no node of the same neighborhood should have the same offset $\bar{t}_i \neq \bar{t}_j \neq \bar{t}_k, \forall i \in V, \forall j,k \in V_i$. If possible, all offsets in the selected neighborhood should be equally spaced over the communication period.

Another important aspect to be considered is the transmission delay, currently not taken into account in the mathematical modeling. As a matter of fact, this delay can be made negligible with respect to the clock period by taking the time-stamping at the MAC layer. If this cannot be done, techniques to compensate transmission delays are available (see [22]).

### D. Packets Rate and Synchronization Accuracy

Goal of this subsection is to provide some insight on the close relationship between packets rate and synchronization accuracy. To this end let us consider, for the sake of simplicity, the simplified case where only offset adjustments are performed ($\rho_l = 0$) and all the nodes (fictitiously) communicate in a synchronous way with a period $m\,T$, where $m$ is a positive integer. The clock evolution every period $T$ is assumed to be given by a nominal value plus an unknown time-varying term:

$$\tau_i((k+1)T) = \tau_i((k)T) + \bar{\alpha}\,T + d_i(k) \tag{12}$$

where $\bar{\alpha}$ is the clock nominal value and $d_i(k)$ is a time-varying bounded disturbance such that $||d_i(k)|| < d_{max}$. It follows that, if we consider the clock evolution every $m\,T$ steps (without adjustments), we have:

$$\tau_i\big(t_{k+1}\big) = \tau_i\big(t_k\big) + m\,\bar{\alpha}T + d_{m,i}(k) \tag{13}$$

where $t_k = (k)\,m\,T$ and $|d_{m,i}(k)| < m\,d_{max}$. Assuming the offset adjustments (9) are performed, it follows that:

$$\begin{aligned}\hat{\tau}_i(t_k^+) &= \tau_i(t_k) + \hat{o}_i(t_k) + (1-\rho_o)(\hat{\tau}_j(t_k) - \hat{\tau}_i(t_k)) \\ &= \hat{\tau}_i(t_k) + (1-\rho_o)(\hat{\tau}_j(t_k) - \hat{\tau}_i(t_k))\end{aligned} \tag{14}$$

where $\Delta\alpha(t_k) = 0$ follows from the assumptions. Since a synchronous communication model is considered, the following overall update model is obtained:

$$\hat{\tau}(t_k^+) = \big(I + \varepsilon\,L\big)\hat{\tau}(t_k) \tag{15}$$

where $\tau = [\tau_1, ..., \tau_n]^T$, $L$ is the usual Laplacian matrix describing the connections of the communication graph and $\varepsilon = 1 - \rho_o$. Note that, the equivalence between the synchronous model and asynchronous model in terms of convergence set holds under the additional assumption that $\varepsilon \in [0, 1/N]$. At this point, using (13) we can compute $\hat{\tau}(t_{k+1})$ as:

$$\hat{\tau}(t_{k+1}) = \hat{\tau}(t_k^+) + 1_{N\times 1}\,m\,\bar{\alpha}T + d_m(t_k) \tag{16}$$

where $d_m(k) = [d_{m,1}(k), ..., d_m(k)]^T$ and it is bounded as $\|d_m(k)\|_\infty < m\,d_{max}$. Using (15) the above equation becomes:

$$\hat{\tau}(t_{k+1}) = \big(I + \varepsilon\,L\big)\hat{\tau}(t_k) + 1_{N\times 1}\,m\,\bar{\alpha}T + d_m(t_k) \tag{17}$$

It is important to remind that, as well know, if the graph is connected the matrix $I + \varepsilon L$ has only one eigenvalue in 1, $\lambda_1 = 1$, while all the others are strictly inside the unit ball ($|\lambda_i| < 1, i = 2, ..., n$). Moreover, the eigenvector associated to $\lambda_1$ is the unit vector $v_1 = 1_{N\times 1}$. As we are interested in studying the synchronization error, define now the generic synchronization error between node

$i$ and $j$ as $y_{ij}(k) = C_{ij}\tau(t_k)$ where $C_{ij} = (e_i - e_j)^T$ and $e_i, e_j$ are the $i$-th and $j$-th vectors of the canonical basis. Interestingly enough, since $C_{ij}\, 1_{N\times 1} = 0$ the term $1_{N\times 1}\, m$ does not contribute to the synchronization error. Moreover for the same reason, being $1_{N\times 1}$ the eigenvector $v_1$, it also means that the eigenvalues $\lambda_1$ has no influence on the synchronization error $e_{ij}$ (in other words $\lambda_1$ is not observable). Then, since all the observable eigenvalues are strictly contained in the unit ball, for negligible initial conditions, it is possible to bound the maximum magnitude of the synchronization error with respect to the maximum magnitude of the disturbance as follows:

$$||y_{ij}(k)||_\infty < l_1 \;\; ||d(k)||_\infty = l_1\, m\, d_{max}, \forall k \qquad (18)$$

where $l_1 = \sum_{r=1}^{N}\sum_{k=0}^{\infty} |C_{i,j}(I + \varepsilon L)^k e_r|$ is the 1-norm of the impulsive response matrix between the disturbance vector and the output $y_{ij}$ [23].

This latter relationship gives an insight on the fact that the synchronization error is linked in a proportional way with the length of the time interval between two consecutive communications which depends upon $m$, and then that the lower the synchronization period, the better the synchronization accuracy. As a last observation, it should be mentioned that the parameter $l_1$ is influenced by the second largest eigenvalue of the matrix which depends upon the connectivity of the graph. In other words, the more the graph is connected, the lowest the influence of the disturbance on the synchronization error.

## V. Multi-Rate Policy

In this section we introduce a multi-rate communication policy to build up the ISS with higher accuracy while the rest of the network remains synchronized with the default one.

Let $V$ be the set of the nodes of a WSN and let $V_{ISS} \subseteq V$ be the ISS, i.e. the subset of nodes that are required to be synchronized with higher accuracy. Recall that the remaining nodes $V_{DSS} = V \setminus V_{ISS}$ belongs to the DSS, i.e., the subset of nodes synchronized with the default accuracy.

To reach a higher accuracy, the nodes belonging to the ISS are required to exchange packets more frequently compared to the remaining nodes. Therefore, a communication policy to regulate the exchange of messages between the ISS and the DSS is required to avoid the notion of time of the two components to drift away. Simultaneously, we must avoid the accuracy of the ISS to be degraded due to the interaction with the DSS. To this aim, we force the communication only from the ISS to the DSS. Indeed, if a node in alert state took into account the synchronization messages of the nodes in quite state, it would tune its clock on the basis of clocks coarsely synchronized. In this way, the ISS component is ensured to quickly converge towards a higher synchronization accuracy, while the rest of the network will remain synchronized with the default one.

**Communication Policy:**

- Each node $i \in V_{ISS}$ communicates every $T_i = T_{ISS}$, while each node $i \in V_{DSS}$ communicates at $T_i = T_{DSS}$, with $T_{DSS}$ multiple integer of $T_{ISS}$;
- Each node $i \in V_{ISS}$ discards synchronization packets received from nodes $j \in V_{DSS}$;

Note that, according to the communication policy, $V_{ISS}$ is a strongly connected subgraph, i.e. for any $i, j \in V_{ISS}$, it exists a path from $i$ to $j$ in $V_{ISS}$. Moreover, as proved by the following lemma, the overall network topology becomes rooted, which is the condition to satisfy the conditions of Theorem 2.

***Lemma** 1:* Let $\mathcal{G} = \{V, E\}$ be a connected undirected graph and $V_{ISS}$ be a connected subgraph. The graph $\mathcal{G} = \{V, \tilde{\mathcal{E}}\}$ with $\tilde{E} = E \setminus \{(i,j) \in E | i \in V_{DSS}, j \in V_{ISS}\}$ is rooted, i.e. it exists a node $i \in V$ such that for any $j \in V$ there exists a path connecting $i$ to $j$.

*Proof:* Consider a node $i \in V_{ISS}$. Being $V_{ISS}$ a connected subgraph, for any $j \in V_{ISS}$ it exists a path internal to $V_{ISS}$ from $i$ to $j$. Let us denote with $Seq_{V_{ISS}}(i,j)$ such a path. Being $\mathcal{G}$ connected, if we select a node in $i \in V_{ISS}$ then, for any $j \in V$ it exists a path in $\mathcal{G}$ from $i$ to $j$ composed of a certain sequence of arcs. The proof is concluded by noticing that if a sequence $(i_1 = i, i_2, i_3, ...., i_{z-1}, i_z = j)$ connecting $i$ to $j$ in $\mathcal{G}$ contains an arc $(i_{k-1}, i_k)$ such that $i_{k-1} \in V_{DSS}$ and $i_k \in V_{ISS}$, then also $(Seq_{V_{ISS}}(i_1 = i, i_k, i_{k+1}, ...., i_{z-1}, i_z = j))$ is a path connecting $i$ to $j$. ■

Note that, Lemma 1 assumes $V_{ISS}$, that is the ISS, to be a connected subgraph. In the context of WSNs this might be not a trivial task to achieve especially if the network is expected to react and self-organize in response to an external event triggering a certain number of nodes in alert state. For this reason the ISS Connector algorithm is described in Section VI.

## VI. ISSs Connector (IC) Algorithm

This section describes a strategy ensuring that the nodes possibly alerted by multiple events always form a unique ISS component. Figure 2 describes the overall idea of the algorithm. In particular, the two possible states (alert and quiet) of the nodes and their related activities are depicted.

Each node detecting an event puts itself in alert mode. Furthermore, it broadcast this information over the network by sending a *detection* packet.

If a node in quiet mode receives a *detection* packet, it stores the ID of the node that sent it, and add its ID to the packet before forwarding it to its neighborhood. To avoid multi-paths and reduce the number of exchanged messages, further detection packets sent by the same source (ID) will be discarded for a certain amount of time $\Delta t$.

If a node in alert mode receives a *detection* packet sent by another node in alert mode, it stores the ID of such an alert node, then it adds its ID to the packet and forwards it to its neighbors. In addition, it sends back a *reception* packet to the alert node that sent this *detection* packet simply by reversing the sequence of IDs. Note that, although the communication is assumed to be broadcast, by providing this information, the number of messages exchanged is significantly reduced. As before, to avoid multi-paths and reduce the number of exchanged messages, further detec-

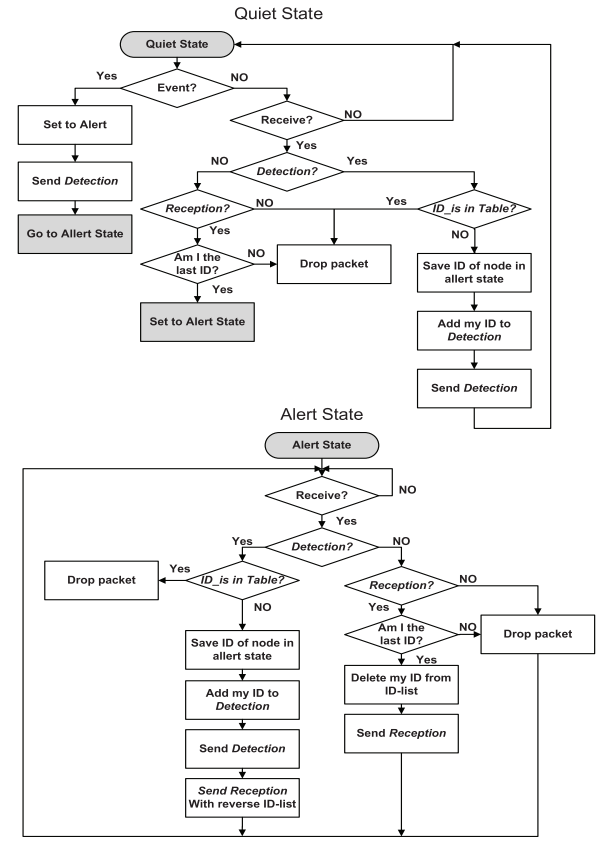



Fig. 2. Flowchart of ISSs Connector Algorithm.

tion packets sent by the same source (ID) will be discarded for a certain amount of time.

If a node in quiet mode receives a *reception* packet, it checks whether it is the last element of the sequence of IDs. If this is the case, it marks itself as an *alert* node. Furthermore, it takes itself away from the sequence and forwards the *reception* packet to its neighborhood. Otherwise, It drops the packet.

Finally, if a node in alert mode receives a *reception* packet, it checks whether it is the last element of the sequence of IDs. If this is the case, it takes itself away from the sequence and forwards the packet, it drops the packet otherwise.

A few remarks are now in order:

- A ISS is composed not only of the nodes that detect an event, but also by additional nodes which serve as a communication mean to ensure its connectedness.
- A DSS is automatically obtained upon a ISS construction, as in practice it is constituted by the remaining portion of the network.
- The proposed approach is a simple heuristic. Therefore, the obtained connected components might not be optimal as additional nodes not strictly required to the connectedness of the ISS might be added.

Note that, the proposed algorithm although very simple does not introduce any communication overhead as the information required to build the ISS can be added to the packets for the loose synchronization which are exchanged at the default low rate.

## VII. Reduction of the Energy Consumption

In order to evaluate the Reduction of the Energy Consumption (REC) factor, the ratio between the energy consumed by using the proposed synchronization method and the one consumed in the case all nodes synchronize with the same synchronization time interval $T_{sync}$ is analyzed. Because each node uses the radio section in the time interval $[t_k - \delta, t_k + \delta]$, the energy in this time interval $E_{2\delta}$ can be considered constant. If all the WSN nodes are equipped with same hardware component, $E_{2\delta}$ can be assumed equal for all nodes. Therefore, in the observation time interval $T_{OS}$, the energy consumed by the node $E_{T_{OS}}$ is:

$$E_{T_{OS}} = E_{2\delta} * \frac{T_{OS}}{T_{synch}}. \tag{19}$$

The energy consumed by using the proposed synchronization algorithm $E_{TOS}^{prop}$ is equal to the sum of the energy consumed by nodes in $V_{ISS}$ and $V_{DSS}$:

$$E_{T_{OS}}^{prop} = E_{2\delta} * \left( |V_{ISS}| * \frac{T_{OS}}{T_{ISS}} + |V_{DSS}| * \frac{T_{OS}}{T_{DSS}} \right), \tag{20}$$

where $|\cdot|$ denotes the cardinality of the argument. The energy consumed by using the traditional synchronization algorithm, i.e. same synchronization period $T_{sync}$ in all nodes $E_{T_{OS}}^{trad}$ is:

$$E_{T_{OS}}^{trad} = E_{2\delta} * N * \frac{T_{OS}}{T_{sync}}. \tag{21}$$

Because the evaluation of $E_{TOS}^{prop}$ and $E_{TOS}^{trad}$ must be performed on the same WSN by guaranteeing same synchronization accuracy, the following conditions hold: $N = |V_{ISS}| + |V_{DSS}|$ and $T_{sync} = T_{ISS}$. Therefore, it is:

$$\begin{aligned} REC &= 1 - \frac{E_{2\delta} * \left( |V_{ISS}| * \frac{T_{OS}}{T_{ISS}} + |V_{DSS}| * \frac{T_{OS}}{T_{DSS}} \right)}{E_{2\delta} * \left( N * \frac{T_{OS}}{T_{ISS}} \right)} \\ &= 1 - \frac{k * |V_{ISS}| + |V_{DSS}|}{k * (|V_{ISS}| + |V_{DSS}|)}, \end{aligned} \tag{22}$$

where $T_{DSS} = kT_{ISS}$ with $k \in \mathbb{N}^+$.

On the basis of the relationship between the packet rate and the synchronization accuracy given in Subsection IV-D it is clear that the optimal ratio between $T_{ISS}$ and $T_{DSS}$ depends on the particular phenomenon under monitoring. $T_{ISS}$ is selected on the basis of the required synchronization accuracy. $T_{DSS}$ is selected on the basis of the allowed maximum delay among nodes and the desired REC factor.

## VIII. Numerical Tests

For the numerical evaluation, we used the clock parameters $\alpha_i$ and $\beta_i$ $i = 1, ..., N$ of the hardware clock of the wireless sensor TelosB used for the experimental validation. In particular, the frequency of the crystal oscillators is $f_{clk} = 32.768\text{kHz} \pm 20\text{ppm}$ [24]. Therefore, $\alpha_i$ varies in the range $[0.999980, 1.000020]f_{clk}$. By taking into consideration that in the real functioning, each node is switched on asynchronously, and the synchronization algorithm is started at the switch on, the offset is assumed in the range $[30, 3000]ms$. As a consequence, $\beta_i$ varies in the range $[10^3, 10^5]T_{clk}$, $T_{clk} = 1/f_{clk}$.

Moreover, we assume the nodes in quiet state to synchronize with $T_{DSS} = 3 \times 10^7\ T_{clk}$; while the ones in alert state with $T_{ISS} = 3 \times 10^6\ T_{clk}$. The $T_{OS}$ is $2 \times 10^9\ T_{clk}$. The simulation steps is set equal to $T_{clk}$.

The effect of the noise on the clock period is taken into account, as well. We consider a normal distribution modeling with standard deviation equal to that characterizing the clock period of the TelosB. Fig. 3 shows the probability density function (pdf) experimentally evaluated on 3000 clock period values. The $\chi^2 - test$ verifies the null

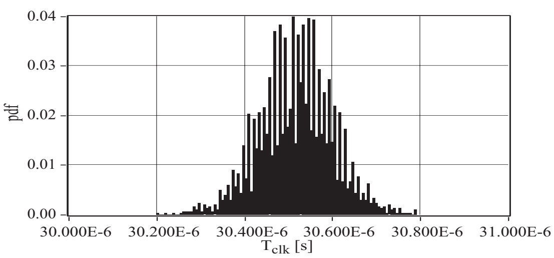



Fig. 3. Experimental pdf of the clock period of the TelosB sensor.

hypotheses of normal distribution. The standard deviation is equal to $84ns$, i.e. $0.0028\ T_{clk}$. In order to test both the convergence properties of the algorithm in the case of two different synchronization time intervals of the nodes and the IC algorithm, we consider a WSN with lattice topology 5x4. The reduced number of nodes makes easy to monitor the trend of the delay of each node during the IC algorithm execution. The sub-sets $V_{ISS}(1)$ and $V_{ISS}(2)$ with lattice topology 2x2 are assumed. The two $V_{ISS}$ are positioned at the opposite corners of the lattice and, then, they are not connected. Fig. 4 shows the topology of the WSN at the end of the IC algorithm execution. In particular, the black dots under the gray rectangles represent the nodes in alert state, which detect the event; the black dots represent the nodes in quiet state that the IC algorithm forces in alert state, the other dots represent the nodes in quiet state.

In the case of Fig. 4 it is $|V_{ISS}| = 13$, $|V_{DSS}| = 7$. The synchronization is performed by imposing $T_{ISS} = 3 \times 10^7 T_{clk}$, $T_{DSS} = 3 \times 10^6 T_{clk}$. The REC factor, obtained by (22) is 31.5%.

Fig. 5 shows the trend of the delay of the nodes of $V_{ISS}(1)$ (black dots), $V_{ISS}(2)$ (dark gray dots), $V_{DSS}$ (gray) respect to node#1 of $V_{ISS}(1)$. In the case the IC

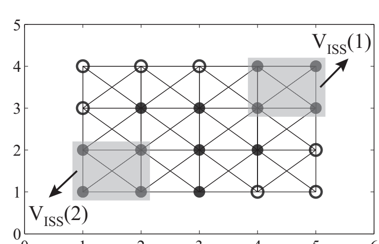
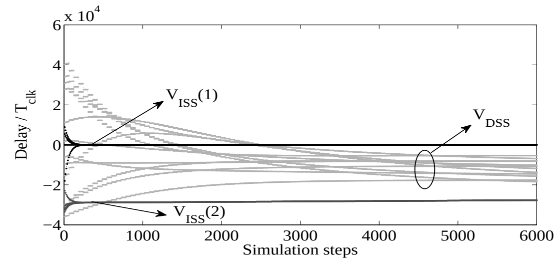
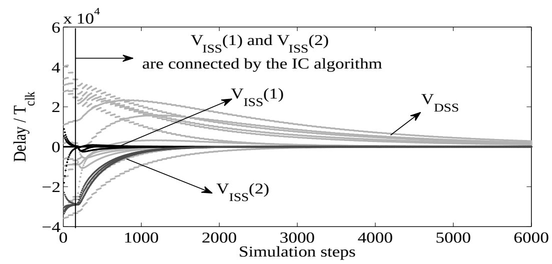



Fig. 4. Topology of the WSN after the execution of the IC Algorithm.

algorithm is not used, Fig. 5 a) highlights the convergence of the nodes of $V_{ISS}(1)$, and of $V_{ISS}(2)$ to two different sense of time. The nodes of $V_{DSS}$ have time clock values depending on the sense of times of both the two $V_{ISS}$. Fig. 5 b), highlights the convergence of the nodes of $V_{ISS}(1)$, and $V_{ISS}(2)$ to two different sense of time before the IC algorithm creates the connection among them. Once connected, all the nodes of WSN converge to mean common sense of time.




(a)

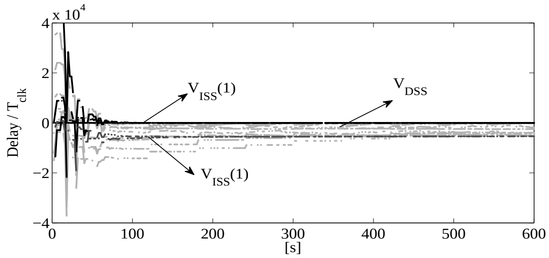



(b)

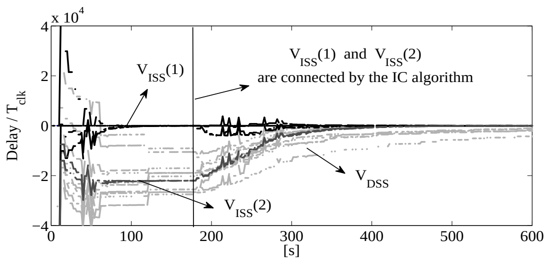
Fig. 5. Trend of the delay of the nodes of $V_{ISS}(1)$ (black dots), $V_{ISS}(2)$ (dark gray dots), $V_{DSS}$ (gray) respect to node#1 of $V_{ISS}(1)$ in the case the IC algorithm is a) not used, and b) used.

## IX. Experimental Results

Experiments have been carried out to evaluate the proposed framework with a real WSN testbed. To have comparable results, the same topology, $T_{DSS}$ and $T_{ISS}$ of the numerical tests are considered. The 20 nodes are deployed in less than one square meter. The lattice topology is obtained by forcing each node to ignore messages sent by nodes which are not considered as neighbors. The high density of node per square meter allows to evaluate the effectiveness of the proposal in a saturated spectrum network. Each node is constituted by TelosB wireless sensor [24]. The fundamental characteristics justifying the selection of this sensor is the MAC-layer time-stamping, allowed by the radio chip CC2420, that reduces potential unpredictable delays between the readings and the transmitting of the synchronization messages. This allows to assume that communication delays can be neglected. To evaluate the delay among all nodes the following procedure is executed. The auxiliary node, not involved in the synchronization procedure, sends the trigger message (TrMsg) with time period $T_s$ in broadcast mode. Each node sends to the PC its software clock value $\hat{\tau}_i(t)$ once received TrMsg. $T_s$ is set lower than $T_{ISS}$ in order to evaluate the trend of the delay during two successive synchronization phases, and bigger enough to reduce the probability of packet col-




(a)




(b)

Fig. 6. Trend of the delay of the nodes of $V_{ISS}(1)$ (black dots), $V_{ISS}(2)$ (dark gray dots), $V_{DSS}$ (gray) respect to the first received time value sent by the nodes of $V_{ISS}(1)$ in the case the IC algorithm is a) not used and b) used.

lision among TrMsg and synchronization messages.

The PC saves $\hat{\tau}_i(t)$, $i = 1...20$ in the report file and, computes the delay of each node respect to the software time $\hat{\tau}_*(t)$ of the first received time value sent by the nodes of $V_{ISS}$: $d_i(t) = \hat{\tau}_i(t) - \hat{\tau}_*(t)$, $i = 1...20$. Differently from the numerical tests, the selection of node#1 as reference is not possible in the experimental tests because the corresponding software clock $\hat{\tau}_1(t)$ may be not received as consequence of the packet loss. Fig. 6 shows the trend of the delay of the nodes of $V_{ISS}(1)$ (black dots), $V_{ISS}(2)$ (dark gray dots), $V_{DSS}$ (gray) respect to the first received time value sent by the nodes of $V_{ISS}(1)$. In the case the IC algorithm is not used, Fig. 6 a) shows the convergence of the nodes of $V_{ISS}(1)$, and of $V_{ISS}(2)$ to two different sense of time. The nodes of $V_{DSS}$ have time clock values depending on both the two $V_{ISS}$. Fig. 6 b), shows the convergence of the nodes of $V_{ISS}(1)$, and $V_{ISS}(2)$ to two different sense of time until the IC algorithm creates the connection among them. The topology of the WSN after the IC algorithm execution is the same of that shown in Fig. 4. Once connected, all the nodes of WSN have a mean common sense of time.

By comparing the numerical ones (Fig. 5) and the experimental results (Fig. 6) it can be noticed that the trend of the delay is the same in spite of the packet loss that can occurs in the real WSN. Particularly, the maximum delay of the nodes respect to node#1 is:

- In the case of numerical test: equal to $16T_{clk}$ for ISS nodes, $55T_{clk}$ for DSS nodes.
- In the case of experimental test: equal to $17T_{clk}$ for ISS nodes, $59T_{clk}$ for DSS nodes.

Furthermore, by comparing the experimental results and numerical results it has been possible to:

- evaluate the robustness of the proposed algorithm against packet lost,
- verify the fact transmission delays can be effectively neglected by exploiting the MAC time stamp.

## X. Conclusions

In this work we addressed the clock synchronization problem in WSN for measurement applications. The proposed algorithm represents a trade-off between synchronization accuracy and network life-time preservation. A theoretical characterization of the convergence properties of the proposed algorithm has been provided. Experimental results confirmed the effectiveness of the proposed framework in a real WSN testbed. Future work will focus on the theoretical and experimental analysis of the synchronization accuracy versus the rate of packet collision events.

## Acknowledgments

This work was partially supported by the Italian grant RIDITT, project "DI.TR.IM.MIS Diffusione e trasferimento di tecnologie ad imprese nel settore delle misure", funded by the Italian Ministry of Economic Development. This work was partially supported by the Italian grant FIRB Futuro in Ricerca, project NECTAR, code RBFR08QWUV and the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257462 HYCON2 Network of excellence. The present work was partially financed by PRIN08 Robust synchronization technique for critical interference conditions in wireless network and stand alone measurement instruments. The co-author, Dr. Francesco Lamonaca, acknowledges the financial support of the Borsa di ricerca allestero di giovani soci del GMEE (Abroad Fellowship for young GMEE members), Edition 2011, provided by the Electric, and Electronic Measurement Group (GMEE).

## References

[1] A. Colombo, D. Fontanelli, D. Macii, and L. Palopoli, "Flexible indoor localization and tracking based on a wearable platform and sensor data fusion," *IEEE Trans. on Instr. and Meas.*, vol. in press, 2013.

[2] A. Gasparri and F. Pascucci, "An interlaced extended information filter for self-localization in sensor networks," *IEEE Trans. on Mobile Computing*, vol. 9, pp. 1491–1504, 2010.

[3] Bing Li, Yigang He, Fengming Guo, and Lei Zuo, "A novel localization algorithm based on isomap and partial least squares for wireless sensor networks," *IEEE Trans. on Instr. and Meas.*, vol. 62, no. 2, pp. 304–314, 2013.

[4] L. Fanucci, S. Saponara, T. Bacchillone, M. Donati, P. Barba, I. Sanchez-Tato, and C. Carmona, "Sensing devices and sensor signal processing for remote monitoring of vital signs in chf patients," *IEEE Trans. on Instr. and Meas.*, vol. 62, no. 3, pp. 553–569, 2013.

[5] A. Araujo, J. Garcia-Palacios, J. Blesa, F. Tirado, E. Romero, A. Samartin, and O. Nieto-Taladriz, "Wireless measurement system for structural health monitoring with high time-synchronization accuracy," *IEEE Trans. on Instr. and Meas.*, vol. 61, no. 3, pp. 801–810, Mar. 2012.

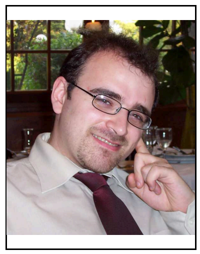
[6] L. Huang-Chen Lee, A. Banerjee, F. Yao-Min, L. Bing-Jean, and K. Chung-Ta, "Design of a multifunctional wireless sensor for in-situ monitoring of debris flows," *IEEE Trans. on Instr. and Meas.*, vol. 59, no. 11, pp. 2958–2967, Nov. 2010.

[7] A. Gasparri, B. Krishnamachari, and G.S. Sukhatme, "A framework for multi-robot node coverage in sensor networks," *Annals of Mathematics and Artificial Intel.*, vol. 52, pp. 281–305, 2008.

[8] H.C. Lee, Y.M. F., B.J. Lee, and C.T. King, "The tube: A rapidly deployable wireless sensor platform for supervising pollution of emergency work," *IEEE Trans. on Instr. and Meas.*, vol. 61, no. 10, pp. 2776–2786, 2012.
[9] J. Gutierrez, J.F. Villa-Medina, A. Nieto-Garibay, and M.A. Porta-Gandara, "Automated irrigation system using a wireless sensor network and gprs module," *IEEE Trans. on Instr. and Meas.*, vol. in press, 2013.
[10] F. Lamonaca, E. Garone, D. Grimaldi, and A. Nastro, "Localized fine accuracy synchronization in wireless sensor network based on consensus approach," in *Proc.of IEEE Inter. Instr. and Meas. Tech. Conference (I2MTC)*, 2012, pp. 2802 –2805.
[11] S. Rahamatkar, *A Light Weight Time Synchronization Approach in Sensor Network: Tree Structured Referencing Time Synchronization Scheme*, LAP Lambert Academic Publishing, Germany, 2012.
[12] K.S. Yildirim and A. Kantarci, "Time synchronization based on slow-flooding in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 244–253, 2014.
[13] S. El Khediri, N. Nasr, A. Kachouri, and A. Wei, "Synchronization in wireless sensors networks using balanced clusters," in *Proc. of 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, 2013, pp. 1–4.
[14] S. Merkel, C.W. Becker, and H. Schmeck, "Firefly-inspired synchronization for energy-efficient distance estimation in mobile ad-hoc networks," in *Proc. of IEEE 31st Inter. Conf. on Performance Computing and Comm.(IPCCC)*, 2012, pp. 205–214.
[15] M. Sasabe and T. Takine, "Continuous-time analysis of the simple averaging scheme for global clock synchronization in sparsely populated manets," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 782–793, 2013.
[16] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 1, pp. 1–13, 2007.
[17] R. Carli and S. Zampieri, "Networked clock synchronization based on second order linear consensus algorithms," in *49th IEEE Conference on Decision and Control (CDC)*, 2010.
[18] L. Schenato and F. Fiorentin, "Average timesynch:a consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.
[19] E. Garone, A. Gasparri, and F. Lamonaca, "Clock synchronization for wireless sensor network with communication delay," in *American Control Conference (ACC)*, 2013, pp. 771–776.
[20] F. Lamonaca, A. Gasparri, and E. Garone, "Wireless sensor networks clock synchronization with selective convergence rate," in *Intelligent Autonomous Vehicles (IAV)*, 2013, pp. 146–151.
[21] M. Yilin, E. Garone, A. Casavola, and B. Sinopoli, "Stochastic sensor scheduling for energy constrained estimation in multi-hop wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2489 –2495, Oct. 2011.
[22] Jianshe Wu, Licheng Jiao, and Ranran Ding, "Average time synchronization in wireless sensor networks by pairwise messages," *Computer Communications*, vol. 35, no. 2, pp. 221 – 233, 2012.
[23] M. Dahleh, A. Dahleh, and G. C. Verghese, *Chapter 15 - External Input Output Stability*, Lecture Notes MIT Course, 1999.
[24] "Crossbow technology, telosb, telosb mote platform, document part number: 6020-0094-01 rev b," .

**Francesco Lamonaca** (M'10), received the M.S. degree in Computer Science Engineering in 2005 and PhD degree in Computer and System Science in 2010 from the University of Calabria, Rende, Italy. He received doctorate equivalences in Science (2010) and Engineering Science (2011) from the Universit Libre de Bruxelles, Belgium. Since 2010, he has been a Postdoctoral Researcher of Electrical and Electronic Measurements at the Department of Computer Sciences, Modelling, Electronics and System Science of the University of Calabria. He was classified first winner of the research funding competition of projects of Young Researchers 2010, and 2012 at the University of Calabria. He was a scholarship winner of the 2011 GMEE National competition for holding a research abroad with the project "Synchronization Techniques for Wireless Sensors based on Consensus". He is a member of the Italian Group of Electrical and Electronic Measurement (GMEE), Institute of Electrical and Electronics Engineers (IEEE); Instrumentation and Measurement Society; International Association of Hydro-Environment Engineering and Research (IAHR). Dr. Lamonaca has authored and co-authored over 90 papers published in international journals and conference proceedings. His current researches include synchronization of networking measurement instruments, wireless sensor network, environmental monitoring, image processing for measurement applications, digital signal processing for monitoring and testing, virtual instrumentation, distributed measurements, and measurement for medical use.

**Andrea Gasparri** received the Graduate degree (cum laude) in computer science in 2004 and the Ph.D. degree in computer science and automation in 2008, both from the University of Rome "Roma Tre", Rome, Italy. He is currently an Assistant Professor at the Engineering Department, University of Rome "Roma Tre". His current research interests include mobile robotics, sensor networks, and, more generally, networked multi-agent systems.



**Emanuele Garone** Emanuele Garone obtained his Laurea degree in 2005 and his Ph.D degree in Systems Engineering from the University of Calabria, Italy, in 2009. Since 2010 he is Assistant Professor at the Control Design and System Analysis department of the Universit Libre de Bruxelles, Belgium. His current research interests include: model predictive control, distributed reference governor, networked control systems and heterogeneous networks of vehicles.

**Domenico Grimaldi** (M'94, SM'10) is Full Professor of Electronic Measurement at the Department of Computer Sciences, Modelling, Electronics and System Science of the University of Calabria-Italy. He has remained there in a variety of research and management positions. He is responsible of the Laboratory for processing the measurement information. He was responsible of the research unit in the frame of National Projects PRIN, FIRB and international projects INTERLINK supported by the Italian Ministry for University and Research. He was responsible in the Tempus Project and Leonardo da Vinci Project supported by the European Union. He was delegate of the Rector of the University of Calabria. He is a member of the Italian Institute of Electrical Engineers (AEI), and Italian Group of Electrical and Electronic Measurements (GMEE), from 1998 to 2000 he was AEI vice-president for the Calabria Region. Prof. Grimaldi has authored and co-authored over 230 papers published in international journals and conference proceedings. His current researches include the characterization of measurement transducers, digital signal processing for monitoring and testing, distributed measurements, synchronization, telecommunication system measurement, and measurement for medical application.