Self-Organizing Time Synchronization of WSNs with Adaptive Value Trackers

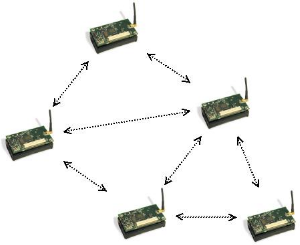
Önder GÜRCAN[1,2] & Kasım Sinan YILDIRIM[1]

[1]Department of Computer Engineering, Ege University, Turkey
[2]Institut de Recherche en Informatique de Toulouse, Paul Sabatier University, France

Seventh IEEE International Conference on Self-Adaptive and Self-Organizing Systems, September, 2013

Need for "Time" Synchronization in Wireless Sensor Networks (WSN)

Tiny sensor nodes with **limited** battery, memory, computation capability.



- **read-only hardware clock**
  - provides **local time notion**
  - frequently **drifts apart** due to aging, battery level, temperature etc.

- WSN applications such as **target tracking** require **global time notion**.

There is a need for *time synchronization* where...

- a *logical clock* value (representing the **global time**) is calculated

Why Self-Organization for Time Synchronization in WSNs?

- When frequent topological changes & node failures in WSNs are considered
  - **local** interactions (peer-to-peer)
  - **decentralized** control (no reference node)
  - **simple** behaviors (no hierarchical topology)
  - **global** organization (network-wide synchronization)
  - **dynamic** adaptivity (reaction to topological changes)

Why Self-Organization for Time Synchronization in WSNs?

- When frequent topological changes & node failures in WSNs are considered
  - **local** interactions (peer-to-peer)
  - **decentralized** control (no reference node)
  - **simple** behaviors (no hierarchical topology)
  - **global** organization (network-wide synchronization)
  - **dynamic** adaptivity (reaction to topological changes)

and these are...

...the properties of **self-organizing systems!** [Serugendo et al., 2011]

- **Several** self-organizing synchronization solutions in the literature! [Babaoglu et al., 2007, Tyrrell and Auer, 2008, Leidenfrost and Elmenreich, 2009, Tyrrell et al., 2010, Pagliari and Scaglione, 2011, Klinglmayr and Bettstetter, 2012, Zhang et al., 2012]
  - either **not designed** for WSNs
  - or provide **only synchronicity,** not **Global Time Notion = Sychronized "Time"**

Why we should want a new protocol?

- Besides, there are **practical** synchronization protocols
  - **a special node acting as a time reference** [Elson et al., 2002, van Greunen and Rabaey, 2003, Ganeriwal et al., 2003, Dai and Han, 2004, Maróti et al., 2004, Sun et al., 2006, Kusy et al., 2006, Lenzen et al., 2009, Schmid et al., 2009, Schmid et al., 2010, Ferrari et al., 2011, Yildirim and Kantarci, 2013b, Yildirim and Kantarci, 2013a]
  - **building a communication infrastructure** (e.g., **a spanning tree**) [van Greunen and Rabaey, 2003, Ganeriwal et al., 2003, Dai and Han, 2004, Sun et al., 2006]
  - **can only provide synchronicity** [Werner-Allen et al., 2005, Yu and Tirkkonen, 2008]
  - **keeping track of** the **time information** of **neighboring nodes** [Sommer and Wattenhofer, 2009, Schenato and Fiorentin, 2011]
    - causes **memory overhead**
    - which neighbors **to track**? which ones **to discard**?
    - a big problem especially in **densely connected** networks [Dousse and Thiran, 2004]

Why we should want a new protocol?

- Besides, there are **practical** synchronization protocols
  - **a special node acting as a time reference** [Elson et al., 2002, van Greunen and Rabaey, 2003, Ganeriwal et al., 2003, Dai and Han, 2004, Maróti et al., 2004, Sun et al., 2006, Kusy et al., 2006, Lenzen et al., 2009, Schmid et al., 2009, Schmid et al., 2010, Ferrari et al., 2011, Yildirim and Kantarci, 2013b, Yildirim and Kantarci, 2013a]
  - **building a communication infrastructure** (e.g., **a spanning tree**) [van Greunen and Rabaey, 2003, Ganeriwal et al., 2003, Dai and Han, 2004, Sun et al., 2006]
  - **can only provide synchronicity** [Werner-Allen et al., 2005, Yu and Tirkkonen, 2008]
  - **keeping track of** the **time information** of **neighboring nodes** [Sommer and Wattenhofer, 2009, Schenato and Fiorentin, 2011]
    - causes **memory overhead**
    - which neighbors **to track**? which ones **to discard**?
    - a big problem especially in **densely connected** networks [Dousse and Thiran, 2004]

What we want is to achieve ...

self-organizing time synchronization **without keeping track of neighbors**.

Self-Organizing Time Sychronization Protocol (STSP)

STSP Algorithm

- Periodically **broadcast the logical clock value to the neighbors (beacon period)**
- **Upon receiving the logical clock value of a neighbor**
  1. **calculate the difference (clock skew)** between my logical clock and the received value
  2. **add clock skew / 2** to my logical clock
  3. **if clock skew** is **lower than** the **max possible skew**
     1. **if clock skew > 0, speed up** my logical clock
     2. **if clock skew < 0, slow down** my logical clock
     3. **else the speed** of my logical clock is said to be **good**

But how this speed adjustment is done?

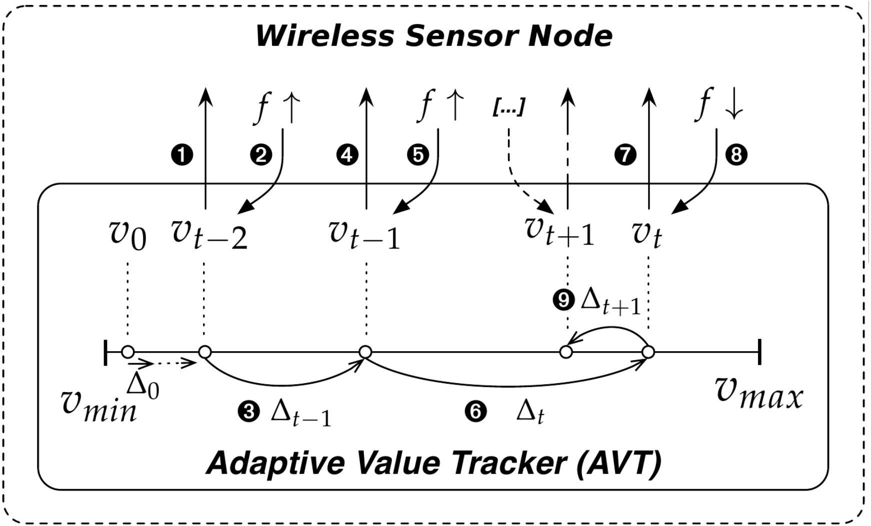


Figure: Let's begin with v_0 and try to find the best value between v_{min} and v_{max}.

STSP Simulation Setup

- **networks consisting of 100** sensor nodes
  - with **densities 10**, **20** and **50**
- **clock drifts** - uniformly distributed [-10^{-4}, 10^{-4}]
- **delays** on the communication links - gaussian random variable
- beacon period of **30** seconds.
- AVT params: $v_{min} = -10^{-4}, v_{max} = 10^{-4}$, $\Delta_{min} = 10^{-10}, \Delta_{max} = 10^{-5}$

STSP Simulation Setup

- **networks consisting of 100** sensor nodes
  - with **densities 10**, **20** and **50**
- **clock drifts** - uniformly distributed [-10^{-4}, 10^{-4}]
- **delays** on the communication links - gaussian random variable
- beacon period of **30** seconds.
- AVT params: $v_{min} = -10^{-4}, v_{max} = 10^{-4}, \Delta_{min} = 10^{-10}, \Delta_{max} = 10^{-5}$

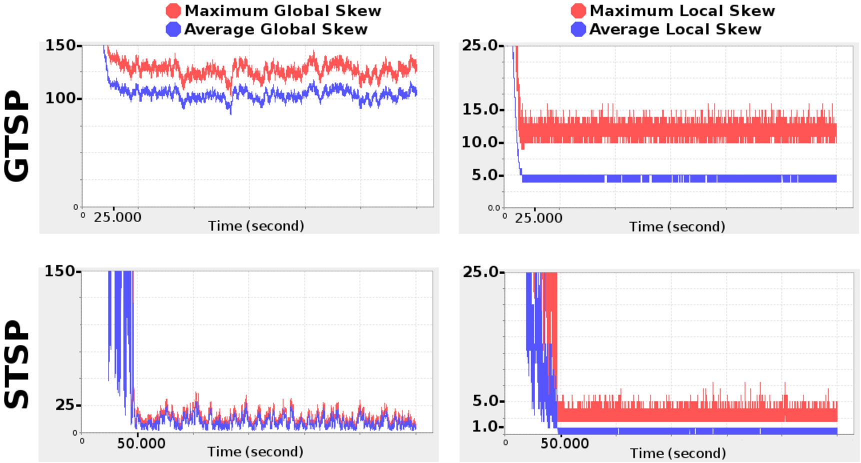
compared with Gradient Time Synchronization Protocol (GTSP) [Sommer and Wattenhofer, 2009]

- converges to the average clock value/speed of all neighbors
- allocates memory to keep track of each neighboring node
- lots of computation for each neighbor

Evaluation metrics: instantaneous synchronization errors

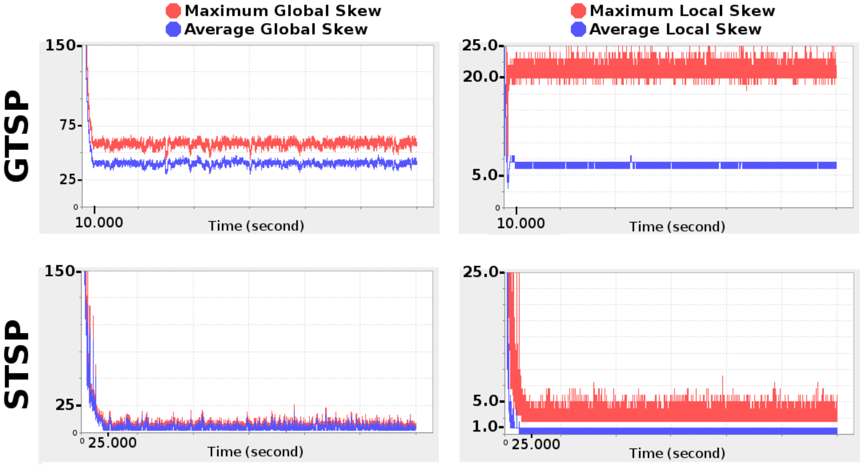
- global skew (arbitrary nodes)
- local skew (neighboring nodes)

Low Density Results - 100 nodes 10 neighbors



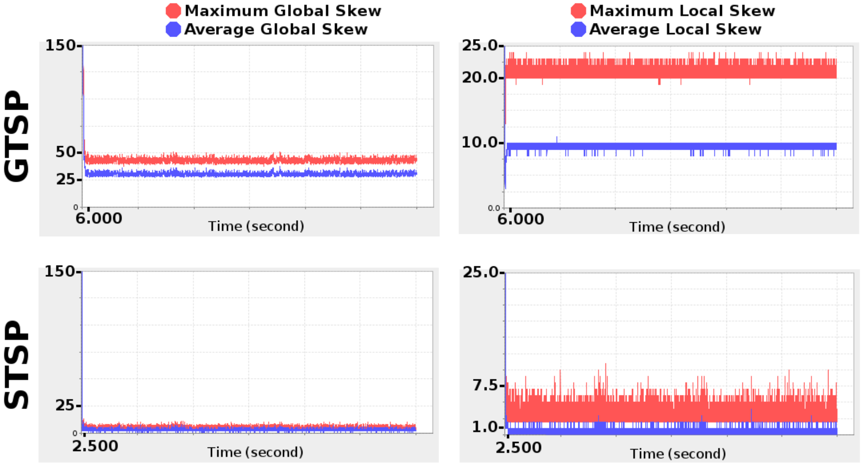
Tighter synchronization :-), worse convergence time :-(

Midium Density Results - 100 nodes 20 neighbors



Tighter synchronization :-), worse convergence time :-(

High Density Results - 100 nodes 50 neighbors



Tighter synchronization :-), better convergence speed :-)

Why STSP is meaningful for you?

- **Scalable**
  - the **denser** the network, the **tighter** the synchronization was
  - **memory & CPU requirements remain the same**.
- **Adaptive**, not **adaptable**.
  - **adaptive** - maintains **some** stable states (e.g., **any** global time value)
  - **adaptable** - maintains **particular** organization (e.g., a **specific** global time value)
- **Adaptivity** is provided using **Adaptive Value Trackers (AVTs)**
  - requires **quite a few arithmetic operations**
  - parameters to be set **carefully**, e.g. high precision (Δ_{min}) is **not good**

Why STSP is meaningful for you?

- **Scalable**
  - the **denser** the network, the **tighter** the synchronization was
  - **memory & CPU requirements remain the same**.
- **Adaptive**, not **adaptable**.
  - **adaptive** - maintains **some** stable states (e.g., **any** global time value)
  - **adaptable** - maintains **particular** organization (e.g., a **specific** global time value)
- **Adaptivity** is provided using **Adaptive Value Trackers (AVTs)**
  - requires **quite a few arithmetic operations**
  - parameters to be set **carefully**, e.g. high precision (Δ_{min}) is **not good**

Can AVTs improve robustness against faulty behavior?

- **Yes**. Because the Δ values converge Δ_{min}
- But **successive erroneous feedbacks** would increase the Δ values **exponentially**

References

Babaoglu, O., Binci, T., Jelasity, M., and Montresor, A. (2007).
Firefly-inspired heartbeat synchronization in overlay networks.
In *Self-Adaptive and Self-Organizing Systems, 2007. SASO '07. First International Conference on*, pages 77–86.

Dai, H. and Han, R. (2004).
Tsync: A lightweight bidirectional time synchronization service for wireless sensor networks.
ACM SIGMOBILE Mobile Computing and Communications Review, 8:125–139.

Dousse, O. and Thiran, P. (2004).
Connectivity vs capacity in dense ad hoc networks.
In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 476–486.

Elson, J., Girod, L., and Estrin, D. (2002).
Fine-grained network time synchronization using reference broadcasts.
SIGOPS Oper. Syst. Rev., 36(SI):147–163.

Ferrari, F., Zimmerling, M., Thiele, L., and Saukh, O. (2011).
Efficient network flooding and time synchronization with glossy.
In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 73–84. IEEE.

Ganeriwal, S., Kumar, R., and Srivastava, M. B. (2003).
Timing-sync protocol for sensor networks.
In *SenSys'03: Proc. of the 1st int. conf. on Embedded networked sensor systems*, pages 138–149, New York, NY, USA. ACM.

Klinglmayr, J. and Bettstetter, C. (2012).
Self-organizing synchronization with inhibitory-coupled oscillators: Convergence and robustness.
ACM Trans. Auton. Adapt. Syst., 7(3):30:1–30:23.

Kusy, B., Dutta, P., Levis, P., Maroti, M., Ledeczi, A., and Culler, D. (2006).
Elapsed time on arrival: A simple and versatile primitive for canonical time synchronisation services.
Int. J. Ad Hoc Ubiquitous Comput., 1(4):239–251.

Leidenfrost, R. and Elmenreich, W. (2009).
Firefly clock synchronization in an 802.15.4 wireless network.
EURASIP J. Embedded Syst., 2009:7:1–7:17.

Lenzen, C., Sommer, P., and Wattenhofer, R. (2009).
Optimal Clock Synchronization in Networks.
In *7th ACM Conference on Embedded Networked Sensor Systems (SenSys), Berkeley, California, USA*.

Maróti, M., Kusy, B., Simon, G., and Lédeczi, A. (2004).
The flooding time synchronization protocol.
In *SenSys'04: Proc. of the 2nd int. conf. on Embedded networked sensor systems*, pages 39–49, New York, NY, USA. ACM.

Pagliari, R. and Scaglione, A. (2011).
Scalable network synchronization with pulse-coupled oscillators.
Mobile Computing, IEEE Transactions on, 10(3):392–405.

Schenato, L. and Fiorentin, F. (2011).
Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks.

Automatica, 47(9):1878–1886.

Schmid, T., Charbiwala, Z., Anagnostopoulou, Z., Srivastava, M. B., and Dutta, P. (2010).
A case against routing-integrated time synchronization.
In *Proc. of the 8th ACM Conf. on Embedded Networked Sensor Systems*, SenSys '10, pages 267–280, New York, NY, USA. ACM.

Schmid, T., Charbiwala, Z., Shea, R., and Srivastava, M. (2009).
Temperature compensated time synchronization.
Embedded Systems Letters, IEEE, 1(2):37 –41.

Serugendo, G., Gleizes, M.-P., and Karageorgos, A., editors (2011).
Natural Computing Series. Springer.

Sommer, P. and Wattenhofer, R. (2009).
Gradient Clock Synchronization in Wireless Sensor Networks.
In *8th ACM/IEEE Inter. Conf. on Information Processing in Sensor Networks (IPSN), San Francisco, USA*.

Sun, K., Ning, P., and Wang, C. (2006).

Tinysersync: secure and resilient time synchronization in wireless sensor networks.
In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 264–277, New York, NY, USA. ACM.

Tyrrell, A. and Auer, G. (2008).
Decentralized inter-base station synchronization inspired from nature.
In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5.

Tyrrell, A., Auer, G., and Bettstetter, C. (2010).
Emergent slot synchronization in wireless networks.
Mobile Computing, IEEE Transactions on, 9(5):719–732.

van Greunen, J. and Rabaey, J. (2003).
Lightweight time synchronization for sensor networks.
In *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 11–19, New York, NY, USA. ACM.

Werner-Allen, G., Tewari, G., Patel, A., Welsh, M., and Nagpal, R. (2005).
Firefly-inspired sensor network synchronicity with realistic radio effects.
In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 142–153, New York, NY, USA. ACM.

Yildirim, K. S. and Kantarci, A. (2013a).
External gradient time synchronization in wireless sensor networks.
Parallel and Distributed Systems, IEEE Transactions on, 99(PrePrints):1.

Yildirim, K. S. and Kantarci, A. (2013b).
Time synchronization based on slow flooding in wireless sensor networks.
IEEE Transactions on Parallel and Distributed Systems, 99(PrePrints):1.

Yu, J. and Tirkkonen, O. (2008).
Self-organized synchronization in wireless network.

In *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, pages 329–338.

Zhang, H., Llorca, J., Davis, C. C., and Milner, S. D. (2012).
Nature-inspired self-organization, control, and optimization in heterogeneous wireless networks.
Mobile Computing, IEEE Transactions on, 11(7):1207–1222.