

# TIMESTAMP-FREE NETWORK SYNCHRONIZATION WITH RANDOM PAIRWISE MESSAGE EXCHANGES

*D. Richard Brown III and Andrew G. Klein*

Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609. Email: {drb,klein@wpi.edu}

## ABSTRACT

This paper describes a network synchronization technique based on bidirectional message exchanges between randomly chosen node pairs without the use of timestamps. Each node in the network has a clock with independent stochastic delay and drift. Relative delay and drift synchronization information is conveyed implicitly in the physical layer through the timing of responses to directed messages. Simultaneous delay and drift correction is performed by the node initiating the message exchange. Steady-state synchronization performance bounds are developed as a function of the synchronization stepsize parameter and the statistics of the delay and drift estimation errors. Numerical results for two different connectivity scenarios show that synchronization among the nodes can be achieved without the overhead of digital timestamps, a dedicated synchronization protocol, or network hierarchy.

**Index Terms**— synchronization, delay estimation, oscillator dynamics, ad hoc networks

## 1. INTRODUCTION

Clock synchronization is an important function in wireless networks and can facilitate scheduling of communication resources, interference avoidance, event detection/ordering, data fusion, and coordinated wake/sleep cycles [1]. A variety of synchronization protocols and systems have been developed for wired and wireless communication networks over the last 30 years, e.g., [1–11]. A common theme running through most of these protocols and systems is that they are based on a dedicated protocol with application-layer or MAC-layer exchanges of digital timestamps between nodes in the network. Digital timestamps inherently have a minimum resolution, limiting accuracy, and (along with the use of a dedicated synchronization protocol) add overhead to the network traffic. Hence, it is of interest to study synchronization techniques that do not require digital timestamps or a dedicated protocol.

Timestamp-free synchronization techniques were originally studied in the context of natural phenomena and led to formal mathematical models for systems of pulse-coupled oscillators in [12, 13] and application of these models to wireless sensor networks in [14–16]. While these studies represented a paradigm shift with respect to much of the prior work, a limitation of the pulse-coupled oscillator literature is that it is based on unidirectional transmissions and assumes negligible propagation delays. This inherently limits the synchronization accuracy of these methods.

Timestamp-free synchronization techniques accounting for propagation delays were developed in the context of retrodirective distributed transmit beamforming systems [17–21]. These

techniques are generally based on serial exchanges of unmodulated beacons through a network with a ring architecture and achieve precise synchronization through estimation of RF carrier phase and frequency offsets. A master-slave timestamp-free synchronization protocol was also developed in [22]. This technique was based on bidirectional message exchanges between a slave and the master node with synchronization information conveyed implicitly in the timing of the master node’s response. While all of these techniques avoid the use of timestamps, the use of a dedicated synchronization protocol and the establishment of a specific network architecture in each case leads to undesirable overhead.

This paper extends the work in [22] by generalizing the idea of timestamp-free synchronization from master-slave networks to ad hoc networks with no network hierarchy. Bidirectional message exchanges between each transmitter/receiver pair are assumed to occur randomly in the network in a fashion similar to “random asymmetric gossip” [23–26]. In these papers, the convergence analysis and numerical results assume ideal clocks with no stochastics, no estimation error, and separate drift and delay compensation. In this paper, we extend [22, 26] by developing a timestamp-free synchronization protocol that performs simultaneous drift/delay compensation, requires no network hierarchy, and accounts for clock stochastics and estimation error. As in [22], our approach is to convey implicit timing information *in the physical layer* through the timing of the message responses from the receiver to the transmitter. Since no timestamps are exchanged and messages occur randomly from the perspective of the synchronization system, synchronization functions can be embedded in existing network traffic. Steady-state synchronization performance bounds are developed as a function of the synchronization stepsize parameter and the statistics of the delay and drift estimation errors. Numerical results for two different connectivity scenarios show that synchronization among the nodes can be achieved without the overhead of digital timestamps, a dedicated synchronization protocol, or network hierarchy.

## 2. SYSTEM MODEL

We assume a time-slotted time-division duplexed (TDD) network of  $N$  nodes and denote the propagation delay from node  $i$  to node  $j$  as  $\tau^{(i,j)}$ . The duration of each timeslot is denoted as  $T$  and the timeslot index is denoted as  $k$ . In each timeslot, a bidirectional message exchange occurs between a single transmit/receive pair. Since all of the channels in the system are TDD, we assume reciprocal propagation delays  $\tau^{(i,j)} = \tau^{(j,i)}$  in each link.

### 2.1. Probabilistic Pairwise Messaging

Since we assume no dedicated synchronization protocol, we specify a transmit/receive probability matrix  $\mathbf{P}[k]$  with  $i, j^{\text{th}}$  entry  $p_{i,j}[k]$

This work was supported by the National Science Foundation awards CCF-1302104 and CCF-1319458.

corresponding to the probability that node  $i$  initiates a message exchange with node  $j$  in timeslot  $k$ . We assume throughout this paper that  $\mathbf{P}[k] \equiv \mathbf{P}$ , i.e., the transmit/receive probabilities are constant. The case  $p_{i,j} = p_{j,i} = 0$  corresponds to the situation where node  $i$  and node  $j$  do not communicate. Note that  $p_{i,i} = 0$  for all  $i = 1, \dots, N$  and  $\sum_i \sum_j p_{i,j} = 1$ . We do not necessarily assume  $p_{i,j} = p_{j,i}$ ; for example,  $p_{i,j} > 0$  and  $p_{j,i} = 0$  corresponds to the case where  $i$  initiates message exchanges with node  $j$  but node  $j$  never initiates message exchanges with node  $i$ .

## 2.2. Local Clock Dynamics

The nodes in the network do not possess a common notion of time. We will use the notation  $t$  to refer to a reference time, i.e. the “true” time, in the system. All time-based quantities such as propagation delays and/or frequencies are specified in reference time unless otherwise noted.

None of the nodes have knowledge of the reference time  $t$ . Based on the two-state models in [27], we define the discrete-time state of the  $i^{\text{th}}$  node’s clock at time  $t = kT$  as

$$\mathbf{x}_i[k] = [\Delta_i[k], \dot{\Delta}_i[k]]^\top$$

where  $\Delta_i[k]$  corresponds to the clock delay at node  $i$  with respect to the reference time. The state update of the  $i^{\text{th}}$  transmit node’s clock is governed by

$$\mathbf{x}_i[k+1] = \mathbf{F}(T)\mathbf{x}_i[k] + \mathbf{u}_i[k] + \mathbf{c}_i[k] \quad (1)$$

with state update matrix

$$\mathbf{F}(T) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad (2)$$

process noise vector  $\mathbf{u}_i[k] \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{Q}_i(T))$ , and local clock correction vector  $\mathbf{c}_i[k]$  as described in Section 3. The process noise vector captures the effect of stochastic phase noise in the  $i^{\text{th}}$  node’s local oscillator. The covariance of the discrete-time process noise is derived from a continuous-time model in [27] and is given as

$$\mathbf{Q}_i(T) = T \begin{bmatrix} p_i + q_i \frac{T^2}{3} & q_i \frac{T}{2} \\ q_i \frac{T}{2} & q_i \end{bmatrix} \quad (3)$$

where  $p_i$  (units of seconds) and  $q_i$  (units of Hertz) are the process noise parameters corresponding to white frequency noise and random walk frequency noise, respectively. The process noise parameters  $p_i$  and  $q_i$  can be estimated by fitting the theoretical Allan variance  $\sigma_y^2(\tau) = p_i/\tau + q_i\tau/3$  to experimental measurements of the Allan variance over a range of  $\tau$  values. For notational convenience hereafter we will assume all of the oscillators to have the same process noise parameters such that  $p_i = p$  and  $q_i = q$  for all  $i = 1, \dots, N$ .

## 2.3. Time Translation

To express the time of events in different timebases, we assume the timeslot duration  $T$  is sufficiently short so that the local time at node  $i$  can be written as

$$t_i = t + \Delta_i[k] + (t - kT)\dot{\Delta}_i[k] \quad t \in [kT, (k+1)T). \quad (4)$$

Given an interval  $[t^{(1)}, t^{(2)}] \subseteq [kT, (k+1)T]$  with duration  $\tau = t^{(2)} - t^{(1)}$  specified in reference time, we can use (4) to express the duration of the interval in local time at node  $i$  as

$$\tau_i = (1 + \dot{\Delta}_i[k])\tau. \quad (5)$$

Solving for  $t$  in (4), the local time at node  $j$  can also be related to the local time at node  $i$  as

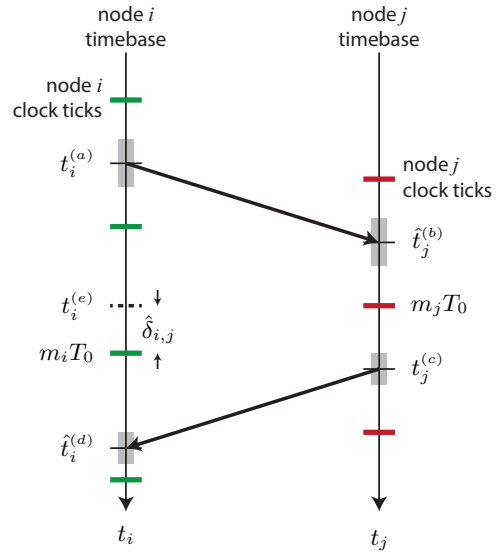
$$\frac{t_i - \Delta_i[k] + kT\dot{\Delta}_i[k]}{1 + \dot{\Delta}_i[k]} = \frac{t_j - \Delta_j[k] + kT\dot{\Delta}_j[k]}{1 + \dot{\Delta}_j[k]}. \quad (6)$$

This last expression implies an invertible mapping such that  $t_j = f_{i,j}(t_i)$  and  $t_i = f_{j,i}(t_j)$ .

## 3. TIMESTAMP-FREE NETWORK SYNCHRONIZATION

This section describes a time-slotted timestamp-free network synchronization protocol that allows each node in the network to arrive at a common clock drift and clock delay through pairwise message exchanges with random neighbors. The goal is not to force the drifts/delays to zero. Rather, the goal is to drive the clock drifts and delays to a *common* value across the network so that the *relative* drifts and delays between any pair of nodes are small. The protocol described here is similar the timestamp-free protocol in [22] with the main difference being that that the protocol described here is non-hierarchical and does not require a master/slave configuration.

Fig. 1 shows the interactions between a pair of nodes denoted as node  $i$  and node  $j$ . In timeslot  $k$ , these nodes are randomly selected with probability specified by  $\mathbf{P}$ . We assume the timeslots are sufficiently short so that the clock states  $\mathbf{x}_i[k]$  and  $\mathbf{x}_j[k]$  are approximately constant.



**Fig. 1.** Timestamp-free synchronization bidirectional message exchange in timeslot  $k$ .

The protocol begins with node  $i$  transmitting a packet to node  $j$  at arbitrary local time  $t_i^{(a)}$ . We use the convention that a packet transmission or reception time corresponds to the time at the center of the packet as shown in Fig. 1. The packet arrives at node  $j$  at local time

$$t_j^{(b)} = f_{i,j}(t_i^{(a)}) + (1 + \dot{\Delta}_j[k])\tau^{(i,j)}$$

where  $\tau^{(i,j)}$  is the propagation delay from node  $i$  to node  $j$  and the time-translation function  $f_{i,j}$  from node  $i$ ’s timebase to node  $j$ ’s timebase is implicit in (6). Node  $j$ ’s estimate of the time of arrival (in node  $j$ ’s timebase) is then

$$\hat{t}_j^{(b)} = t_j^{(b)} + \epsilon^{(b)}$$

where  $\epsilon^{(b)} \sim \mathcal{N}(0, R_1)$  is the time-of-arrival estimation error. Node  $j$  then transmits a reply back to node  $i$  at time  $t_j^{(c)}$  where  $t_j^{(c)}$  is selected such that

$$\frac{\hat{t}_j^{(b)} + t_j^{(c)}}{2} = m_j T_0 \quad (7)$$

where  $T_0$  is the nominal clock tick period (typically a large multiple of the fundamental period of the local oscillator and a fraction of timeslot period  $T$ ) and  $m_j$  is an integer such that  $m_j T_0 > \hat{t}_j^{(b)}$ . Note that all quantities in (7) are in node  $j$ 's timebase. Also note that, in the example shown in Fig. 1, the first clock tick after  $\hat{t}_j^{(b)}$  is used as the center point between  $\hat{t}_j^{(b)}$  and  $t_j^{(c)}$ , but this is not required. Any clock tick after  $\hat{t}_j^{(b)}$  can be used as the center point.

Since the channel is reciprocal, node  $i$  receives the reply packet from node  $j$  at local time

$$t_i^{(d)} = f_{j,i}(t_j^{(c)}) + (1 + \dot{\Delta}_i[k])\tau^{(i,j)}.$$

Node  $i$ 's estimate of the time of arrival (in node  $i$ 's timebase) is then

$$\hat{t}_i^{(d)} = t_i^{(d)} + \epsilon^{(d)}$$

where  $\epsilon^{(d)} \sim \mathcal{N}(0, R_1)$  is the time-of-arrival estimation error and is assumed to be independent of  $\epsilon^{(b)}$ . Using only local times, node  $i$  can then calculate the midpoint

$$t_i^{(e)} = \frac{t_i^{(a)} + \hat{t}_i^{(d)}}{2}$$

and then compute the modulo relative clock offset

$$\hat{\delta}_{i,j} = m_i T_0 - t_i^{(e)}$$

where  $m_i$  is the time of the local clock tick nearest to  $t_i^{(e)}$ . Note that  $\hat{\delta}_{i,j}$  serves as an estimate of the modulo relative clock offset at node  $i$  with respect to node  $j$ , where a positive (negative) value indicates that node  $i$ 's clock is lagging (leading) node  $j$ 's clock.

Assuming each node derives its carrier and/or symbol rate from the same oscillator driving its local clock, node  $i$  can also estimate its *relative clock drift* with respect to node  $j$  via carrier frequency offset and/or symbol rate offset present in the reply packet from node  $j$ . Denoting the relative clock drift estimate as

$$\hat{\nu}_{i,j}[k] = \hat{\Delta}_j[k] - \hat{\Delta}_i[k] + \eta[k]$$

where  $\eta[k] \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, R_2)$  is the relative drift estimation error, assumed to be independent of the time of arrival estimation errors, node  $i$  then computes a clock correction vector as

$$\mathbf{c}_i[k] = \mu \begin{bmatrix} \hat{\delta}_{i,j}[k] \\ \hat{\nu}_{i,j}[k] \end{bmatrix} \quad (8)$$

where  $\mu > 0$  is a stepsize that provides a tradeoff between convergence speed and steady-state performance. Convergence conditions on  $\mu$  for separate drift/delay compensation in the absence of estimation error are provided in [26].

In timeslot  $k$ , only node  $i$  and node  $j$  exchange messages. Only node  $i$  performs clock correction based on the timing of node  $j$  response. The clock correction vectors  $\mathbf{c}_\ell[k] = [0, 0]^\top$  for all  $\ell \neq i$ . In subsequent timeslots, different  $i, j$  pairs are selected according to the probabilistic pairwise messaging matrix  $\mathbf{P}$  and the process is repeated with only one node performing clock correction in each timeslot.

#### 4. STEADY-STATE PERFORMANCE ANALYSIS

In this section, we develop a bound on the achievable performance of the time-stamp free synchronization protocol under the simplifying assumptions that (i) there are only  $N = 2$  nodes, (ii) node 1 always initiates bidirectional message exchanges with node 2, (iii) the clocks of nodes 1 and 2 are sufficiently close so that modulo effects can be ignored and (iv) oscillator stochastics are negligible ( $\mathbf{u}_1[k] = \mathbf{u}_2[k] = [0, 0]^\top$  for all  $k$ ).

Under these assumptions, the clock correction vectors

$$\mathbf{c}_1[k] = \mu(\mathbf{x}_2[k] - \mathbf{x}_1[k] + \boldsymbol{\epsilon}[k]), \quad (9)$$

and  $\mathbf{c}_2[k] = [0, 0]^\top$  for all  $k$ , hence we can use (1) to write the relative clock offset and drift as

$$\mathbf{x}_1[k+1] - \mathbf{x}_2[k+1] = \mathbf{F}(T)(\mathbf{x}_1[k] - \mathbf{x}_2[k]) + \mathbf{c}_1[k].$$

Denoting  $\mathbf{z}[k] = \mathbf{x}_1[k] - \mathbf{x}_2[k]$  and using (9), we can express the dynamics of the relative clock offset and drift between nodes 1 and 2 as

$$\mathbf{z}[k+1] = (\mathbf{F}(T) - \mu\mathbf{I})\mathbf{z}[k] + \mu\boldsymbol{\epsilon}[k]$$

If  $\mu$  is small enough so that a steady-state solution is achieved, we have the steady-state relative offset and drift covariance

$$\mathbb{E}[\mathbf{z}[k+1]\mathbf{z}^\top[k+1]] = \mathbb{E}[\mathbf{z}[k]\mathbf{z}^\top[k]] = \mathbf{S}$$

and

$$\mathbf{S} = (\mathbf{F}(T) - \mu\mathbf{I})\mathbf{S}(\mathbf{F}(T) - \mu\mathbf{I})^\top + \mu^2\mathbf{R} \quad (10)$$

where  $\mathbf{R} = \mathbb{E}[\boldsymbol{\epsilon}[k]\boldsymbol{\epsilon}^\top[k]]$  is the covariance of the delay and drift measurement errors. Under the measurement error assumptions in Section 3, it can be shown that  $\mathbf{R} = \text{diag}(R_1/2, R_2)$ . Since (10) is a  $2 \times 2$  discrete-time algebraic Riccati equation, a closed-form solution for  $\mathbf{S}$  can be straightforwardly calculated as

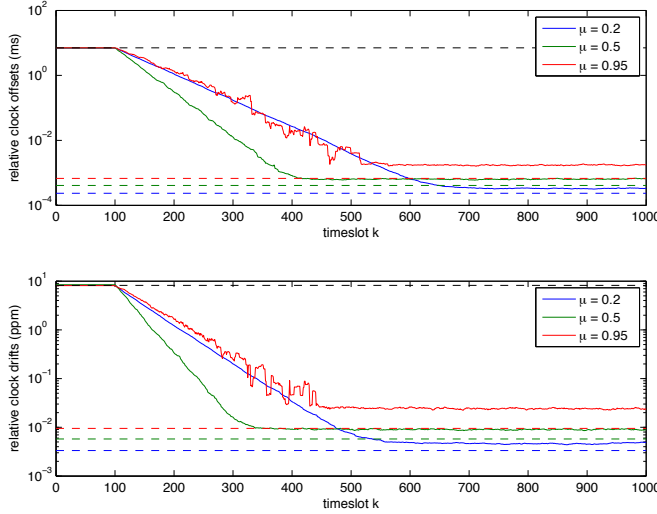
$$\mathbf{S} = \begin{bmatrix} \frac{2-2\mu+\mu^2}{(2-\mu)^3\mu} T^2 R_2 + \frac{\mu R_1}{2(2-\mu)} & \frac{(1-\mu)TR_2}{(2-\mu)^2} \\ \frac{(1-\mu)TR_2}{(2-\mu)^2} & \frac{\mu R_2}{2-\mu} \end{bmatrix}. \quad (11)$$

The following section shows these steady-state synchronization performance results, although developed for the  $N = 2$  case under ideal clock assumptions, can be good predictors of the steady-state performance for general values of  $N$  and with stochastic clocks.

#### 5. NUMERICAL RESULTS

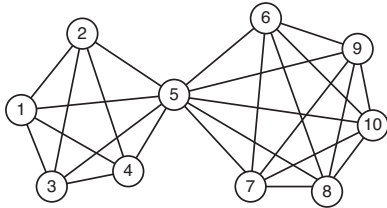
The numerical results in this section assume a network with  $N = 10$  nodes. The initial clock delays  $\Delta_i[0]$  were generated as i.i.d. zero-mean Gaussian distributed random variables with standard deviation 5 ms, and the initial drifts  $\dot{\Delta}_i[0]$  were generated as i.i.d. uniformly distributed random variables on  $\pm 10$  parts-per-million (ppm) for  $i = 1, \dots, N$ . The stochastic oscillator parameters  $p$  and  $q$  were derived via a least-squares fit from a table of typical Allan variance characteristics of a "good" crystal oscillator in [28]. The nominal clock period  $T_0$  was set to 100 ms and the messaging period  $T$  was set to 250 ms. The standard deviation of each delay estimate was set to 1  $\mu\text{s}$  and the standard deviation of each relative drift estimate was set to 0.01 ppm, i.e.,  $R_1 = 10^{-12}$  and  $R_2 = 10^{-16}$ . Delay and drift estimation errors were assumed to be independent.

Fig. 2 shows the average behavior of the timestamp-free synchronization protocol in a fully-connected network with equiprobable transmit/receive pairs in terms of the standard deviation of the clock offsets and drifts relative to node 10 for nodes 1,  $\dots$ , 9. The results are averaged over 1000 realizations of the protocol for each value of  $\mu$ . In the first 100 timeslots, no clock corrections are performed. Clock corrections are performed according to the protocol in Section 3 in all of the subsequent timeslots. These results show that the relative drifts and offsets converge to values close to the lower bounds derived in Section 4 and that better steady-state performance is achieved in this case with smaller values of  $\mu$ .



**Fig. 2.** Standard deviation of relative clock offsets and drifts with equiprobable transmit/receive pairs. The colored dashed lines are from (10). The black dashed lines are the standard deviations of the initial conditions.

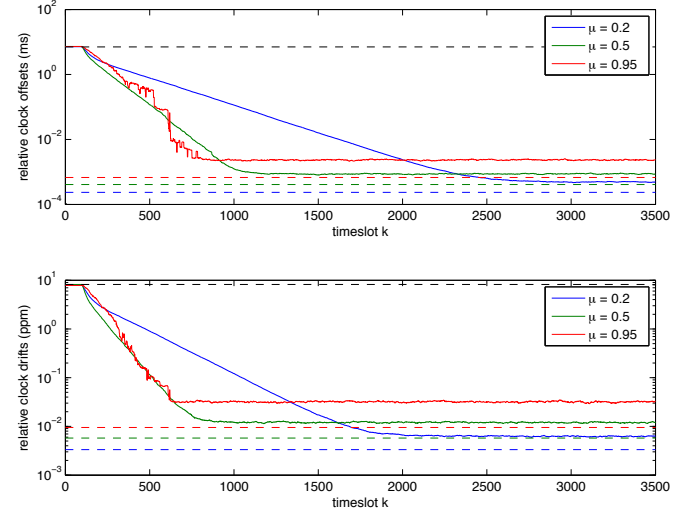
We now consider the partitioned network with a gateway node shown in Fig. 3. In this example, the  $N = 10$  node network is partitioned into sets  $\mathcal{S}_1 = \{1, 2, 3, 4, 5\}$  and  $\mathcal{S}_2 = \{5, 6, 7, 8, 9, 10\}$  where communication between members of the same set has probability  $p_{i,j} = \frac{1}{50}$  and communication between members of different sets has probability  $p_{i,j} = 0$ .



**Fig. 3.**  $N = 10$  partitioned network topology with node 5 serving as the gateway node. Each edge in the graph corresponds to a transmit/receive pair with probability  $p = \frac{1}{50}$  in either direction.

Fig. 4 shows the average behavior of the timestamp-free synchronization protocol in the partitioned network with a gateway node shown in Fig. 3 in terms of the standard deviation of the clock offsets and drifts relative to node 10 for nodes 1,  $\dots$ , 9. Other than the message probabilities and the number of timeslots, the simulation

parameters were otherwise identical to those in Fig. 2. These results show that, although partitioned network tends to synchronize slower than the fully-connected network, the relative drifts and offsets still converge to values close to the lower bounds derived in Section 4.



**Fig. 4.** Standard deviation of relative clock offset for the partitioned network with a gateway node shown in Fig. 3. Note the change of axis limits with respect to Fig. 2. The colored dashed lines are from (10). The black dashed lines are the standard deviations of the initial conditions.

## 6. CONCLUSIONS

This paper describes a new approach to network synchronization that performs simultaneous delay and drift compensation, requires no network hierarchy, and exploits the physical layer properties of existing bidirectional messages in a wireless network to achieve synchronization without the exchange of timestamps. Since no timestamps are exchanged, synchronization functions can be embedded in existing network traffic. Also, since the protocol is based on bidirectional message exchanges, it accounts for propagation delay and its accuracy is only limited by the stochastic properties of the oscillators and the accuracy of the delay and drift estimates. Steady-state synchronization performance bounds are developed as a function of the synchronization stepsize parameter and the statistics of the delay and drift estimation errors. Numerical results for two different connectivity scenarios show that synchronization among the nodes can be achieved without the overhead of digital timestamps, a dedicated synchronization protocol, or network hierarchy.

## 7. REFERENCES

- [1] M. Maggs, S. O'Keefe, and D. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2269–2277, Jun. 2012.
- [2] D. Mills, "Internet time synchronization: the network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.
- [3] J. Hahn and E. Powers, "Implementation of the GPS to Galileo time offset (GGTO)," in *Proc. of the 2005 Joint IEEE International Frequency Control Symposium and Precise Time and*

*Time Interval (PTTI) Systems and Applications Meeting*, Aug. 2005, pp. 33–212.

- [4] S. Ganeriwal, R. Kumar, and M. Srivastava, “Timing-sync protocol for sensor networks,” in *Proceedings ACM SenSys 2003*. ACM New York, NY, USA, Nov. 2003, pp. 138–149.
- [5] J. van Greunen and J. Rabaey, “Lightweight time synchronization for sensor networks,” in *Proc. 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA 2003)*, Sep. 2003.
- [6] M. Sichitiu and C. Veerarittiphan, “Simple, accurate time synchronization for wireless sensor networks,” in *Proc. IEEE Wireless Communications and Networking (WCNC’03)*, vol. 2, Mar. 2003, pp. 1266–1273.
- [7] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, “The flooding time synchronization protocol,” in *Proc. 2nd Intl. Conf. on Embedded Networked Sensor Systems*, Nov. 2004.
- [8] W. Su and I. Akyildiz, “Time-diffusion synchronization protocol for wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 384–397, Apr. 2005.
- [9] Q. Li and D. Rus, “Global clock synchronization in sensor networks,” *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–226, Feb. 2006.
- [10] R. Solis, V. Borkar, and P. Kumar, “A new distributed time synchronization protocol for multihop wireless networks,” in *IEEE Conf. on Decision and Control*, Dec. 2006, pp. 2734–2739.
- [11] L. Schenato and G. Gamba, “A distributed consensus protocol for clock synchronization in wireless sensor network,” in *IEEE Conf. on Decision and Control*, Dec. 2007, pp. 2289–2294.
- [12] R. Mirollo and S. Strogatz, “Synchronization of pulse-coupled biological oscillators,” *SIAM J. on Appl. Math.*, pp. 1645–1662, 1990.
- [13] D. Lucarelli and I. Wang, “Decentralized synchronization protocols with nearest neighbor communication,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems (ACM SenSys’04)*, November 3–5 2004, pp. 62–68.
- [14] Y.-W. Hong and A. Scaglione, “A scalable synchronization protocol for large scale sensor networks and its applications,” *IEEE Journal on Selected Areas in Comm.*, vol. 23, no. 5, pp. 1085–1099, May 2005.
- [15] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, “Firefly-inspired sensor network synchronicity with realistic radio effects,” in *Proc. of the 3rd international conf. on Embedded networked sensor systems (ACM SenSys’05)*, November 2–4 2005, pp. 142–153.
- [16] A.-S. Hu and S. Servetto, “On the scalability of cooperative time synchronization in pulse-connected networks,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2725–2748, Jun. 2006.
- [17] D.R. Brown III, G. Prince, and J. McNeill, “A method for carrier frequency and phase synchronization of two autonomous cooperative transmitters,” in *Proc. of the 6th IEEE Signal Processing Advances in Wireless Comm.*, New York, NY, June 5–8 2005, pp. 260–264.
- [18] I. Ozil and D.R. Brown III, “Time-slotted round-trip carrier synchronization,” in *Proceedings of the 41st Asilomar Conf. on Signals, Systems, and Comp.*, Pacific Grove, CA, Nov. 4–7 2007, pp. 1781–1785.
- [19] D.R. Brown III and H.V. Poor, “Time-slotted round-trip carrier synchronization for distributed beamforming,” *IEEE Trans. on Signal Processing*, vol. 56, no. 11, pp. 5630–5643, November 2008.
- [20] R. Preuss and D.R. Brown III, “Retrodirective distributed transmit beamforming with two-way source synchronization,” in *Proceedings of the Conference of Information Sciences and Systems (CISS 2010)*, Princeton, NJ, March 2010, pp. 1–6.
- [21] —, “Two-way synchronization for coordinated multi-cell retrodirective downlink beamforming,” *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5415–5427, Nov. 2011.
- [22] D.R. Brown III and A. Klein, “Precise timestamp-free network synchronization,” in *Conf. Inf. Sciences and Systems (CISS2013)*, Mar. 2013.
- [23] F. Fagnani and S. Zampieri, “Asymmetric randomized gossip algorithms for consensus,” in *Proc. IFAC world congress*, 2008, pp. 9051–9056.
- [24] —, “Randomized consensus algorithms over large scale networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 634–649, 2008.
- [25] R. Carli, E. D’Elia, and S. Zampieri, “A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 7512–7517.
- [26] D.R. Brown III, A. Klein, and R. Wang, “Convergence conditions for consensus synchronization in wireless networks,” *IEEE Transactions on Signal Processing*, In review. tbd.
- [27] L. Galleani, “A tutorial on the 2-state model of the atomic clock noise,” *Metrologia*, vol. 45, no. 6, pp. S175–S182, Dec. 2008.
- [28] W. Klepczynski and P. Ward, “Frequency stability requirements for narrow band receivers,” in *32nd Annual Precise Time and Time Interval Meeting*, November 2000.