A PI consensus controller with gossip communication for clock synchronization in wireless sensors networks

Saverio Bolognani[1], Ruggero Carli[2], Sandro Zampieri[3]

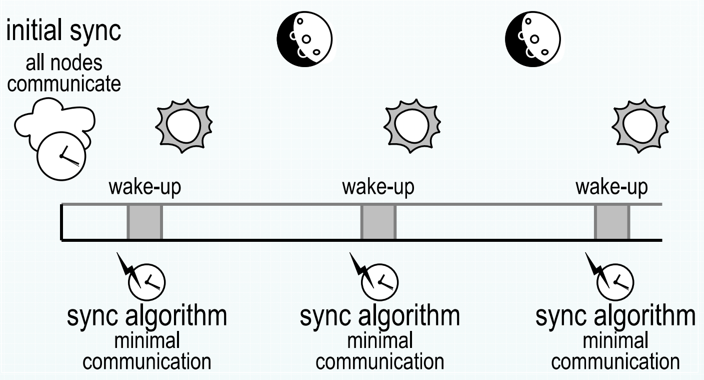
[1]Department of Information Engineering, University of Padova, Italy, saverio.bolognani@dei.unipd.it
[2]CCDC, University of California at Santa Barbara, CA 93106, USA, carlirug@engr.ucsb.edu
[3] Department of Information Engineering, University of Padova, Italy, zampi@dei.unipd.it

Clock synchronization in wireless sensor networks

Several WSN application either benefit from, or require, **time synchronization**:

- mobile target tracking
- habitat monitoring
- power scheduling and TDMA communication
- coordination of powerline nodes

Consider a network in which every agent can communicate with another agent in a bidirectional fashion. At regular time intervals node **wake up** and **communicate with a random neighbor**. If their clocks are synchronized, the wake up window can be thinned to reduce energy consumption.



time interval between updates	T	1 day = 86400 s
clock offset standard deviation	σ_x	$10^{-5}T$
clock skew standard deviation	σ_v	10^{-6}

Clock dynamics (affected by **offset** and **skew** errors):

$$x_i(t+1) = x_i(t) + d_i$$

PI-like algorithm proposed by

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008) "A PI consensus controller for networked clocks synchronization."

$$x_i(t+1) = x_i(t) + d_i + u_i(t)$$
$$w(t+1) = w(t) - \alpha K x(t)$$
$$u(t) = w(t) - K x(t)$$

Where K must be compatible with the communication graph and protocol. α in (0,1) is a design **parameter** to be chosen .

Algorithm convergence and scalability

Proposed algorithm:

$$x_i(t+1) = \frac{1}{2}(x_i(t) + x_j(t)) + w_i(t) + d_i(t)$$
$$x_j(t+1) = \frac{1}{2}(x_i(t) + x_j(t)) + w_j(t) + d_j(t)$$
$$w_i(t+1) = \frac{\alpha}{2}(-x_i(t) + x_j(t)) + w_i(t)$$
$$w_j(t+1) = \frac{\alpha}{2}(-x_j(t) + x_i(t)) + w_j(t)$$

compatible with a **random gossip** communication scheme.

Consider the special case of **complete graph** and **uniform probability** for all edges. In this case it is possible to derive a **mean square analysis** of the synchronization error

$$P(t) = \mathbf{E}\left[\begin{bmatrix}\bar{x}(t)\\ \bar{v}(t)\end{bmatrix}\left[\bar{x}(t)^* \ \bar{v}(t)^*\right]\right] = \begin{bmatrix}P_{11} & P_{12}\\ P_{21}^* & P_{22}\end{bmatrix}$$

In the **convenient parametrization**

$$P_{ij} = p_{ij}\Omega, \quad \Omega = I - \frac{1}{N}\mathbf{1}\mathbf{1}^*$$

the mean square error evolves according to the linear system

$$\begin{bmatrix}p_{11}(t+1)\\ p_{12}(t+1)\\ p_{22}(t+1)\end{bmatrix} = \begin{bmatrix}\frac{N-2}{N-1} & 2\frac{N-2}{N-1} & 1\\ 0 & 1-\frac{\alpha+1}{N-1} & 1\\ \frac{\alpha^2}{N-1} & -\frac{2\alpha}{N-1} & 1\end{bmatrix}\begin{bmatrix}p_{11}(t)\\ p_{12}(t)\\ p_{22}(t)\end{bmatrix}$$

Its stability can be analyzed by **Routh criterion** applied to the characteristic polynomial, obtaining conditions on the parameter α.

$$\alpha < \bar{\alpha} = \frac{3}{2} - N + \frac{1}{2}\sqrt{4N^2 - 12N + 17}$$

or the more conservative but meaningful bound

$$\alpha \leq \frac{1}{N-1}$$

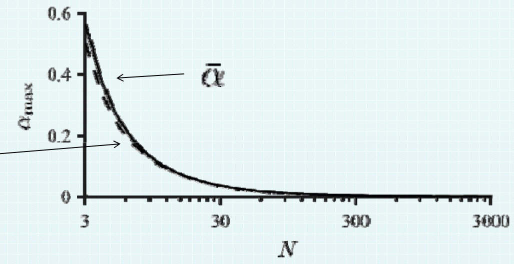
As linear stability guarantees convergence in any norm, and by Borell-Cantelli's lemma, **with probability 1** it will eventually hold

$$\left\|\begin{bmatrix}\bar{x}(t)\\ \bar{v}(t)\end{bmatrix}\right\|_2^2 \leq e^{-\eta t}$$

for some positive real number η .

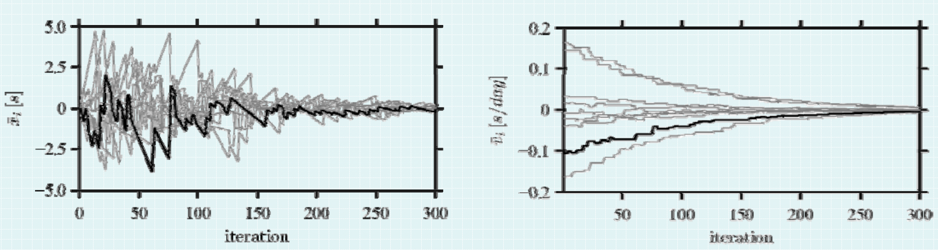
This analysis points out what is the main issue affecting this algorithm: **scalability. Indeed, ad the network size grows,** α **has to be smaller, and the algorithm convergence is slower.**

Parameter bound as a function of network size

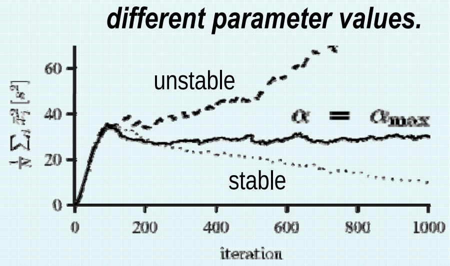


Simulations and remarks

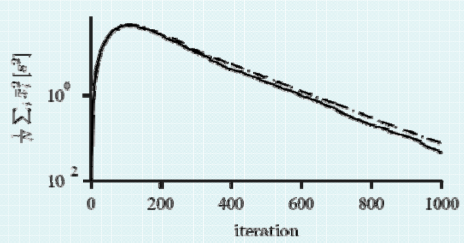
Clock offset and skew during a single run of the algorithm.



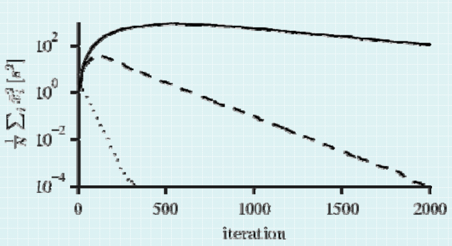
Squared clock errors for different parameter values.



Squared clock errors for different network sizes (50 nodes – continuous, 500 nodes – dashed, 5000 nodes – dotted) in the case of Multiple Symetric Gossip.

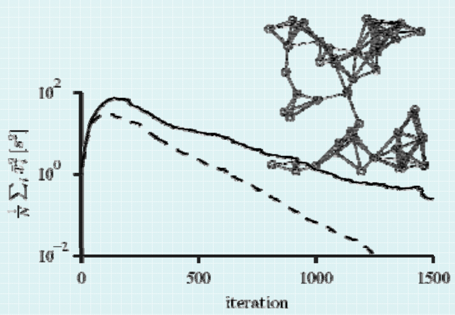


Simultations confirm that the behavior of the system is **qualitatively different** according to the value of α. Numerical simulations (below) confirm that the algorithm is **poorly scalable**, as convergence becomes **slower for larger networks.** The behavior of the algorithm on **random geometric graphs** is qualitatively consistent.



← Algorithm convergence for different network sizes, with $\alpha = 0.5\alpha_{max}$. *(10 nodes – dotted, 50 nodes – dashed, 250 nodes – continuous).*

→ Behavior of the algorithm on a random geometric graph (continuous line) compared to complete graph.



A solution to the scalability issue is investigated in the paper by simulation: **Multiple Symmetric Gossip.** At each time step, more than edge is activated (e.g. a fixed, small, portion of them, triggered by a local random variable).

This way two goal are achieved:

- the algorithm becomes **scalable**
- its implementation is **completely distributed.**