# Set–Valued Consensus for Distributed Clock Synchronization

Adriano Fagiolini, Simone Martini, and Antonio Bicchi

***Abstract*— This paper addresses the clock synchronization problem in a wireless sensor network (WSN) and proposes a distributed solution that consists of a form of consensus, where agents are able to exchange data representing intervals or sets. The solution is based on a centralized algorithm for clock synchronization, proposed by Marzullo, that determines the smallest interval that is in common with the maximum number of measured intervals. We first show how to convert such an algorithm into a problem involving only operations on sets, and then we convert it into a set–valued consensus. The solution is valid for more general scenarios where agents have uncertain measures of e.g. the position of an object detected by a vision system, a temperature in a room, but it will be applied to the case where a set of uncertain time values are propagated through a WSN. Under suitable joint conditions on the communication connectivity and bounded agent failure, we prove the correctness of the algorithm that indeed allows the network agents to consent on the value of a unique global time.**

## I. Introduction

The proliferation of robotic devices and the growing number of their potential applications has recently led to an increased interest towards multiple–robot applications, such as consensus, rendezvous, sensor coverage, and simultaneous localization and mapping. All these applications demand for the availability of a global clock, i.e. every node in the network must be able to refer to a unique time. The reason for this necessity is twofold. First, only if a very accurately synchronized clock is available, every node can get a global picture of an event that is sensed by several nodes. Secondly, clock synchronization is essential for reducing node's energy consumption due to communication. Indeed, any two node willing to exchange a message with each other establish a rendezvous. The better the clock synchronization, the less energy is wasted in the necessary guard times to not miss the rendezvous point.

In a centralized system the solution of this problem is trivial: the centralized server will just decide the system's time. In a distributed system, the problem takes on more complexity because a global time is not easily known. In a WSN, clock synchronization poses two major problems. The former is the connectivity, which is related to the fact that nodes of a sensor network cannot directly communicate with each other, and some information may need to be relayed by other nodes. Therefore, it is not possible to choose a reference node to which all other nodes can be synchronized to. The latter problem is related to unpredictable random delays that may normally occur between any pair of nodes. It is indeed known that delivery time of radio messages in WSN is subject to interferences, and node failures which may cause unknown variations to the standard communication time.

In this respect, during the last years clock synchronization in distributed systems has been extensively studied. As a result of this effort, many different approaches have been proposed (see e.g. [1]–[6]). Specifically developed for WSN are the Reference Broadcast Synchronization (RBS) [7], Timing–sync Protocol for Sensor Networks (TPSN) [8] and the Precision Time Protocol (PTP) [9]). More in detail, the RBS exploits the broadcast nature of the physical channel to synchronize a set of receivers with one another. The timestamp of the reception of a broadcast message is recorded at each node and these timestamps are exchanged to calculate relative clock offsets between nodes. The TPSN algorithm builds a spanning tree of the network during the level discovery phase. In the synchronization phase of the algorithm, nodes synchronize to their parent in the tree by a two-way message exchange. PTP is a time-transfer protocol defined in the IEEE 1588 standard that allows precise synchronization of networks (e.g., Ethernet).

More recently, so–called Average TimeSync protocol [10] has been proposed as a different approach to clock synchronization. The main idea underlying this approach is to average local information to achieve a global agreement on a specific quantity of interest, and this is obtained by transposing the synchronization problem into a linear consensus problem. Notwithstanding, the problem of clock synchronization in a WSN is far from been completely solved. As a matter of fact, available solutions, such as NTP, require operating conditions that are not guaranteed in a WSN. These difficulties are due to limited energy and bandwidth availability, that are in turn necessary to allow sensors a longer operating life. Furthermore, the fact that the topology is dynamically changing is another issue that makes the clock synchronization problem in a WSN more difficult than in traditional network, and actually a very challenging one.

Moreover, there has been a recent thrust toward the use of nonlinear consensus in different application domains (see e.g. [11]–[13]). The main motivations for this is the fact that many control and robotics problem can not be solved by simple linear consensus, or at least it would be more natural to formulate solutions involving more general forms of consensus. In [12], a set–valued consensus algorithm was used in a security context to allow a set of agents reach an agreement on the presence of misbehaving neighbors.

A. Fagiolini, S. Martini, and A. Bicchi are with the Interdepartmental Research Center "E. Piaggio" of the Università di Pisa, Italy, fagiolini@ieee.org, simone.martini@ing.unipi.it, bicchi@ing.unipi.it.

In this paper, we will proceed along the same line, by transposing the clock synchronization problem into a consensus problem on sets. This is achieved by exploiting the idea that uncertain measures of a clock that have been propagated through the network can be represented as interval. This idea was first proposed by Marzullo in [14] that showed a centralized algorithm to determine the smallest confidence interval that is contained in the largest number of sensor measures. The solution therein proposed now forms the basis of the Network Time Protocol (NTP) [15], a protocol that is implemented in many software platforms and operating systems. NTP sets and maintains the system time of day in synchronism with Internet standard time servers. NTP does most computations in 64-bit floating–point arithmetic and does relatively clumsy 64–bit fixed–point operations only when necessary to preserve the ultimate precision, about $2.32 \times 10^{-10}$ seconds (232 picoseconds). While the ultimate precision is not achievable with ordinary workstations and networks of today, it may be required with future gigahertz CPU clocks and gigabit LANs. However, Marzullo's solution is actually centralized and is extended to a distributed approach in this paper.

The outline of this paper is the following. In Section II, an extension of Marzullo's problem for clock synchronization is introduced, and a centralized translation involving only operations on sets is derived. In Section III a distributed version of the algorithm is found that consists of a set–valued consensus system. Finally, in Section IV, the effectiveness of the proposed solution is shown through some numerical simulations.

## II. A Centralized Extension of Marzullo's Algorithm

Consider the clock synchronization algorithm originally proposed by Marzullo in [14]. In this section, we propose an extension of the algorithm and we show how to convert it into a problem involving only operations on sets.

Suppose to have $n$ sensors that are able to measure the value of quantity of interest within a confidence interval or set. Let us denote with $u \in U$ this quantity that may range from time, temperature, to the position of an object during a SLAM application, or to the configuration of a neighboring car (as e.g. in the IDS considered in [12]).

Suppose that one or more of these sensors may fail and produce a confidence interval or set that is not consistent with the others. Then, given these $n$ sets, $U_1, \ldots, U_n$, we want to compute the smallest set $Y$ that is contained in the largest number of such sensor measures. This quantity represents the set that is most likely to contain the real clock value. A solution to this problem can be found that uses only operations on sets (union $\cup$, intersection $\cap$, and complementation $\mathcal{C}(\cdot)$).

Given $n$ agents' indices $1, 2, \ldots, n$, consider the number of combinations of $i$ of such indices out of $n$ being specified by

$$c(n,i) \stackrel{\text{def}}{=} \left( \begin{array}{c} n \\ i \end{array} \right) .$$

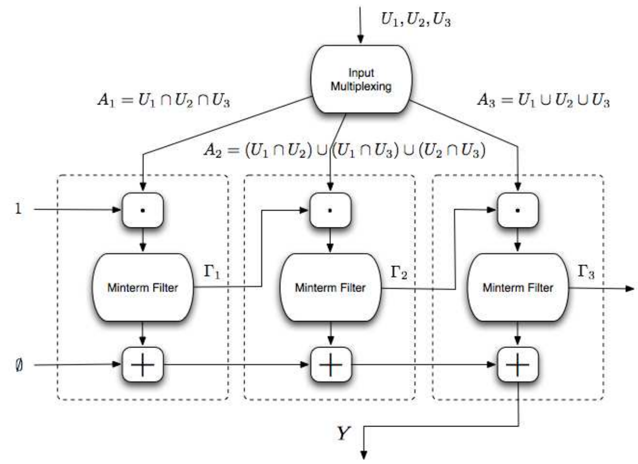



Fig. 1. An instance of the algorithm to solve the set–valued formulation of Marzullo's problem with 3 input sets.

Then, consider the sets

$$A_i = \bigcup_{l=1}^{c(n,n-i+1)} \bigcap_{h=1}^{n-i+1} U_{i_{l,h}} ,$$

for $i = 1, 2, \ldots, n$, where $i_{l,1}, \ldots, i_{l,n-i+1}$ are distinct combinations of agents' indices. More explicitly, the sets are given by

- $A_1 = U_1 \cap U_2 \cap \cdots \cap U_n$ (the intersection of all $n$ sensor measures),
- $A_2 = (U_1 \cap U_2 \cap \cdots \cap U_{n-1}) \cup (U_1 \cap U_2 \cap \cdots \cap U_{n-2} \cap \cap U_n) \cup \ldots$ (the union of the $n$ possible intersections of $n-1$ sensor measures),
- $\cdots$,
- $A_n = U_1 \cup U_2 \cup \ldots \cup U_n$ (the union of the $n$ individual sensor measures).

Consider also the following filtering sets:

$$\Gamma_1 = U ,$$
$$\Gamma_i = \Gamma_{i-1} \cap \left\{ \begin{array}{ll} U & \text{if } A_{i-1} = \emptyset , \\ \emptyset & \text{if } A_{i-1} \neq \emptyset , \end{array} \right. \quad \text{for } i = 2, ..., n . \tag{1}$$

Then, the desired confidence set $Y$ is readily given by

$$Y = \bigcup_{i=1}^{n} (\Gamma_i \cap A_i) \stackrel{\text{def}}{=} \mathsf{M}(U_1, \ldots, U_n) . \tag{2}$$

We can give the following

*Definition 1:* Given $n$ set measures $U_1, \ldots, U_n$, we say that $U_j$ is *consistent* if it shares an intersection with the smallest set that is common with the maximum number of the other sets, i.e.

$$U_j \cap \mathsf{M}(U_1, \ldots, U_n) \neq \emptyset .$$

Conversely, we say that $U_j$ is *inconsistent*.

A pictorial representation of the algorithm is reported in Fig. 1. The figure outlines the modular structure of the algorithm, where the output of a module is the input of the following one.

Once the best estimated set $Y$ has been computed from the initial collection $X_1, \dots, X_m$, one has typically to extract a scalar value $b \in \mathcal{X}$ to be used in a control loop. For time synchronization, the complete set is $\mathcal{X} = [0, \infty)$, and a common choice is to take the earliest interval $[t_{min}, t_{max}] \subseteq Y$ and extract its middle value

$$b = \frac{t_{max} - t_{min}}{2}\,.$$

*A. Example*

Suppose to have the interval $U = [0, \infty)$, and $m = 3$ sensors providing the following confidence interval: $U_1 = [1, 10]$, $U_2 = [30, 40]$, and $U_3 = [6, 29]$. In this case, map $\phi$ computes the following 3 intervals:

$$\begin{aligned} A_1 &= U_1 \cap U_2 \cap U_3 = \emptyset\,, \\ A_2 &= (U_1 \cap U_2) \cup (U_1 \cap U_3) \cup (U_2 \cap U_3) = [6, 10]\,, \\ A_3 &= U_1 \cup U_2 \cup U_3 = [1, 29] \cup [30, 40]\,. \end{aligned}$$

The filtering sets are the intervals $\Gamma_1 = U$, $\Gamma_2 = U$, and $\Gamma_3 = \emptyset$. Thus, the smallest interval that is in common to the largest number of the given intervals is

$$\begin{aligned} Y &= (\Gamma_1 \cap A_1) \cup (\Gamma_2 \cap A_2) \cup (\Gamma_3 \cap A_3) = \\ &= ([0, \infty) \cap \emptyset) \cup ([0, \infty) \cap [6, 10]) \cup \\ &\quad \cup (\emptyset \cap ([1, 29] \cup [30, 40])) = \\ &= [6, 10]\,, \end{aligned}$$

which is contained in $U_1$ and $U_2$ (see Fig. II-A). On the contrary, the third sensor's measure is faulty. Indeed, we have:

$$Y \cap U_3 = \emptyset\,.$$

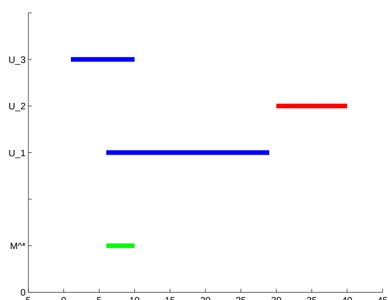



Fig. 2. Example of clock interval estimated by the centralized solution of Marzullo's algorithm. The estimated set is in green, the correct measures $U_1$ and $U_2$ are in blue, and the inconsistent measure $U_3$ is in red.

## III. Distributed Clock Synchronization

The algorithm presented in Section II is able to solve the extended Marzullo's problem only in a centralized setting, whereas we want that every agent have a consistent information on the quantity of interest $u$, so that any of them can be polled by an external user. To this aim, let us suppose that every generic agent $\mathcal{A}_i$ has a state $X_i \subseteq U$ that represents its estimate of the quantity of interest $u$ and is able to share the value of its state with all its neighbors by exchanging a message with them. Suppose that every agent initializes its state with the value of its local estimate of the quantity of interest, i.e.

$$X_i(0) \leftarrow U_i(0)\,.$$

Then, we want to design a distributed iteration rule of the form

$$X(t+1) = F(X(t))\,,$$

where $X = (X_1, \dots, X_n)^T$ is the system's state, and $t$ is a discrete time, such that, starting from any initial state $X(0)$, every agent will consent on Marzullo's centralized decision, i.e. there exists a finite time $\bar{t}$ such that

$$X(\bar{t}) = \mathsf{M}(U(0)) = \mathsf{M}(X(0))\,.$$

Furthermore, due to the result stated in [16] and concerning the *impossibility to reach a consensus* with corrupted data, we must add a further hypothesis to the problem guaranteeing that the maximum number of inconsistent measures is at most $\gamma$. We will refer to this as the *bounded inconsistency hypothesis*.

From [17], recall the following

*Definition 2:* A graph $G = (V, E)$ is said to be *k–connected* if there does not exist a set of $k - 1$ vertices in $V$ whose removal disconnects the graph, i.e. the vertex connectivity of $G$ is greater or equal to $k$.

Therefore, a connected graph is 1–connected, and a biconnected graph is 2–connected.

First of all, note that the hypothesis of bounded inconsistency implies that there exists at least one combination of $n - \gamma$ sets that share an intersection with Marzullo's centralized decision, i.e.

$$\exists\, i_1, \cdots, i_{n-\gamma} \in \{1, \dots, n\} \,|\, X_{i_j} \cap M^* \neq \emptyset\,, \qquad (3)$$

where $M^* = \mathsf{M}(X_1(0), \dots, X_n(0))$. In case of virtuous scenario with only consistent measures ($\gamma = 0$), this condition implies that

$$X_1(0) \cap \cdots \cap X_n(0) \neq \emptyset\,,$$

and a solution to the problem can be obtained by simply replicating the M-Algorithm on every node according to its communication neighbors, i.e.

$$F_i(X) = \mathsf{M}(X_{i_1}, \cdots, X_{i_{n_i}})\,, \text{for } i = 1, \dots, n\,,$$

where $n_i$ is any number of agent $\mathcal{A}_i$'s neighbors, and $i_1, \cdots, i_{n_i}$ are their indices. Indeed, in the virtuous hypothesis, the algorithm on agent $\mathcal{A}_i$ reduces to a pure intersection of the data received from its communication neighbors. As it is well-known, set intersection is associative, commutative, and idempotent ($X \cap X = X$). Therefore, given that the underlying communication graph is connected, the network convergence toward the centralized decision $X^*$ is guaranteed ( [12]). Thus its distributed application allows the agents to consent on the value of the centralized intersection, which is also the desired solution of the extended Marzullo's problem. Consider the case with $\gamma > 0$. Suppose that all initial measures are represented by *compact* sets so that, if a common intersection exists, the intersection itself is

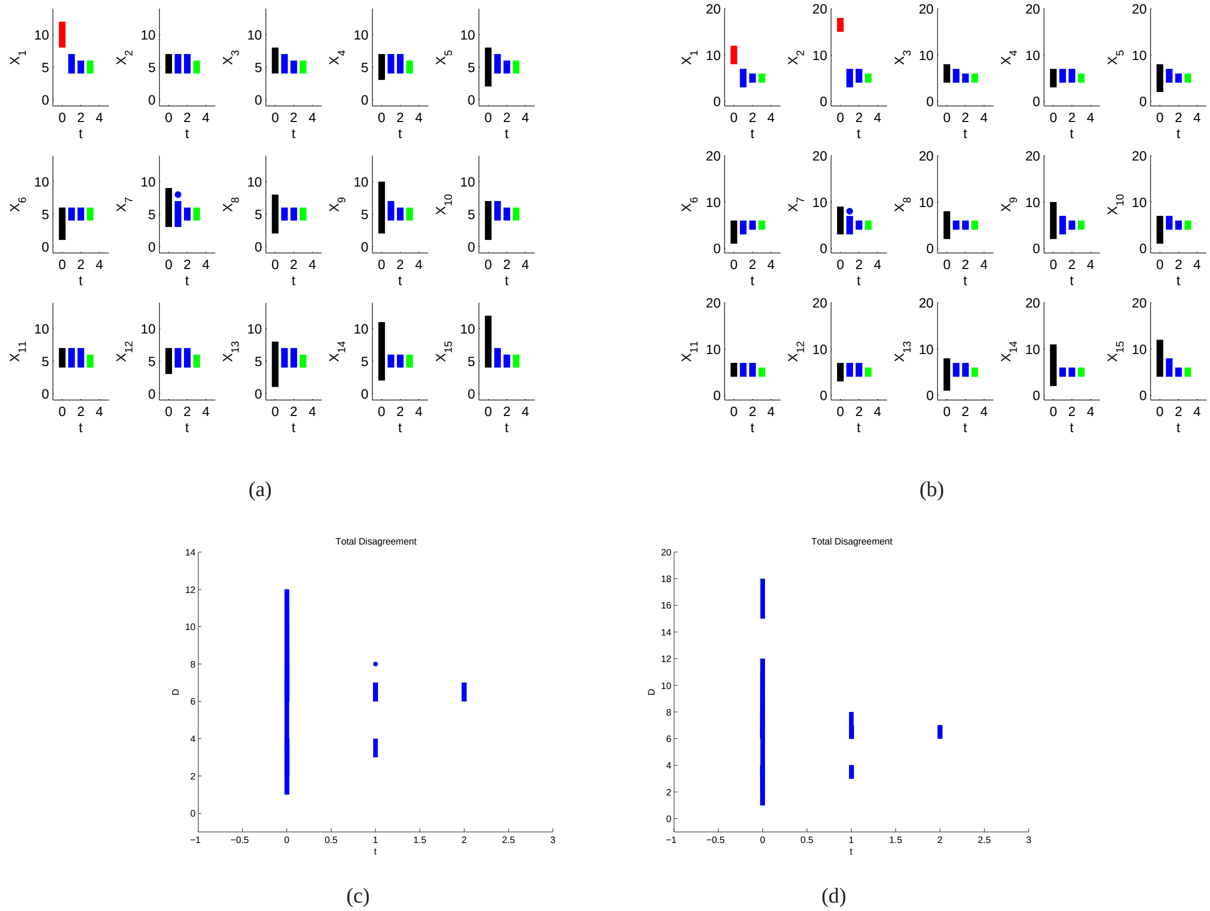



Fig. 3. Simulation run of a dynamic systems estimating Marzullo's solution with $\gamma = 1$ (a), and $\gamma = 2$ (b) possible inconsistencies, and corresponding behavior of the network disagreement w.r.t. Marzullo's centralized decision (d,e).

a compact set. Due to this, all points in the Marzullo's centralized decision are in common with the very same initial sets. In this case, the analysis of convergence is more complex. Anyway, we are able to prove the following

*Theorem 1:* Consider a network of agents evolving according to the iterative rule

$$\begin{aligned} X_i(t+1) &= F_i(X) = \mathsf{M}\left(X_{i_1}(t), \cdots, X_{i_{n_i}}(t)\right), \\ X_i(0) &= U_i\,, \end{aligned}$$

for $i = 1, \ldots, n$, where at most $\gamma$ initial states may be inconsistent. The network can reach a consensus on the centralized Marzullo decision

$$X^* = (M^*, \cdots, M^*)^T\,,$$
$$M^* \stackrel{\text{def}}{=} \mathsf{M}\left(X_1(0), \ldots, X_n(0)\right),$$

if the agents can exchange messages according to a communication matrix $C$ that is at least $r$–connected, with $r = 2\,\gamma + 1$. The consensus is achieved in at most $n$ steps.

*Proof:* We have to prove that if $r = 2\,\gamma + 1$, the distributed algorithm converges to the centralized version of the Marzullo Algorithm $(M^*)$. First observe that, under bounded inconsistency, the algorithm of Marzullo's centralized decision returns the same value of the algorithm itself truncated at the $\gamma + 1$–th term of the union in Eq. 2. This happens because of Eq. 3 which implies that $\Gamma_i = \emptyset$, for $i \geq \gamma + 2$. Thus, the algorithm itself reduces to

$$Y = \bigcup_{i=1}^{\gamma+1} \left(\Gamma_i \cap A_i\right)\,. \tag{4}$$

Moreover, note that, by construction, it holds the general property

$$A_i \subseteq A_{i+1}\,, \quad \text{for}\, i = 1, \ldots, n-1.$$

which implies that an algorithm computing only the term $A_i$ returns a value that is upper bounded by an algorithm computing only the term $A_{i+1}$. We replicate the Algorithm on every node according to its communication neighbors,, i.e.

$$F_i(X) = \mathsf{M}\left(X_{i_1}, \cdots, X_{i_r}\right), \text{for}\, i = 1, \ldots, n\,,$$

Consider the worst case, i.e. a node is connected with all the faulty nodes. We define $c_i = \left\{a, .., f\right\}$ and $nc_i = \left\{h, ..., v\right\}$ with $\#c_i = \gamma + 1$ and $\#nc_i = \gamma$ the sets available at the node $i$, containing the indices of the consistent measure and the indices of the inconsistent measures respectively. The bounded inconsistency hypothesis guarantees that

$$X_k \cap M^* \neq \emptyset\,, \forall k \in c_i.$$

As $X_k$ and $X_j$, $k, j \in c_i$ are two consistent measures, then it must hold $M^* \subseteq X_k, X_j$, which also implies that

$$X_k \cap X_j \neq \emptyset \quad \forall k, j \in c_i .$$

In other words we have

$$I_{c_i} = \bigcap_s X_s \neq 0 \,, \quad s \in c_i , \tag{5}$$

$$I_{nc_i} = \bigcap_s X_p \quad s \in nc_i , \tag{6}$$

being $I_{c_i}$ the intersection between consistent measure, and $I_{nc_i}$ the intersection between inconsistent measure. Note that while $I_{c_i}$ is always different from the empty-set, $I_{nc_i}$ can be an emptyset or not.

Moreover, since $I_{nc_i}$ are inconsistent values, it is easy to verify that

$$\begin{array}{l} I_{nc_i} \cap I_{c_i} = \emptyset \\ \bigcup_p (I_{c_i} \cap X_p) = \emptyset \quad p \in nc_i \end{array}$$

Recalling the update rule in (1) we now have

$$M\left(X_{i_1}, \cdots, X_{i_r}\right) = \left(\bigcap_{s_i} X_{s_i}\right) = I_{c_i}$$

where $s_i \in c_i$ . This is because the M-Algorithm computes the smallest confidence set that is contained in the largest number of such sensor measures, and we have that $\#c_i > \#nc_i$. This guarantees that the inconsistent measures do not affect the state X, as it also happens for the centralized execution of M-algorithm (see Section II). Under bounded inconsistency hypothesis, execution of the algorithm at the generic agent $\mathcal{A}_i$ reduces to the intersection of any data received by its neighbors. Furthermore, the update rule in (1) is associative, distributive, and idempotent w.r.t. its input arguments. Since the the communication graph is connected, according to [12], the convergence of the dynamic system toward the centralized decision is guaranteed in at most $n$ steps and it proves the thesis. ■

## IV. Simulation

The effectiveness of the proposed solution is shown through numerical simulations with $n = 15$ agents. We have considered 2 different scenarios with $\gamma = 1, 2$ possible inconsistent measures. Agents are able to communicate according to a graph, that is not reported for the sake of space, but that is chosen so as to guarantee the minimum redundancy required. Fig. 3(a) reports the behavior of the dynamic system starting from an initial condition where only one agent, agent $\mathcal{A}_1$, has an inconsistent measure. In this case, the required number of neighbors in the communication graph is $r = 2\,\gamma + 1 = 3$. The figure shows that the inconsistent measure is tolerated, and that every agent reach the correct final clock interval. Fig. 3(b) refers to a simulation of a dynamic system that has been design under the hypothesis of $\gamma = 2$ possible inconsistencies. In this case, $r = 5$ neighbors in the communication graph are required. The figure shows that the inconsistent measures of agents $\mathcal{A}_1$ and $\mathcal{A}_2$ are tolerated. In both figures, the final decision coinciding with the Marzullo's centralized estimated interval is reported in green. The inconsistent measures are instead colored in red. Finally, both figures reports the behavior of network's *relative disagreement*

$$E \stackrel{\text{def}}{=} \sum_{i=1}^{n} \mathcal{D}(X_i, \mathsf{M}\left(X(0)\right),$$

where $\mathcal{D}$ is the vector distance, based on the symmetric difference between two intervals. The figure shows that the disagreement becomes $\emptyset$ in a number of steps less or equal to the diameter of the chosen communication graph. Recall that the diameter of a graph is the maximum distance between any two nodes in the graph, and the distance is the length of the minimum path connecting the nodes.

As we have stated, the proposed solution is valid also when the data that is exchanges is a set. To show this, we conclude with a further example with $U = [0, \infty) \times [0, \infty)$, and $m = 4$ sensors providing the following measured confidence sets:

$$\begin{array}{rcl} U_1 &=& [2,5] \times [1,6]\,, \\ U_2 &=& [8,14] \times [3,8]\,, \\ U_3 &=& [1,10] \times [4,9]\,, \\ U_4 &=& [8,13] \times [0,2]\,. \end{array}$$

Before showing the simulation results, let us first compute the centralized solution. As described in Section II, this involves computation of the following sets

$$\begin{array}{rcl} A_1 &=& U_1 \cap U_2 \cap U_3 \cap U_4 = \emptyset\,, \\ A_2 &=& (U_1 \cap U_2 \cap U_3) \cup (U_1 \cap U_2 \cap U_4) \cup \\ &\cup& (U_2 \cap U_3 \cap U_4) = \emptyset\,, \\ A_3 &=& (U_1 \cap U_2) \cup (U_1 \cap U_3) \cup (U_1 \cap U_4) \cup \\ &\cup& (U_2 \cap U_3) \cup (U_2 \cap U_4) \cup (U_3 \cap U_4) = \\ &=& \emptyset \,\cup\, ([2,5] \times [4,6]) \,\cup\, \emptyset \cup \\ &\cup& ([8,10] \times [4,8]) \,\cup\, \emptyset \,\cup\, \emptyset = \\ &=& ([2,5] \times [4,6]) \,\cup\, ([8,10] \times [4,8])\,, \\ A_4 &=& U_1 \cup U_2 \cup U_3 \cup U_4 = \\ &=& ([2,5] \times [1,6]) \,\cup\, ([8,14] \times [3,8]) \cup \\ &\cup& ([1,10] \times [4,9]) \cup ([8,13] \times [0,2]). \end{array}$$

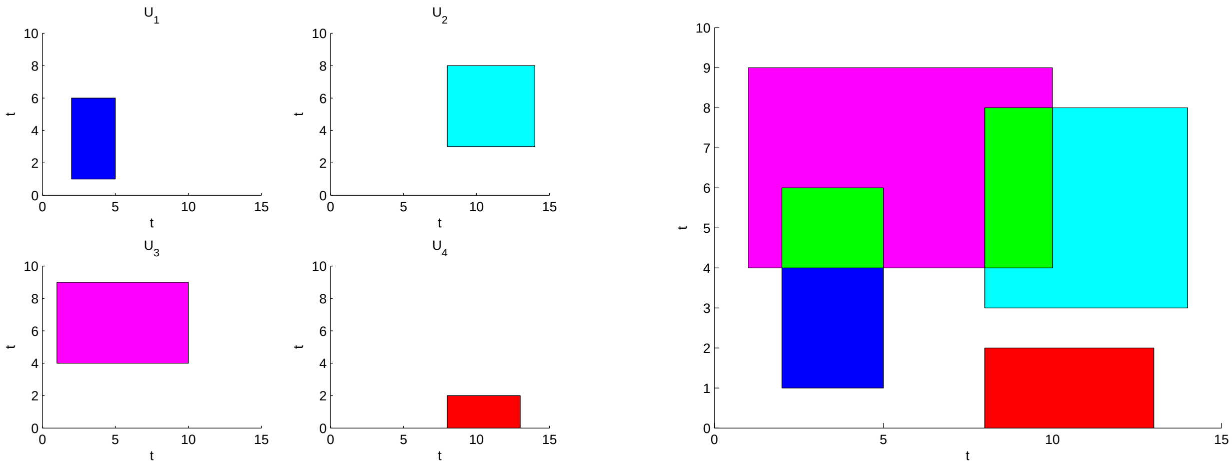
The filtering sets are $\Gamma_1 = U$, $\Gamma_2 = U$, $\Gamma_3 = U$, and $\Gamma_4 = \emptyset$. Thus the smallest set that is in common to the largest number of the given sets is

$$\begin{array}{rcl} Y &=& (\Gamma_1 \cap A_1) \cup (\Gamma_2 \cap A_2) \cup (\Gamma_3 \cap A_3) \cup (\Gamma_4 \cap A_4) = \\ &=& ([0,\infty) \cap \emptyset) \,\cup\, ([0,\infty) \cap \emptyset) \cup \\ &\cup& ([0,\infty) \cap A_3) \cup\, (\emptyset \cap A_4) = \\ &=& ([2,5] \times [4,6]) \,\cup\, ([8,10] \times [4,8])\,, \end{array}$$

which is partially contained in $U_1$ and $U_3$, and partially contained in $U_2$ and $U_3$. On the contrary, the forth sensor's measure is faulty. Indeed, we have:

$$Y \cap U_4 = \emptyset\,.$$

Fig. 4 refers to a simulation of the dynamic system that has been design under the hypothesis of one possible faulty node. Fig 4(a) shows the initial status $X(0)$ of the agents, while Fig. 4(b) reports the final decision coinciding with the Marzullo's centralized estimated interval is reported in green. The inconsistent measure is instead colored in red.




(a) (b)

Fig. 4. Simulation run with $n = 4$ sensors that have four uncertain measures of a quantity of interest on the plane (a). The final result show that the inconsistent set (in red) has been excluded and tolerated (b).

Simulation show that the centralized estimate is achieved also by distributed execution of the algorithm.

## V. Conclusion and Future Work

This paper presented a distributed set–valued algorithm for solving clock synchronization in a WSN. The algorithm is based on a centralized approach, proposed by Marzullo, that represents uncertain measures of a clock trasmitted though a network channel as an interval. Based on previous work on set–valued consensus, the authors showed that the clock synchronization can be transposed into a problem involving only operations on sets. Effectiveness of the solution was shown through simulation and showed that the algorithm is also able to tolerate a maximum number of faulty sensors.

Future development will concern the implementation of the algorithm on a real WSN, and its performance evaluation compared to standard clock synchronization techniques that are in use today.

## VI. Acknowledgment

This work has been partially supported by the European Commission with contract FP7-IST-2008-224428 “CHAT - Control of Heterogeneous Automation Systems: Technologies for scalability, reconfigurability and security”, and with contract number FP7-2007-2-224053 CONET, the “Cooperating Objects Network of Excellence”. This work has been also partially supported by Research Project 2007 funded by Cassa di Risparmio di Livorno, Lucca e Pisa.

## References


[1] K. Römer, “Time synchronization in ad hoc networks,” *In Mobi Hoc’01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, 2001.

[2] M. Sichitiu and C. Veerarittiphan, “Simple, accurate time synchronization for wireless sensor networks,” *In WCNC’03: Proceedings of the IEEE Wireless Communications and Networking Conference*, 2003.

[3] H. Dai and R. Han, “Tsync: a lightweight bidirectional time synchronization service for wireless sensor networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, no. 1, 2004.

[4] J. van Greunen and J. Rabaey, “Lightweight timesynchronization for sensor networks,” *In WSNA’03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003.

[5] M. Maròti, B. Kusy, G. Simon, and A. Lèdeczi, “The flooding time synchronization protocol,” *In SenSys’04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004.

[6] R. Solis, V. Borkar, and P. Kumar, “A new distributed time synchronization protocol for multi hop wireless networks,” *45th IEEE Conference on Decision and Control*, 2006.

[7] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPSOper. Syst. Rev.*, vol. 36, 2002.

[8] S. Ganeriwal, R. Kumar, and M. Srivastava, “Timing-sync protocol for sensor networks,” *In SenSys’03: Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003.

[9] J. Eidson and K. Lee, “Ieee 1588 standard for a precision clock synchronization protocol for networked measurement and control systems,” *Sensors for Industry Conference, 2002. 2nd ISA/IEEE*, pp. 98– 105, Nov. 2002.

[10] L. Schenato and G. Gamba, “A distributed consensus protocol for clock synchronization inwireless sensor network,” *46th IEEE Conference on Decision and Control*, Dec 2007.

[11] J. Cortès, “Distributed algorithms for reaching consensus on general functions,” *Automatica*, vol. 44, no. 3, pp. 726–737, Mar. 2008.

[12] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, “Consensus based Distributed Intrusion Detection for Multi Robot Systems,” in *Proc. IEEE International Conf. on Robotics and Automation*, 2008, pp. 120–127.

[13] A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, “Decentralized Intrusion Detection For Secure Cooperative Multi-Agent Systems,” in *Proc. 46th IEEE Conf. on Decision and Control*, 2007, pp. 1553–1558.

[14] K. Marzullo, “Maintaining the time in a distributed system: An example of a loosely-coupled distributed service.” *Dissertation Abstracts International Part B: Science and Engineering[DISS. ABST. INT. PT. B- SCI. & ENG.],*, vol. 46, no. 1, 1985.

[15] D. Mills, “Internet time synchronization: the network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, Oct 1991.

[16] M. Fischer, N. Lynch, and M. S. Paterson, “Impossibility of distributed consensus with one faulty process,” *Journal of the Association for Computing Machinery (ACM).*, vol. 32, no. 2, pp. 374–382, April 1985.

[17] C. D. Godsil and G. F. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. New York: Springer Verlag, 2001, vol. 207.