Distributed Time Synchronization for Networks with Random Delays and Measurement Noise

Miloš S. Stanković, Srdjan S. Stanković and Karl Henrik Johansson

May 31, 2017

Abstract

In this paper new distributed asynchronous algorithms are proposed for time synchronization in networks with random communication delays, measurement noise and communication dropouts. Three algorithms are proposed for drift correction, based on different current local time increments. Under nonrestrictive conditions concerning network properties, it is proved that all the algorithms provide convergence in the mean square sense and with probability one (w.p.1) of the corrected drifts of all the nodes to the same value (consensus). An estimate of the convergence rate of these algorithms is derived. It is shown that it is possible to achieve convergence to the common virtual clock. For offset correction, a new algorithm is proposed based on local time differences, containing a compensation parameter coping with the influence of random delays, and with additional special terms taking care of the influence of linearly increasing time and drift correction. It is proved that the corrected offsets of all the nodes converge in the mean square sense and w.p.1. An efficient algorithm based on consensus on compensation parameters is also proposed. It is shown that all the
proposed algorithms can be implemented as flooding algorithms with one reference node. Simulation results provide an additional practical insight into the properties of the algorithms.

1 Introduction

Cyber-Physical Systems (CPS), Internet of Things (IoT) and Sensor Networks (SN) have emerged as research areas of paramount importance with many conceptual and practical challenges and numerous applications (Kim and Kumar, 2012; Holler et al., 2014; Akyildiz and Vuran, 2010). One of the basic requirements in networked systems, in general, is time synchronization, i.e., necessity for all the nodes to share a common notion of time. The problem of time synchronization has attracted a lot of attention, but still represents a challenge due to multi-hop communications, stochastic delays, communication and measurement noise, unpredictable packet losses and high probability of node failures, e.g., (Sundaraman et al., 2005). There are numerous approaches to time synchronization starting from different assumptions and using different methodologies, e.g., (Elson et al., 2002; Sundaraman et al., 2005; Sivrikaya and Yener, 2004). An important class of time synchronization algorithms is based on full distribution of functions (Simeone et al., 2008; Solis et al., 2006). Distributed schemes with the so-called gradient property have been proposed (Sommer and Wattenhofer, 2009; Fan and Lynch, 2006). A class of consensus based algorithms, called CBTS (Consensus-Based Time Synchronization) algorithms has attracted considerable attention, e.g., (Li and Rus, 2006; Xiong and Kishore, 2009; He et al., 2014b; He et al., 2014a; Schenato and Fiorentin, 2011; Liao and Barooah, 2013a; Tian, 2015). It has been treated in a unified way in a recent survey (Tian et al., 2016), providing figure of merit of the principal approaches. In (Carli et al., 2008; Yildirim et al., 2015) a control-based approach to distributed time synchronization has been adopted. Fundamental and yet unsolved problems in all time synchronization approaches are connected with communication delays and measurement noise; see (Freris et al., 2011) for basic issues, and (Chaudhari et al., 2008; Xiong and Kishore, 2009; Choi et al., 2012; Garone et al., 2015) for different aspects of delay influence.

In this paper we propose a new asynchronous distributed algorithm for time synchronization in lossy networks characterized by random communication delays, measurement noise and communication dropouts. The algorithm is composed of two distributed recursions of asynchronous stochastic approximation type based on broadcast gossip and derived from predefined
local error functions. The recursions are aimed at achieving asymptotic consensus on the corrected drifts and corrected offsets and, consequently, at obtaining common virtual clock for all the nodes in the network.

The proposed recursion for drift synchronization (presented in a preliminary form in (Stanković et al., 2016)) is based on noisy time increments defined in three characteristic forms. We prove convergence to consensus of the corrected drifts in the mean square sense and with probability one (w.p.1), under nonrestrictive conditions. Furthermore, we provide an estimate of the corresponding asymptotic convergence rate to consensus, essential for utilization within offset estimation and for achieving convergence to a common global virtual clock. Compared to the existing analogous algorithms (Schenato and Fiorentin, 2011; Tian, 2015), the proposed scheme is structurally different and simpler (not involving mutual drift estimation typical for the CBTS algorithms) and provides the best performance. Notice that the algorithm proposed in (Schenato and Fiorentin, 2011) cannot handle communication delays or measurement noise, while the paper (Tian, 2015) treats random delays, but not the case of measurement noise and communication dropouts. Moreover, the algorithm proposed therein cannot provide convergence rate achievable by the proposed methodology. The approach in (Garone et al., 2015) does not ensure consensus of corrected drifts in spite of additional pairwise inter-node communications.

The proposed recursion for offset synchronization is constructed starting from a specific error function derived from the difference between local times. Two important refinements are introduced aiming at eliminating the effect of linearly increasing absolute time and coping with the influence of delays by introducing additional delay compensation parameters. It is proved that the algorithm provides convergence in the mean square sense and w.p.1 to a set of finite random variables. The algorithm for the offset correction proposed in (Schenato and Fiorentin, 2011) cannot handle these problems, while the algorithm in (Tian, 2015) is proposed only for drift correction assuming perfect clock readings. The approach in (Yildirim et al., 2015) does not provide a rigorous insight into overall network stability. Attention is also paid to an improvement of the offset correction algorithm, based on the introduction of linear consensus iterations in the recursion for the delay compensating parameters, aiming at decreasing the dispersion of the offset convergence points of different nodes. This modification can be a simple and efficient tool in practice. Special cases related to delay and noise are discussed in order to clarify potentials of the proposed algorithms.

Both drift and offset correction algorithms ensure finite differences between local virtual clocks in the mean square sense and w.p.1. To the
authors' knowledge, the proposed algorithm represents the first method with such a performance in the case of random delays, measurement noise and communication dropouts.

It is also demonstrated that the proposed algorithms can be implemented as flooding algorithms, with one predefined reference node.

Finally, some illustrative simulation results are presented, giving an additional insight into the theoretically discussed issues.

2 Synchronization Algorithms

2.1 Time and Network Models

Assume a network consisting of $n$ nodes, formally represented by a directed graph $G = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ the set of arcs defining the structure of inter-node communications. Denote by $\mathcal{N}^+_i$ the out-neighborhood and by $\mathcal{N}^-_i$ the in-neighborhood of node $i$, $i = 1, \ldots, n$. Assume that each node has a local clock, whose output, defining local time, is given for any absolute time $t \in \mathbb{R}$ by

$$
\tau_i(t) = \alpha_i t + \beta_i + \xi_i(t),
$$

where $\alpha_i$ is the local drift (gain), $\beta_i$ is the local offset, while $\xi_i(t)$ is measurement noise, appearing due to equipment instabilities, round-off errors, thermal noise, etc. [Liao and Barooah, 2013a; Liao and Barooah, 2013b; Schenato and Fiorentin, 2011; Stanković et al., 2012].

Each node $i$ applies an affine transformation to $\tau_i(t)$, producing the corrected local time

$$
\bar{\tau}_i(t) = a_i \tau_i(t) + b_i = g_i t + f_i + a_i \xi_i(t),
$$

where $a_i$ and $b_i$ are local correction parameters, $g_i = a_i \alpha_i$ is the corrected drift and $f_i = a_i \beta_i + b_i$ the corrected offset, $i = 1, \ldots, n$.

The goal of distributed time synchronization is to provide a common virtual clock, i.e., equal corrected drifts $g_i$ and equal corrected offsets $f_i$, $i = 1, \ldots, n$ by distributed real-time estimation of the parameters $a_i$ and $b_i$. We assume that the nodes communicate according to the broadcast gossip scheme, e.g., [Nedić, 2011; Aysal et al., 2009; Bolognani et al., 2012], without global supervision or fusion center. Namely, we assume that each node $j \in \mathcal{N}$ has its own local communication clock that ticks according to a Poisson process with the rate $\mu_j$, independently of other nodes. At each tick of its communication clock (denoted by $t^j_b$, $b = 0, 1, 2, \ldots$), node $j$ broadcasts its current local time (together with its current estimates of
correction parameters) to its out-neighbors \( i \in N_j^+ \). Each node \( i \in N_j^+ \) hears the broadcast with probability \( p_{ij} > 0 \). Let \( \{ t_l^{j,i} \} \), \( l = 0, 1, 2, \ldots \), be the sequence of absolute time instants corresponding to the messages heard by node \( i \). The message sent at \( t_l^{j,i} \) is received at node \( i \) at the time instant

\[
\bar{t}_l^{j,i} = t_l^{j,i} + \delta_l^{j,i},
\]

where \( \delta_l^{j,i} \) represents the corresponding communication delay. See (Xiong and Kishore, 2009; Leng and Wu, 2011; Chaudhari et al., 2008; Freris et al., 2011; Choi et al., 2012) for presentation of physical and technical sources of the delays. We assume in the sequel that the communication delay can be decomposed as

\[
\delta_l^{j,i} = \bar{\delta}_l^{j,i} + \eta_l^{j,i}(t_l^{j,i}),
\]

(3)

where \( \bar{\delta}_l^{j,i} \) is assumed to be constant (depending only on the chosen arc \((j, i)\)), while \( \eta_l^{j,i}(t_l^{j,i}) \) represents a stochastically time-varying component with zero mean. After receiving a message from node \( j \), node \( i \) reads its current local time, calculates its own current corrected local time and updates the values of its correction parameters \( a_i \) and \( b_i \). The process is repeated after each tick of the communication clock of any node in the network; we assume, as usually, that time is dense and only one communication clock can tick at a given time (Nedić, 2011).

2.2 Drift Correction Algorithm

The drift correction algorithm is given as Algorithm 1 and described in the following. The recursion for updating the value of parameter \( a_i \) at node \( i \), as a response to a message coming from node \( j \), is based on the following error function:

\[
\bar{\varphi}_l^{a_i}(t_l^{j,i}) = \Delta \bar{\tau}_j(t_l^{j,i}) - \Delta \bar{\tau}_i(t_l^{j,i}),
\]

(4)

where \( \Delta \bar{\tau}_j(t_l^{j,i}) \) and \( \Delta \bar{\tau}_i(t_l^{j,i}) \) are increments of the corrected local times, given by

\[
\Delta \bar{\tau}_j(t_l^{j,i}) = \bar{\tau}_j(t_l^{j,i}) - \bar{\tau}_j(t_m^{j,i}) = a_j \Delta \tau_j(t_l^{j,i}),
\]

\[
\Delta \bar{\tau}_i(t_l^{j,i}) = \bar{\tau}_i(t_l^{j,i}) - \bar{\tau}_i(t_m^{j,i}) = a_i \Delta \tau_i(t_l^{j,i}),
\]

where \( m \in \{0, \ldots, l-1\} \),

\[
\Delta \tau_j(t_l^{j,i}) = \tau_j(t_l^{j,i}) - \tau_j(t_m^{j,i}) = \alpha_j \Delta t_l^{j,i} + \Delta \xi_j(t_l^{j,i}),
\]

\[
\Delta \tau_i(t_l^{j,i}) = \alpha_i \Delta t_l^{j,i} + \Delta \xi_i(t_l^{j,i}),
\]
\[ \Delta t_{ij}^l = t_{ij}^l - t_{mj}^l, \quad \Delta \xi_j(t_{ij}^l) = \xi_j(t_{ij}^l) - \xi_j(t_{mj}^l), \quad \Delta \bar{t}_j^l = \bar{t}_j^l - \bar{t}_m^l = \Delta t_{ij}^l + \Delta \delta_{ij}^l, \]

with \( \Delta \delta_{ij}^l = \delta_{ij}^l - \delta_{im}^l \), and \( \Delta \xi_i(\bar{t}_{ij}^l) = \xi_i(\bar{t}_{ij}^l) - \xi_i(\bar{t}_{im}^l) \); by (3), we have \( \Delta \delta_{ij}^l = \Delta \eta_i(\bar{t}_{ij}^l) \), where \( \Delta \eta_i(\bar{t}_{ij}^l) = \eta_i(\bar{t}_{ij}^l) - \eta_i(\bar{t}_{im}^l) \).

Here \( m \) denotes the index of the past time instant with respect to which the time increment is calculated. The choice of \( m \) leads to different definitions of the time increment, and to algorithms with different properties. In this paper we shall consider the following three characteristic cases (which we denote AlgDrift.a, AlgDrift.b and AlgDrift.c; see Algorithm 1):

a) \( m = l - L \), where \( L \) is a predefined constant (AlgDrift.a);

b) \( m = \lfloor \nu l \rfloor \) \((0 < \nu < 1)\), where \( \lfloor x \rfloor \) denotes the largest integer less than or equal to \( x \) (AlgDrift.b);

c) \( m = l_0 \), where \( l_0 \) is a fixed integer (AlgDrift.c).

Remark 1 In case a) we have finite memory determined by \( L \), which can be carefully chosen in advance (see the discussion below). The case b), when we have both \( \lim_{l \to \infty} m = \infty \) and \( \lim_{l \to \infty} (l - m) = \infty \), and the case c) are conceptually very important (see Theorem 2 below). It will be shown below that high performance and convergence to a common virtual clock can be theoretically obtained in case b), while case a) offers solutions efficiently implementable in practice.

Using (4) we define the following updating procedure for parameter \( a_i \) at node \( i \), to be executed immediately after node \( i \) receives the message from node \( j \) \((j = 1, \ldots, n, i \in N_j^+)\):

\[ \hat{a}_i(\bar{t}_{ij}^{l+i}) = \hat{a}_i(\bar{t}_{ij}^l) + \varepsilon_{ai}(\bar{t}_{ij}^l) \gamma_{ij} \phi_i(\bar{t}_{ij}^l), \quad (5) \]

where \( \gamma_{ij} \) are a priori adopted nonnegative weights expressing relative importance of communication links (their role will be discussed below), \( \phi_i(\bar{t}_{ij}^l) = \Delta \hat{\tau}_j(\bar{t}_{ij}^l) - \Delta \hat{\tau}_i(\bar{t}_{ij}^l) \),

\[ \Delta \hat{\tau}_j(\bar{t}_{ij}^l) = \Delta \hat{\tau}_j(\bar{t}_{ij}^l)|_{t_{ij} = \hat{a}_j(\bar{t}_{ij}^l)}; \quad (6) \]

\[ \Delta \hat{\tau}_i(\bar{t}_{ij}^l) = \Delta \hat{\tau}_i(\bar{t}_{ij}^l)|_{t_{ij} = \hat{a}_i(\bar{t}_{ij}^l)}; \quad (7) \]

\( \hat{a}_j(\bar{t}_{ij}^l) \) and \( \hat{a}_i(\bar{t}_{ij}^l) \) are the old estimates, \( \hat{a}_i(\bar{t}_{ij}^{l+i}) \) the new estimate, while \( \varepsilon_{ai}(\bar{t}_{ij}^l) \) is a positive step size. The corresponding pseudocodes are presented
Algorithm 1 AlgDrift.a, AlgDrift.b and AlgDrift.c

for All the nodes \( i \in \mathcal{N} \) do
    Initialize \( \hat{a}_i(t^j_{0,i}) = 1 \)
end for

loop
    if Tick \( t^j_b \) of a local communication clock of a node \( j \in \mathcal{N} \) then
        Read the current local time value \( \tau_j(t^j_b) \)
        Broadcast \( \tau_j(t^j_b) \) and \( \hat{a}_j(t^j_b) \) to the out-neighbours \( \mathcal{N}^+_j \)
    end if
end loop

loop
    if A message received by a node \( i \in \mathcal{N} \) from a node \( j \in \mathcal{N}_j^- \) (at absolute time \( \bar{t}_{j,i} \)) then
        if The first message from the node \( j \) then
            Save the received initial local time of node \( j \) \( \tau_j(t^j_{0,i}) \)
            Read and save the initial local time \( \tau_i(\bar{t}_{j,i}) \)
        else
            Read the current local time value \( \tau_i(\bar{t}_{j,i}) \)
            Calculate \( \Delta \hat{\tau}_j(t^j_{i}) \) and \( \Delta \hat{\tau}_i(t^j_{i}) \) according to (6) and (7), where \( m = l - L \) for AlgDrift.a, \( m = [vl] \) for AlgDrift.b, and \( m = 0 \) for AlgDrift.c
            Calculate a new estimate of the drift correction parameter according to (5)
        end if
    end if
end if
end loop
as Algorithm 1. The updating procedure (5) generates, in such a way, recursions of distributed asynchronous stochastic approximation type. It will be assumed that the initial estimates are \( \hat{a}_i(\bar{t}^0_{j,i}) = 1 \).

In terms of the corrected drift \( \hat{g}_i(\cdot) = \hat{a}_i(\cdot) \alpha_i \), (5) gives:

\[
\hat{g}_i(\bar{t}^{j,i}_{j,i} + l) = \hat{g}_i(\bar{t}^{j,i}_{j,i}) + \epsilon_a(\bar{t}^{j,i}_{j,i}) \gamma_{ij} \hat{\psi}_i(\bar{t}^{j,i}_{j,i}),
\]

where

\[
\hat{\psi}_i(\bar{t}^{j,i}_{j,i}) = \alpha_i \{ [\hat{g}_j(t^{j,i}_{j,i}) - \hat{g}_i(\bar{t}^{j,i}_{j,i})] \Delta_{t^{j,i}_{j,i}} + \frac{1}{\alpha_j} \hat{g}_j(t^{j,i}_{j,i}) \Delta \xi_j(t^{j,i}_{j,i}) - \frac{1}{\alpha_i} \hat{g}_i(\bar{t}^{j,i}_{j,i}) \Delta \xi_i(\bar{t}^{j,i}_{j,i}) - \hat{g}_i(\bar{t}^{j,i}_{j,i}) \Delta \eta_i(\bar{t}^{j,i}_{j,i}) \}.
\]

**Remark 2** The basic drift correction estimation scheme (5), (8) is autonomous (independent of offset correction), with a role analogous to the distributed drift estimation schemes in (Schenato and Fiorentin, 2011; Tian et al., 2016; Tian, 2015). However, it does not belong to the class of the so-called CBTS algorithms (Tian et al., 2016): it is structurally different and simpler, not requiring the step of relative drift estimation, which introduces unnecessary dynamics and additional nonlinearities. Even a robustified version of the algorithm in (Schenato and Fiorentin, 2011) proposed in (Garone et al., 2015) cannot achieve consensus of corrected drifts in the case of stochastic delays.

**Remark 3** For \( L = 1 \) in AlgDrift.a and \( l_0 = 0 \) in AlgDrift.c one obtains the drift correction algorithms proposed in (Stanković et al., 2016). Notice that the idea to use \( m = l - 1 \) in the increment definition has its roots in the algorithm proposed in (Schenato and Fiorentin, 2011), while the choice \( m = 0 \) has been initially described in (Tian, 2015) (both within the context of CBTS algorithms). A pseudo periodic version of (5) with \( m = l - 1 \) has been proposed and analyzed in (Stanković et al., 2012).

### 2.3 Offset Correction Algorithm

The offset correction algorithm is given as Algorithm 2 and described below. The recursion for updating parameter \( b_i \) is based on the following error function:

\[
\varphi^b_i(\bar{t}^{j,i}_{j,i}) = \bar{\tau}_j(t^{j,i}_{j,i}) - a_j T_j(t^{j,i}_{j,i}) - (\bar{\tau}_i(\bar{t}^{j,i}_{j,i}) - a_i T_i(\bar{t}^{j,i}_{j,i})) + c_i,
\]

\( j = 1, \ldots, n, \; i \in \mathcal{N}^+_j \), where

\[
T_j(t^{j,i}_{j,i}) = \Delta \tau_j(t^{j,i}_{j,i})|_{m=0}, \; T_i(\bar{t}^{j,i}_{j,i}) = \Delta \tau_i(\bar{t}^{j,i}_{j,i})|_{m=0},
\]
while $c_i$ is an additional delay compensation parameter.

Using (10), we come up with the following updates for $b_i$ and $c_i$ (based on the assumption that an estimate of $a_i$ is given):

$$
\hat{b}_i(t_i^{j,i} + ) = \hat{b}_i(t_i^{j,i}) + \hat{c}_i(t_i^{j,i}) \gamma_{ij} \gamma_i \psi_i(t_i^{j,i})
$$

(12)

$$
\hat{c}_i(t_i^{j,i} + ) = \hat{c}_i(t_i^{j,i}) - \hat{b}_i(t_i^{j,i}) \gamma_{ij} \gamma_i \psi_i(t_i^{j,i})
$$

(13)

where $\phi_i(t_i^{j,i}) = \hat{\tau}_j(t_i^{j,i}) - \hat{a}_j(t_i^{j,i}) T_j(t_i^{j,i}) - (\hat{\tau}_i(t_i^{j,i}) - \hat{a}_i(t_i^{j,i}) T_i(t_i^{j,i})) + \hat{c}_i(t_i^{j,i})$.

The initial estimates are supposed to be $\hat{b}_i(t_i^{j,i}) = 0$ and $\hat{c}_i(t_i^{j,i}) = 0$. The estimates of the drift correction parameters can be generated by any convenient algorithm; when it is generated by (5), we obtain a complete new time synchronization algorithm.

In terms of $g_i(\cdot) = \hat{a}_i(\cdot) \alpha_i$ and $\hat{f}_i(\cdot) = \hat{a}_i(\cdot) \beta_i + \hat{b}_i(\cdot)$, (12) and (13) become:

$$
\hat{f}_i(t_i^{j,i} + ) + \Delta\hat{g}_i(t_i^{j,i} + ) = \hat{f}_i(t_i^{j,i}) + \hat{c}_i(t_i^{j,i}) \gamma_{ij} \gamma_i \psi_i(t_i^{j,i}),
$$

(16)

$$
\hat{c}_i(t_i^{j,i} + ) = \hat{c}_i(t_i^{j,i}) - \hat{b}_i(t_i^{j,i}) \gamma_{ij} \gamma_i \psi_i(t_i^{j,i}),
$$

(17)

where $\Delta\hat{g}_i(t_i^{j,i} + ) = \hat{g}_i(t_i^{j,i} + ) - \hat{g}_i(t_i^{j,i})$ and

$$
\psi_i(t_i^{j,i}) = [\hat{\gamma}_j(t_i^{j,i}) - \hat{g}_i(t_i^{j,i}) T_i^{j,i}] l_i^{j} + \hat{f}_j(t_i^{j,i}) - \hat{f}_i(t_i^{j,i}) - \hat{g}_j(t_i^{j,i}) [\phi_i^j + \eta_i(t_i^{j,i})]
$$

+ $\hat{c}_i(t_i^{j,i}) \gamma_{ij} \gamma_i \psi_i(t_i^{j,i}) - \alpha_i \hat{g}_i(t_i^{j,i}) \xi_i(t_i^{j,i}).
$$

(18)

A consensus-based modification of (13) and (17) will be considered apart. This modification is formally obtained by replacing $\hat{c}_i(t_i^{j,i})$ at the right hand side of (13) and (17) by the following convex combination

$$
\hat{c}_i^{con}(t_i^{j,i}) = \sigma_i \hat{c}_i(t_i^{j,i}) + (1 - \sigma_i) \hat{c}_i(t_i^{j,i})
$$

(19)

with tuning parameter $0 < \sigma_i \leq 1$. This modification is motivated by a realistic assumption that the delays in the network are not too far from each other, with the aim to achieve smaller dissipation of the convergence points for $\hat{f}_i(\cdot)$ (see Remark 9 below). We refer to this algorithm as AlgOffset.b. The pseudocode of the algorithms AlgOffset.a and AlgOffset.b are presented as Algorithm 2.
Algorithm 2 \textit{AlgOffset.a} and \textit{AlgOffset.b}

for All the nodes $i \in \mathcal{N}$ do
    Initialize $\hat{b}_i(\bar{t}_{j,i}^0) = 0$ and $\hat{c}_i(\bar{t}_{j,i}^0) = 0$
end for

loop
    if Tick $t_b^j$ of a local communication clock of a node $j \in \mathcal{N}$ then
        Read the current local time value $\tau_j(t_b^j)$
        Broadcast $\tau_j(t_b^j), \hat{a}_j(t_b^j), \hat{b}_j(t_b^j)$ and $\hat{c}_j(t_b^j)$ to the out-neighbours $\mathcal{N}_j^+$
    end if
end loop

loop
    if A message received by a node $i \in \mathcal{N}$ from a node $j \in \mathcal{N}_j^-$ (at absolute time $\bar{t}_{j,i}^l$) then
        Read the current local time value $\tau_i(\bar{t}_{j,i}^l)$
        Calculate $\tilde{\tau}_j(t_{j,i}^l)$ and $\bar{\tau}_i(t_{j,i}^l)$ using (14) and (15)
        Calculate $T_j(t_{j,i}^l)$ and $T_i(\bar{t}_{j,i}^l)$ using (11)
        Calculate new estimates of the offset correction parameters according to (12) and (13) (and (19) for the algorithm \textit{AlgOffset.b})
    end if
end loop
Remark 4 The proposed offset estimation scheme represented by (12)–(17) is based on two major modifications of the basic error function $\varphi_i^b(\bar{t}_{j,i}^l)^0 = \bar{\tau}_j(t_{i,j}^l) - \bar{\tau}_i(\bar{t}_{j,i}^l)$, which has been utilized in all the existing CBTS algorithms (see (Tian et al., 2016) and the references therein). The first modification introduces two easily computable terms $T_j(t_{i,j}^l)$ and $T_i(\bar{t}_{j,i}^l)$, the role of which is to cope with the unboundedly increasing term $t_{i,j}^l$ in the expression for $\varphi_i^b(\bar{t}_{j,i}^l)^0$, in such a way as to replace it with the bounded term $t_{i,j}^l$ in $\varphi_i^b(\bar{t}_{j,i}^l)$. The second modification consists of introducing a new parameter to be updated, $\hat{c}_i(\bar{t}_{j,i}^l)$. Estimation of this parameter is aimed at coping directly with the effects of communication delays and enabling convergence of the offset correction parameter estimates.

Remark 5 The proposed time synchronization algorithm requires very small communication and computation efforts. At each tick $t_{i,j}^l$, a packet is sent by the $j$-th node to its neighbors $i \in N_j^+$, containing the current local time $\tau_j(t_{i,j}^l)$ and the current local drift and offset correction parameter estimates $\hat{a}_j(t_{i,j}^l)$, $\hat{b}_j(t_{i,j}^l)$ and $\hat{c}_j(t_{i,j}^l)$. After receiving this packet, the neighbors calculate the corresponding $\Delta \hat{\tau}_j(t_{i,j}^l)$, $\Delta \hat{\tau}_i(\bar{t}_{j,i}^l)$, $\hat{\tau}_j(t_{i,j}^l)$, $\hat{\tau}_i(\bar{t}_{j,i}^l)$, $T_j(t_{i,j}^l)$ and $T_i(\bar{t}_{j,i}^l)$, and update their own parameter estimates according to (5), (12), (13) and (19). The same procedure is repeated after each new tick of any of the nodes.

2.4 Global Model

Next we derive a global model of the overall time synchronization network. Parameter updating at the network level is driven by a global virtual communication clock, with the rate equal to $\mu = \sum_{i=1}^n \mu_i$, that ticks whenever any of the local communication clocks tick (e.g., (Nedić, 2011; Aysal et al., 2009)). Starting from this fact, a global model for the whole network has been defined in (Stanković et al., 2016) in the form of a recursion in which the $k$-th iteration corresponds to the $k$-th tick of the global virtual communication clock. In this paper, we shall adopt an alternative approach, providing more direct insight into the whole updating process. Namely, we shall assume that every local update in the network produces a unique iteration number $k$ in the global model of the parameter estimates, and, vice versa, that every $k$ is connected to a local node update (for an update of $i$-th node, the corresponding continuous time instant is $\bar{t}_{i,j}^l$ for some $j$ and $l$). In such a way, at a click of $j$-th communication clock we have $N(j)$ consecutive updates or iterations (assuming that we have only one update at a time), $N(j) \leq |N_j^+|$. Following analogous approaches in (Nedić,
with finite sets of possible realizations composed of \( \tau_i(t_{ij}^k) = \tau_i(k), \bar{\tau}_i(t_{ij}^k) = \bar{\tau}_i(k), \xi_j(t_{ij}^k) = \xi_j(k) \), etc; accordingly, we also write \( \tau_j(t_{ij}^k) = \tau_j(k), \bar{\tau}_j(t_{ij}^k) = \bar{\tau}_j(k), \xi_j(t_{ij}^k) = \xi_j(k) \), etc. In the case of delays, we write \( \hat{\delta}^{ij} = \bar{\delta}_j(k) \) and \( \eta_i(t_{ij}^k) = \eta_i(k) \).

Assume that \( k \) is connected to an update at node \( i \), initiated by a tick of node \( j \). Let \( \hat{g}(k) = [\hat{g}_1(k) \ldots \hat{g}_n(k)]^T, \hat{f}(k) = [\hat{f}_1(k) \ldots \hat{f}_n(k)]^T \) and \( \hat{c}(k) = [\hat{c}_1(k) \ldots \hat{c}_n(k)]^T \), where \( \hat{g}_i(k) = \hat{a}_i(k)\mu_i, \hat{a}_i(t_{ij}^k) \), \( \hat{f}_j(k) = \hat{a}_j(k)\eta_j + \hat{b}_j(k), \hat{b}_j(k) = \hat{b}_j(t_{ij}^k) \) and \( \hat{c}_i(k) = \hat{g}_i(t_{ij}^k), \mu = 1, \ldots, n \). Then, (8) gives

\[
\hat{g}(k+1) = \hat{g}(k) + \varepsilon^a(k)Z(k)\hat{g}(k),
\]

where \( \varepsilon^a(k) = \text{diag}\{e_1^a(k), \ldots, e_n^a(k)\} \), \( \varepsilon^a(t_i^k) = \varepsilon_i^a(t_{ij}^k) \) (see (5)),

\[
Z(k) = A\Gamma(k)\Delta t(k) + N_g(k),
\]

\[ A = \text{diag}\{\alpha_1, \ldots, \alpha_n\}, \Gamma(k) = [\Gamma(k)_{\mu\nu}], \text{with } \Gamma(k)_{ii} = -\gamma_{ij} \text{ and } \Gamma(k)_{ij} = \gamma_{ij}, \]

with \( \Gamma(k)_{\mu\nu} = 0 \) otherwise, \( \Delta t(k) = t_{ij}^k - t_{im}^k \), while the noise term is defined as

\[ N_g(k) = -A\Gamma_d(k)\Delta \eta_d(k) + A\Gamma(k)\Delta \xi_d(k)A^{-1}, \]

where \( \Gamma_d(k) = \text{diag}\{\text{diag}\{\gamma_{ij}, \ldots, \gamma_{nj}\}\omega(k)\}, \omega(k) = [\omega_1(k) \ldots \omega_n(k)]^T, \omega_i(k) = 1, \omega_{\mu}(k) = 0 \) for \( \mu \neq i, \Delta \eta_d(k) = \text{diag } \Delta \eta(k), \Delta \eta(k) = [\Delta \eta_1(k) \ldots \Delta \eta_n(k)]^T, \Delta \xi_d(k) = \text{diag } \Delta \xi(k) \text{ and } \Delta \xi(k) = [\Delta \xi_1(k) \ldots \Delta \xi_n(k)]^T. \]

Similarly, from (16) and (17) we obtain

\[
\hat{f}(k+1) + \Delta \hat{g}(k+1) = \hat{f}(k) + \varepsilon^b(k)Y(k)
\]

\[
\hat{c}(k+1) = \hat{c}(k) - \varepsilon^b(k)Y(k),
\]

where \( \Delta \hat{g}(k+1) = \text{diag } \omega(k)(\hat{g}(k+1) - \hat{g}(k)), Y(k) = \Gamma(k)\hat{f}(k) + [\xi_0(t_{ij}^k)\Gamma(k) - \Gamma_d(k)\hat{d}(k) - \Gamma_d(k)\eta_d(k) + \Gamma(k)\xi_0^0(t_{ij}^k)A^{-1}]\hat{g}(k) + \Gamma_d(k)\hat{c}(k), \xi_0(t_{ij}^k) = t_{ij}^k, \hat{d}(k) = \text{diag } \delta(k), \hat{c}(k) = [\delta_{ij}^1 \ldots \delta_{ij}^n]^T, \eta_d^0(k) = \text{diag } \eta^0(k), \xi_0^0(k) = [\xi_0^0(k) \ldots \xi_0^0(t_{ij}^k)]^T, \xi_0(t_{ij}^k) = \text{random sequences with finite sets of possible realizations composed of } t_{ij}^k, \eta_i(t_{ij}^k) \text{ and } \xi_j(t_{ij}^k) \text{ (or } \xi_i(t_{ij}^k)) \text{, obtained at each } k \text{ by choosing } j \text{ and } i \text{ at random.}

In AlgOffset, \( \hat{c}(k) \) is replaced by \( \hat{c}^{com}(k) = C(k)\hat{c}(k) \), where \( C(k) = [C(k)_{\mu\nu}] \), with \( C(k)_{\mu\nu} = \sigma_\mu \) and \( C(k)_{\mu\nu} = 1 - \sigma_\mu \) for all \( \mu \in N_j^+ \), with \( C(k)_{\mu\nu} = 0 \) otherwise.
3 Convergence Analysis

3.1 Preliminaries

Within the exposed general setting, we additionally assume:

(A1) Graph $G$ has a spanning tree.

(A2) $\{\xi_i(k)\}$ and $\{\eta_i(k)\}$, $i = 1, \ldots, n$, are mutually independent zero mean i.i.d. random sequences, bounded w.p.1.

(A3) The step sizes $\epsilon_i^a(k)$ and $\epsilon_i^b(k)$ are defined in the following way:

$\epsilon_i^a(k) = \epsilon_i(k)_{\zeta = \zeta'}$ for AlgDrift.a,

$\epsilon_i^a(k) = \epsilon_i(k)_{\zeta = 1 + \zeta'}$ for AlgDrift.b and AlgDrift.c, and

$\epsilon_i^b(k) = \epsilon_i(k)_{\zeta = \zeta''}$ for AlgOffset.a and AlgOffset.b,

where $\epsilon_i(k) = \nu_i(k)^{-\zeta}$, $\nu_i(k) = \sum_{m=1}^{k} I\{\text{node } i \text{ received a message}\}$, representing the number of updates of node $i$ up to the instant $k$ ($I\{\cdot\}$ denotes the indicator function), while $\frac{1}{2} < \zeta', \zeta'' \leq 1$.

Remark 6 (A1) implies that graph $G$ has a center node from which all the remaining nodes are reachable (Olfati-Saber et al., 2007; Stanković et al., 2015). (A2) is a standard assumption, in which boundedness, which always holds in practice, is introduced for making derivations easier. (A3) is practically very important: it eliminates the need for a centralized clock which would define the common step size for all the nodes as a function of $k$. The choice of the exponent in the expression for $\epsilon_i^a(k)$ for AlgDrift.b and AlgDrift.c is motivated by the properties of the random variable $\Delta t(k)$ which diverges linearly to infinity (see Theorem 2). The choice of $\zeta'$ and $\zeta''$ is standard for stochastic approximation algorithms.

Asymptotical behavior of the step size is given by the following lemma. Proofs of all the lemmas and theorems are given in the appendix.

Lemma 1 Let (A1) and (A3) be satisfied, let $p_i$ be the unconditional probability of node $i$ to update its parameters at $k$-th iteration, and let $\zeta > 0$. Then, for a given $q \in (0, \frac{1}{2})$, there exists $k$ such that w.p.1 for all $k \geq k$

$$\epsilon_i(k) = \frac{1}{k^{\zeta}} (\frac{\bar{N}}{p_i})^\zeta + \tilde{\epsilon}_i(k),$$

(23)

where $\bar{N} = E_j \{E\{N(j)\}\}$ represents the average number of updates per one tick of the global virtual clock and $|\tilde{\epsilon}_i(k)| \leq \tilde{\epsilon}_i \frac{1}{k^{\zeta + \frac{1}{2} - q}}$, $0 < \tilde{\epsilon}_i < \infty$, $i = 1, \ldots, n$. 

13
Properties of the matrix $\Gamma(k)$ defined in the previous section are essential for convergence of (20–22); its expectation $\bar{\Gamma} = E\{\Gamma(k)\}$ has a central role in the analysis, since it contains all the information about the network structure and the weights of particular links. It has the structure of a weighted Laplacian matrix for $G$:

$$
\bar{\Gamma} = \begin{bmatrix}
-\sum_{j,j\neq 1} \gamma_{1j} \pi_{1j} & \gamma_{12} \pi_{12} & \cdots & \gamma_{1n} \pi_{1n} \\
\gamma_{21} \pi_{21} & -\sum_{j,j\neq 2} \gamma_{2j} \pi_{2j} & \cdots & \gamma_{2n} \pi_{2n} \\
\vdots & \ddots & \ddots & \vdots \\
\gamma_{n1} \pi_{n1} & \gamma_{n2} \pi_{n2} & \cdots & -\sum_{j,j\neq n} \gamma_{nj} \pi_{nj}
\end{bmatrix}
$$

(24)

($\gamma_{ij} = 0$ when $j \notin N_i$), where $\pi_{ij}$ is unconditional probability that the node $j$ broadcasts and node $i$ updates its parameters as a consequence ($\pi_{ij} = \pi_j p_{ij}$, where $\pi_j$ is the unconditional probability for node $j$ to broadcast).

According to (20) and Lemma 1, we shall consider $B(k) = P^{-\zeta} A \Gamma(k)$ and $\bar{B} = E\{B(k)\} = P^{-\zeta} A \bar{\Gamma}$ ($P^{-\zeta} = N^\zeta \text{diag}\{p_1^{-\zeta}, \ldots, p_n^{-\zeta}\}$). The latter matrix has the following properties.

**Lemma 2 (Stanković et al., 2015)** Matrix $\bar{B}$ has one eigenvalue at the origin, and the remaining ones in the left half plane. Let $T = \begin{bmatrix} 1^{T} \end{bmatrix}_{n \times (n-1)}$, where $T_{n \times (n-1)}$ is such that $\text{span}\{T_{n \times (n-1)}\} = \text{span}\{\bar{B}\}$, while $1 = [1 \ldots 1]^T$. Then,

$$
T^{-1} \bar{B} T = \begin{bmatrix}
0 & \bar{B}^* \\
\bar{B} \quad 0_{(n-1) \times 1}
\end{bmatrix},
$$

(25)

where $\bar{B}^*$ is Hurwitz.

Consequently, there exists $R^g > 0$ satisfying

$$
R^g \bar{B}^* + \bar{B}^* T R^g = -Q^g,
$$

(26)

for any given $Q^g > 0$. It also follows from the derivation of (25) that $T^{-1} B(k) T = \begin{bmatrix} B_1(k) \\
B_2(k) \end{bmatrix}_{0 \times 1}$, with $E\{B_1(k)\} = 0$ and $E\{B_2(k)\} = \bar{B}^*$.

Properties of the random variable $\Delta t(k)$ are important for further analysis of (20).
Lemma 3  \( E\{\Delta t(k)\} = \frac{l-m}{\mu_j}, \) \( \text{var}\{\Delta t(k)\} = \frac{l-m}{\mu^2_j}, \) where \( l-m = L \) for AlgDrift.a, \( l-m = [(1-\nu)l] \) for AlgDrift.b and \( l-m = l \) for AlgDrift.c; for large \( l \), we have \( l \sim \pi_{ij}k \).

3.2 Convergence of Drift Correction Algorithm

After coming back to (20), we first insert \( \varepsilon_a(k) \) from (23). Then, we introduce \( \tilde{g}(k) = \frac{1}{T} - \hat{g}(k) \) and decompose \( \tilde{g}(k) = \tilde{g}(k)^{[1]} \) as \( \tilde{g}(k)^{[1]} = \tilde{g}_1(k) \) and \( \tilde{g}(k)^{[2]} = [\tilde{g}_2(k) \ldots \tilde{g}_n(k)]^T \). After neglecting the higher order terms from (23), we obtain

\[
\tilde{g}(k+1)^{[1]} = \tilde{g}(k)^{[1]} + \frac{1}{k^2} F_1(k) \Delta t(k) \tilde{g}(k)^{[2]} + \frac{1}{k^2} H_1(k) \tilde{g}(k) \tag{27}
\]

\[
\tilde{g}(k+1)^{[2]} = \{ I + \frac{1}{k^2} [B^* + F_2(k)] \Delta t(k) \} \tilde{g}(k)^{[2]} + \frac{1}{k^2} H_2(k) \tilde{g}(k), \tag{28}
\]

where matrices \( F_1(k) \) and \( F_2(k) \) are defined by

\[
T^{-1} [B(k) - B^*] T = \begin{bmatrix}
0 & F_1(k) \\
0_{(n-1) \times 1} & F_2(k)
\end{bmatrix},
\]

while \( H_1(k) \) and \( H_2(k) \) are defined by \( T^{-1} P^{-c} N_g(k) T = \begin{bmatrix} H_1(k) \\ H_2(k) \end{bmatrix} \).

We now have the following convergence result for the drift correction algorithm.

Theorem 1 Let assumptions (A1)–(A3) be satisfied. Then, \( \tilde{g}(k)^{[1]} \) from (27) converges to a random variable \( \chi^* \) with bounded second moment, and \( \tilde{g}(k)^{[2]} \) from (28) to zero in the mean square sense and w.p.1 for Algorithm 1; in other words, \( \tilde{g}(k) \) generated by (20) converges for all three choices of \( m \) to \( \tilde{g}_\infty = \chi^* 1 \) in the mean square sense and w.p.1.

The rate of convergence of the drift estimation scheme is of utmost importance not only for the convergence of local clocks to a common virtual clock, but also for the convergence of the offset estimation algorithm. Asymptotic rate of convergence to consensus of the algorithm (20) will be studied through the behavior of \( \tilde{g}(k)^{[2]} \) in (28), using the methodology of (Chen, 2002, Chapter 3).

Theorem 2 Let (A1)–(A3) hold. Then, \( z(k) = k^d \tilde{g}(k)^{[2]} \), where \( d > 0 \) and \( \tilde{g}(k)^{[2]} \) is defined by (28), converges to zero in the mean square sense and
w.p.1, when $\zeta' < 1$ for:
\[ \zeta d < \zeta' - \frac{1}{2} \] (AlgDrift.a),
\[ \zeta d < \frac{1}{2} + \zeta' \] (AlgDrift.b) and
\[ \zeta d < 1 \] (AlgDrift.c),
and when $\zeta' = 1$ for:
\[ d < \min\left(\frac{1}{2}, 2qr\right) \] (AlgDrift.a),
\[ d < \min\left(\frac{3}{4}, qr\right) \] (AlgDrift.b) and
\[ d < \min\left(\frac{1}{2}, qr\right) \] (AlgDrift.c),
where
\[ r = \frac{\lambda_{\min}(Q)}{\lambda_{\max}(R)}, \quad q = \frac{L}{\max_{i,j}(\mu_j \rho_{ij})} \]
for AlgDrift.a, $q = \frac{1-\nu}{\mu}$ for AlgDrift.b and $q = \frac{1}{\mu}$ for AlgDrift.c.

Remark 7 The applied methodology provides sufficient, in general conservative conditions; the results hold asymptotically, for $k$ large enough. They indicate that the AlgDrift.b provides the best results: in the case when $\zeta' < 1$ the condition $\zeta d > 1$ is achieved, ensuring convergence to a common virtual clock (see Corollary 1 below). This is a consequence of the fulfillment of the requirements for $m$. They ensure an effectively increasing signal-to-noise ratio, together with a sufficient number of realizations of the noise term at the left end of the interval $[m, l]$ (enabling appropriate averaging). AlgDrift.a and AlgDrift.c fail in some of these conditions. However, in practice, it is sufficient to choose $l - m = L$ large enough and to apply AlgDrift.a, avoiding in such a way problems connected with the increase of memory inherent to AlgDrift.b. It will be demonstrated in Section 4 by simulation that practically the best results can be obtained by AlgDrift.a for $L$ moderately high.

Notice that the CBTS algorithms discussed in [Tian et al., 2016] cannot achieve convergence rate $\zeta d > 1$, important for achieving convergence to a global virtual clock (see [34]).

An important conclusion resulting from Theorems 1 and 2 is that
\[ \hat{g}(k) = \chi(k)1 + \hat{g}(k)[2], \] w.p.1
(29)
where $\chi(k) = \hat{g}(k)[1]$ and $\hat{g}(k)[2] = T_{n \times (n-1)} \hat{g}(k)[2]$, with $\chi(k) = \chi^* + o(1)$ and $\|\hat{g}(k)[2]\| = o\left(\frac{1}{k^{\frac{1}{2}}}\right)$. The last relation is fundamental for the convergence analysis of the offset correction estimation.

3.3 Convergence of Offset Correction Algorithm

In this subsection we shall analyze convergence of the offset estimation algorithms AlgOffset.a and AlgOffset.b assuming that the drift correction parameter estimates converging to consensus are available.
where \( \hat{\zeta} \) AlgDrift.a is the unmodified error function pending on \( \Delta \hat{\zeta} \) through the term \( \hat{\zeta} \) role in the offset correction algorithm. It influences Remark 8 Rate of convergence of and \( \hat{\eta} \) to \( \bar{\eta} \) through the term (see Remark 4) makes the vast majority of drift estimation and indirectly through the remaining terms depending on \( \hat{\zeta} \). The standard offset estimation algorithms derived from the unmodified error function \( \hat{\zeta}_0^{(t_{ij}^i)} = \hat{\tau}_j(t_{ij}^i) - \hat{\tau}_i(t_{ij}^i) \) with its linearly increasing term (see Remark 4) makes the vast majority of drift estimation algorithm inapplicable in the case of measurement noise. According to Theorem 3, all the proposed algorithms for drift estimation can be utilized, theoretically under certain constraints. Theorem 3 can be readily generalized in the sense that \( \hat{g}(k) \) can be generated by any algorithm providing sufficient convergence rate to consensus. Notice that for the proposed algorithms with \( \zeta' = 1 \), the related conditions following from Theorem 2 can be readily incorporated.

Algorithm AlgOffset.b is particularly important for practice.

We start the analysis by introducing the following expressions in (21) and (22):

\[
\begin{align*}
\Gamma(k) &= \hat{\Gamma} + \tilde{\Gamma}(k), \quad \Gamma_d(k) = \hat{\Gamma}_d + \tilde{\Gamma}_d(k), \\
\xi^0(k) &= \xi^0 + \tilde{\xi}^0(k), \quad \eta^0(k) = \tilde{\eta}^0 + \tilde{\eta}^0(k), \\
\delta(k) &= \delta^0 + \tilde{\delta}(k), \quad \delta^0(k) = \tilde{\delta}^0 + \tilde{\delta}^0(k),
\end{align*}
\]  

where \( \hat{\Gamma} = E\{\Gamma(k)\} \), \( \hat{\Gamma}_d = E\{\Gamma_d(k)\} \), \( \tilde{\xi}^0 = E\{\xi^0(k)\} = \sum_{j=1}^n \xi(t_{ij}^i) \pi_j \), \( \tilde{\eta}^0 = E\{\eta^0(k)\} = \sum_{j=1}^n \eta_0(t_{ij}^i) \pi_j \), \( \tilde{\delta}^0 = E\{\delta(k)\} = \sum_{j=1}^n \delta_0(t_{ij}^i) \pi_j \) and \( \tilde{\delta}^0 = E\{\delta^0(k)\} = \sum_{j=1}^n \delta_0(t_{ij}^i) \pi_j \). Therefore, \( \{\hat{\Gamma}(k)\}, \{\hat{\Gamma}_d(k)\}, \{\tilde{\xi}^0(k)\}, \{\tilde{\eta}^0(k)\}, \{\tilde{\delta}^0(k)\} \) and \( \{\tilde{\delta}^0(k)\} \) are zero mean i.i.d. random sequences (due to randomness in determining the transmitting node for a given \( k \)).

**Theorem 3** Let assumptions (A1)–(A3) be satisfied and let \( \hat{g}(k) \) be generated by AlgDrift.a with \( \zeta' \in \left( \frac{1}{2}, 1 \right) \), and by AlgDrift.b or AlgDrift.c with \( \zeta' < 1 \). Then, \( \hat{f}(k) \) from (21) converges to \( \hat{f}^* \) and \( \hat{c}(k) \) from (22) converges to \( \hat{c}^* \) in the mean square sense and w.p.1 for all \( \zeta'' \in \left( \frac{1}{2}, 1 \right) \) in the case of AlgDrift.b and AlgDrift.c, and for all \( \zeta'' \in \left( \frac{1}{2}, 1 \right) \), \( \zeta'' > \frac{3}{2} - \zeta' \), in the case of AlgDrift.a; \( \hat{f}^* \) and \( \hat{c}^* \) satisfy the equation

\[
[\hat{\Gamma}; \hat{\Gamma}_d] \hat{h}^* = 0,
\]

where \( \hat{h}^* = ([\hat{f}^* + \chi^* \xi^0_d A^{-1} 1]^T (\hat{c}^* - \chi^* A(\hat{\eta}^0 + \hat{\delta}))^T)^T \).

**Remark 8** Rate of convergence of \( \hat{g}(k) \) to consensus plays an important role in the offset correction algorithm. It influences \( \hat{f}(k) \) in (21) directly through the term \( \Delta \hat{g}(k + 1) \) and indirectly through the remaining terms depending on \( \hat{g}(k) \). The standard offset estimation algorithms derived from the unmodified error function \( \tilde{\xi}_0^{(t_{ij}^i)} = \tilde{\tau}_j(t_{ij}^i) - \tilde{\tau}_i(t_{ij}^i) \) with its linearly increasing term (see Remark 4) makes the vast majority of drift estimation algorithm inapplicable in the case of measurement noise. According to Theorem 3, all the proposed algorithms for drift estimation can be utilized, theoretically under certain constraints. Theorem 3 can be readily generalized in the sense that \( \hat{g}(k) \) can be generated by any algorithm providing sufficient convergence rate to consensus. Notice that for the proposed algorithms with \( \zeta' = 1 \), the related conditions following from Theorem 2 can be readily incorporated.
Theorem 4 Let the assumptions of Theorem 3 hold. Then \( \hat{f}(k) \) and \( \hat{c}(k) \), generated by the algorithm AlgOffset.b ((21), (22) with consensus iterations on \( \hat{c}(k) \) using (19)), converge in the mean square sense and w.p.1 to \( \hat{f}^* \) and \( \hat{c}^* = \hat{c}^{con} \), respectively (\( \hat{c}^{con} \) is a scalar), where \( \hat{f}^* \) and \( \hat{c}^{con} \) satisfy the equation 
\[
M^{con}_1 h^{con} = 0, 
\]
where
\[
M^{con}_1 = \begin{bmatrix}
\bar{\Gamma} & \text{vec}\{\bar{\Gamma}_d\} \\
-\sum_{i=1}^n \tilde{\phi}_i \bar{\Gamma}^{(i)} & -\sum_{i=1}^n \tilde{\phi}_i \text{vec}\{\bar{\Gamma}_d\}_i
\end{bmatrix},
\] (32)
\[
h^{con} = [(\hat{f}^*+\chi_d^* A^{-1}1)^T; \hat{c}^{con} - \sum_{i=1}^n \tilde{\phi}_i \chi^* (A \eta^0 + A \delta)]^T, \text{ with } \tilde{\phi} = [\tilde{\phi}_1 \ldots \tilde{\phi}_n],
\]
\[
\bar{\phi} \bar{C} = \bar{\phi} \text{ and } C = E\{C(k)\}; \text{ } \Gamma^{(i)} \text{ denotes } i\text{-th row of the matrix } \bar{\Gamma}, \text{ and vec}\{\bar{\Gamma}_d\}_i \text{ } i\text{-th element of vec}\{\bar{\Gamma}_d\}.
\]

Remark 9 Theorems 3 and 4 specify precisely the convergence points for the proposed offset correction algorithms. They depend explicitly in the definition of \( h^* \) not only on delays and measurement noise properties, but also on the convergence point of the drift estimation algorithm; they depend on the network properties and the a priori selected weights through the relations (27) and (32). In general, the corrected offsets do not converge to the same point for all the nodes. However, comparison between (31) and (32), indicates clearly that it could be expected to achieve lower dispersion of the components of \( \hat{f}^* \) within \( h^{con} \) due to lower number of degrees of freedom implied by the consensus scheme. Simulation results presented in Section 4 confirm this statement. See also discussion in the subsection on Special Cases given below.

3.4 Special Cases

When communication delays and measurement noise can be neglected, the algorithm AlgOffset.b ((20), (21), (22) with (19)) is able to achieve consensus on both corrected drifts \( \hat{g}_i(k) \) and corrected offsets \( \hat{f}_i(k) \). Namely, in this case we have
\[
\bar{\Gamma} \hat{f}^* + \bar{\Gamma}_d \hat{c}^{con} = 0
\]
\[
\sum_{i=1}^n \tilde{\phi}_i \{(\Gamma^{(i)} \hat{f}^* + (\bar{\Gamma}_d 1) \hat{c}^{con})\} = 0
\] (33)
The equation \( \bar{\Gamma} \hat{f}^* = -\bar{\Gamma}_d 1 \hat{c}^{con} \) has a nontrivial solution for \( \hat{f}^* \) only for \( \hat{c}^{con} = 0 \), having in mind that \( \bar{\Gamma}_d 1 \) does not belong to the column space of \( \bar{\Gamma} \). Therefore, we have \( \bar{\Gamma} \hat{f}^* = 0 \), wherefrom the result follows.
However, according to Theorem 3, AlgOffset.a does not guarantee convergence of $\hat{f}(k)$ to consensus, due to additional degrees of freedom in the solution of $[\Gamma;\Gamma_d]\hat{h}^* = 0$. Elimination of the recursion for $\hat{c}(k)$ leads to divergence of offset estimates.

When the stochastic terms $\xi(\cdot)$ and $\eta(\cdot)$ are equal to zero, it is possible to achieve exponential convergence rate by adopting constant step size in AlgDrift.a, AlgOffset.a and AlgOffset.b, and $\varepsilon_i^a(k) = \varepsilon_i^b(k)^{-1}$ in AlgDrift.b and AlgDrift.c. However, the offset correction algorithm again does not provide consensus, in general.

When, in addition, the delay is equal to zero, the algorithm can be further simplified. Assuming that AlgDrift.a is applied and that $\hat{\varphi}_b^b(\hat{t}^i) = \hat{\varphi}_b^b(\hat{t}^i)^0 = \hat{\tau}_j(t^i_l) - \hat{\tau}_i(t^i_l)$ for offset estimation, we come up with a synchronization algorithm in which $b_i$ is estimated using (12), where $\hat{\varphi}_b^b(\hat{t}^j) = \hat{\tau}_j(t^j_l) - \hat{\tau}_i(t^j_l)$. The resulting algorithm is able to achieve exponential convergence to consensus for both corrected drifts and offsets. This result is obtained for the first time in (Stanković et al., 2012) for pseudo periodic communication sequences (basically, convergence properties of such an algorithm are equivalent to those from (Schenato and Fiorentin, 2011)). When the delays are constant ($\delta_{j,i}^l = \bar{\delta}_{j,i}^l$), the offset estimates diverge. Introduction of a recursion for $\hat{c}(k)$ leads to convergence in the sense of Theorem 3.

### 3.5 Convergence to Common Virtual Clock

As pointed out, the principal aim of clock synchronization is convergence of local corrected times to a common virtual time. In view of the above results, we have:

**Corollary 1** Let (A1)–(A3) be satisfied, with $\zeta' < 1$. Then, for AlgDrift.b and either AlgOffset.a or AlgOffset.b, $\sup_{i,j} \Delta \hat{\tau}_{i,j}(k) = \hat{\tau}_i(k) - \hat{\tau}_j(k)$ is bounded in the mean square sense and w.p.1.

**Remark 10** Since

$$\Delta \hat{\tau}_{i,j}(k) = [\hat{g}_i(k) - \hat{g}_j(k)]t(k) + \hat{f}_i(k) - \hat{f}_j(k),$$

according to Theorem 2 we have that for $\zeta' < 1$ the first term at the right-hand side tends to zero for AlgDrift.b; for AlgDrift.a and AlgDrift.c convergence of $\Delta \hat{\tau}_{i,j}(k)$ cannot be proved. However, it will be shown that the best choice in practice, for finite observation time, is to choose AlgDrift.a with $L$ large enough (see simulation results). Notice also that the differences
\[ \hat{f}_i(k) - \hat{f}_j(k) \text{ remain bounded by virtue of the adoption of the special error function in (10); otherwise, they diverge.} \]

**Remark 11** As already stated, the overall approach in (Garone et al., 2015) suffers from the problem that \([\hat{g}_i(k) - \hat{g}_j(k)]\) does not tend to zero, allowing unbounded increase of the first term in (34). The control-based approach from (Yildirim et al., 2015) attempts to reduce \(\Delta \hat{\tau}_{i,j}(k)\) directly by a careful choice of PI regulator parameters; however, the given analysis does not provide an insight into overall network stability.

### 3.6 Network Weights Tuning and Flooding Scheme

Coefficient \(\gamma_{ij}\) in (5) and (12) is the weight of the update at node \(i\) occurring as a consequence of a tick at node \(j\), \(i, j = 1, \ldots, n\). If one wishes to express high confidence in the precision of a given clock, there are two basic implementations: 1) to increase either all the elements \(\gamma_{ji}\), \(i = 1, \ldots, n\), or the Poisson rate \(\mu_j\) for a given \(j\); 2) to decrease the weights \(\gamma_{ij}\), \(j = 1, \ldots, n\), for a given \(i\). The first way clearly gives more weight to the sender. The second way is related to the receiver, implying lower increments of the local parameter changes at node \(i\), and, therefore, lower influence of the rest of the network to the corresponding local parameter estimates. In the limit, node \(i\) does not update its parameters (\(\gamma_{ij} = 0, j = 1, \ldots, n\)), and becomes a reference node with fixed corrected parameters. The whole algorithm becomes in such a way an algorithm of flooding type (Maroti et al., 2004; Wu et al., 2011; Su and Akyildiz, 2005).

**Corollary 2** Let the assumptions of Theorem 1 be satisfied. Let node \(\lambda\) be a center node in \(G\), with the corrected drift \(\hat{g}_\lambda^*\). Then, after setting \(N_{\lambda}^- = \emptyset\) (or \(\gamma_{\lambda j} = 0, j = 1, \ldots, n\)), algorithm (5) provides convergence of all the corrected drifts \(\hat{g}_i(k)\), \(i = 1, \ldots, n, i \neq \lambda\), to \(\hat{g}_\lambda\) in the mean square sense and w.p.1.

### 4 Simulations

Numerous simulation experiments have been undertaken in order to get a practical insight into the proposed distributed time synchronization algorithm. Different networks have been simulated with variable number of nodes. The assumed network topology corresponds to a modification of Geometric Random Graphs (Gupta and Kumar, 2006). The nodes represent randomly spatially distributed agents within a square area. Initially, the
nodes are assumed to be connected if their Euclidean distance is less than a predefined number: this results in an undirected graph. The obtained graph is modified in such a way as to transform a certain percentage (roughly 10 percent) of the original two-way communications into one-way communications. A program is developed for final optimization, which ensures, on the basis of additional modifications, that assumption (A1) is satisfied. Parameters $\alpha_i$ and $\beta_i$ are randomly chosen in the intervals $(0.96, 1.04)$ and $(-0.2, 0.2)$, respectively. Average communication delays $\bar{\delta}^{j,i}$ have been chosen to be 0.1, while $\{\eta(k)\}$ and $\{\xi(k)\}$ have been simulated as zero-mean Gaussian white noise sequences with specified standard deviation $\sigma$. It has been adopted that $\zeta' = \zeta'' = 0.99$ and that the communication dropouts occur according to the probability $p_{ij} = 0.9$.

Typical behavior of the corrected drifts generated by $AlgDrift.a$ $(L = 1$ and $L = 100)$, $AlgDrift.b$ $(\nu = \frac{1}{2})$ and $AlgDrift.c$ $(l_0 = 0)$ in the presence of stochastic delays and measurement noise with $\sigma = 0.05$ is presented in Fig. 1 for a network with ten nodes. Convergence to consensus can be clearly

![Graphs showing corrected drifts for different algorithms and parameters](attachment:graphs.png)

**Figure 1**: Corrected drifts: in spite of the theoretically proved asymptotic superiority of $AlgDrift.b$, $AlgDrift.a$ with $L = 100$ practically achieves the best rate of convergence to consensus and noise immunity.
Figure 2: Corrected offsets: \( \text{AlgOffset.b} \), which includes consensus iterations on \( \hat{c}(k) \), has lower dispersion of the converged estimates than \( \text{AlgOffset.a} \). Corrected offsets do not converge if we set \( T_i = 0 \) (c) or \( c_i = 0 \) (d), \( i = 1, \ldots, n \), illustrating the importance of the introduced modifications in the error function \( (10) \).

observed in all cases. Analogous schemes from the literature (e.g., [Tian et al., 2016]) cannot achieve such a performance. The algorithm proposed in (Schenato and Fiorentin, 2011) is very sensitive to noise and practically inapplicable under the given conditions, while the algorithm from (Tian, 2015) achieves results similar to the ones obtained by \( \text{AlgDrift.c} \), but with typically lower convergence rate. It should be noticed that the best results are achieved by \( \text{AlgDrift.a} \) with \( L = 100 \); \( \text{AlgDrift.b} \) is practically inferior on finite intervals, in spite of the asymptotic results from Theorem 2. This indicates that the best choice of drift estimation algorithm should be in practice connected to \( \text{AlgDrift.a} \) with a suitably selected \( L \); it represents the best compromise between the signal to noise ratio and computational burden.
Typical behavior of the proposed offset correction algorithms \textit{AlgOffset.a} and \textit{AlgOffset.b} is illustrated in Fig. 2 (a) and (b); \textit{AlgDrift.a} with $L = 100$ has been used for drift correction. Convergence of all the components of the vector $\hat{f}(k)$ is evident in both cases. The algorithm \textit{AlgOffset.b} provides a lower dispersion of the asymptotic values of the corrected offsets, as expected. Fig. 2 (c) and (d) illustrate the importance of introducing $T_j(\cdot)$ and $T_i(\cdot)$, given by (11), and the delay correction parameter vector $c_i$ in the definition of $\varphi^b_i(\cdot)$ in (10), respectively. Fig. 2 (c) corresponds to $T_j(\cdot) = T_i(\cdot) = 0$, and Fig. 2 (d) to $c_i = 0$. It is evident that the offset estimates diverge in both cases. Introduction of $T_j(\cdot), T_i(\cdot)$ and $c_i$ appears to be essential for obtaining convergence of the corrected offset estimates.

In order to provide an insight into scalability of the proposed scheme, Fig. 3 contains the mean square disagreement (the squared error between the local corrected drifts averaged over the number of nodes) for networks with 10, 20, 50 and 100 nodes, as indicated in the figure. Obviously, the network radius does not deteriorate the performance significantly, proving high scalability.

Fig. 4 illustrates the rate of convergence to a common virtual clock (see Subsection 3.5): it represents the mean square disagreement multiplied by
Figure 4: Rate of convergence to a common virtual clock: the mean square disagreement multiplied by $k^{2\rho}$, for $\rho = 1, 1.1, 1.2$ (for AlgDrift.b) and $\rho = 1$ (for AlgDrift.c).

$k^{2\rho}$ where the exponent $\rho$ has been chosen to be 1, 1.1 and 1.2 for AlgDrift.b, and $\rho = 1$ for AlgDrift.c, as indicated in the figure. The curve corresponding to AlgDrift.c is not capable to ensure convergence to a common virtual clock.

5 Conclusion

In this paper a new distributed asynchronous algorithm have been proposed for time synchronization in networks with random communication delays, measurement noise and communication dropouts. A new algorithm is proposed for drift correction parameter estimation, based on an error function derived from specially defined local time increments. It has been proved, using the stochastic approximation arguments, that this algorithm achieves asymptotic consensus of the corrected drifts in the mean square sense and w.p.1 under general conditions concerning network properties. It is important that the algorithm achieves convergence rate superior to all similar schemes, especially in view of convergence to a virtual global clock. For offset estimation a new algorithm has been proposed starting from local time differences and adding: 1) special terms that take care of the influence of increasing time due to drift estimates which is an inevitable input to the algorithm, and 2) compensation parameters that take care of communication
delays. It has been proved that the corrected offsets converge in the mean square sense and w.p.1 to finite random variables. An efficient algorithm for practical applications based on introducing consensus on compensation parameters has also been proposed. It has been also shown that the proposed algorithms can be used as flooding algorithm with one reference node. Simulation results provide illustrations of the presented theoretical results.

A Proof of Lemma 1

According to (A2), \( p_i > 0, i = 1, \ldots, n \). By Lemma 3 in (Nedić, 2011),
\[
\left| \frac{1}{\nu_i(k)} - \frac{1}{kp_i} \right| \leq \kappa \frac{1}{k^\frac{1}{2} - q} \quad \text{w.p.1, } \kappa > 0, \text{ for } k \text{ large enough.}
\]
Therefore, there exists \( \tilde{\varepsilon}_i > 0 \) such that
\[
\left| \frac{1}{\nu_i(k)^\xi} - \frac{1}{(kp_i)^\xi} \right| \leq \tilde{\varepsilon}_i \frac{1}{k^{\xi + \frac{1}{2} - q}} \quad \text{w.p.1.}
\]
The result of Lemma 1 follows after taking into account that \( k \sim \tilde{k}\bar{N} \) for \( \tilde{k} \) large enough. Formally, \( E\{N(j)|j\} \), the average number of updates for a broadcast from node \( j \), can be obtained from the transmission probabilities \( p_{ij} \), while \( \bar{N} = \sum_{j=1}^{n} \pi_j E\{N(j)|j\} \), where \( \pi_j \) is the unconditional probability of node \( j \) to broadcast.

B Proof of Lemma 3

The result follows simply from the basic properties of the Poisson processes and the definition of the iteration number \( k \).

C Proof of Theorem 1

Introduce Lyapunov functions
\[
V^g(k) = E\{(\tilde{g}(k)^{[1]}^2\}
\]
and
\[
W^g(k) = E\{\tilde{g}(k)^{[2]^T} R^g \tilde{g}(k)^{[2]}\},
\]
where \( R^g > 0 \) satisfies (26) for a given \( Q^g > 0 \).

In order to obtain an estimate of \( V^g(k) \), we decompose \( \tilde{g}(k+1)^{[1]} \) from (27) into the sum of zero input and zero state responses, defined by
\[
\tilde{g}_1(k+1)^{[1]} = \Pi(k,1)^{[1]} \tilde{g}(1)^{[1]}
\]
(35)
\[ \bar{g}_2(k+1)^{[1]} = \sum_{\sigma=1}^{k} \frac{1}{\sigma^q} \Pi(k, \sigma + 1)^{[1]} [F_1(\sigma) \Delta t(\sigma) + H_1(\sigma)^{[2]}] \bar{g}(\sigma)^{[2]}, \quad (36) \]

respectively, where \( \Pi(k, l)^{[1]} = \prod_{\sigma=l}^{k} (1 + \frac{1}{\sigma^q} H_1(\sigma)^{[1]}), \Pi(k, k + 1)^{[1]} = 1, \) and \( H_1(k)^{[1]} \) follows from the decomposition \( H_1(k) = [H_1(k)^{[1]}_1; H_1(k)^{[2]}_1]. \) Therefore, \( V_g(k) \leq 2V_2^a(k) + 2V_2^b(k), \) where \( V_2^a(k) = E\{(\bar{g}_1(k)^{[1]}_1)^2\} \) and \( V_2^b(k) = E\{\bar{g}_2(k)^{[1]}_2)^2\}. \)

Introduce \( \sum_{i=1}^{n} |N_i^-| \) infinite subsequences \( \{\kappa^{ij}(v)\} \) of the set of nonnegative integers \( \mathbb{I}^+, i = 1, \ldots, n, j \in N_i^-, v = 0, 1, 2, \ldots, \) in which \( \kappa^{ij}(v) \) for a given \( v \) defines an instant \( k \) corresponding to an update at node \( i \) realized as a consequence of a tick of node \( j \) \( \kappa^{ij}(v_1) < \kappa^{ij}(v_2) \) for \( v_1 < v_2 \) and \( \cup_{i,j}\{\kappa^{ij}(v)\} = \mathbb{I}^+ \). Define \( \Pi(k, 1)^{[1]}_s = \prod_{\sigma \in \{\kappa^{ij}(v)\}, \sigma \leq k} (1 + \frac{1}{\sigma^q} H_1(\sigma)^{[1]}), \) \( s = 1, \ldots, n \), \( \sum_{i} |N_i^-|, i = 1, \ldots, n, j \in N_i^-, \) so that \( \prod_s \Pi(k, 1)^{[1]}_s = \Pi(k, 1)^{[1]} \).

According to the definition of \( \mathcal{N}^g(k) \), for AlgDrift.a and AlgDrift.b, the zero mean random sequences \( \{H_1(\sigma)^{[1]}\}, \sigma \in \{\kappa^{ij}(v)\} \), have the property that \( \{H_1(\sigma)^{[1]}\}_{\sigma = \{\kappa^{ij}(v)\}} \) is correlated only with \( \{H_1(\sigma)^{[1]}\}_{\sigma = \{\kappa^{ij}(v-1)\}} \) and \( \{H_1(\sigma)^{[1]}\}_{\sigma = \{\kappa^{ij}(v+1)\}} \). Therefore, it follows that \( E\{(\Pi(k, 1)^{[1]}_s)^2\} < \infty \), since \( \Pi(k, 1)^{[1]}_s \) are mutually independent. For AlgDrift.c, we have that \( H_1(\sigma)^{[1]} - \bar{H}_1(1)^{[1]} - \bar{H}_1(0)^{[1]} \), where \( \bar{H}_1(\sigma)^{[1]} \) is zero mean i.i.d. and \( \bar{H}_1(\sigma_0)^{[1]} \) a finite w.p.1 random variable \( (\sigma_0 = \kappa^{ij}(0)) \). Therefore, we have

\[ E\{(1 - \frac{1}{\sigma_1 + \zeta_\sigma} H_1(\sigma)^{[2]} | \mathcal{F}_\sigma_0 \} \leq 1 - c_1 \frac{1}{\sigma_1 + \zeta_\sigma} + c_2 \frac{1}{\sigma_2(1 + \zeta_\sigma)}, \quad (37) \]

where \( \mathcal{F}_\sigma_0 \) is the minimal sigma algebra generated by the measurements up to \( \sigma_0 \). It follows that \( E\{(\Pi(k, 1)^{[1]}_s)^2\} \leq \infty \). Therefore, we obtain that \( \sup_k V_1^g(k) < \infty \) for all three algorithms.

Estimation of \( V_2^b(k) \) for AlgDrift.a and AlgDrift.b starts from decomposing the sum at the right hand side of \( (36) \) into \( \sum_{i=1}^{n} |N_i^-| \) sums with indices \( \sigma \) belonging to \( \{\kappa^{ij}(v)\}, \sigma \leq k \). All these sums contain weighted mean random variables \( F_1(\sigma) \Delta t(\sigma) + H_1(\sigma)^{[2]} \); their correlation with \( \bar{g}(\sigma) \) is w.p.1 of the order of magnitude of \( \frac{1}{\sigma^q} \), so that it can be neglected for \( k \) large enough w.r.t. the corresponding terms in the expression for \( V_2^b(k) \). It is important to notice that \( \sum_{\sigma} E\{\frac{1}{\sigma^q} F_1(\sigma)^2 \Delta t(\sigma)^2\} \leq \infty \) for AlgDrift.a and AlgDrift.b, by virtue of Lemma 3. Noticing also that \( \sup_k \Pi(k, 1)^{[1]}_s \leq \infty \), it follows, after straightforward technicalities, that

\[ V_2^g(k+1) \leq C_1 \sum_{\sigma=1}^{k} \frac{1}{\sigma^{1+q'}} W^g(\sigma), \quad (38) \]
where $C_1 > 0$ and $q'' > 0$. For AlgDrift.c, the sum at the right hand side of (36) contains the terms $H_1(\sigma)[2] = H_1(\sigma) - \tilde{H}_1(\sigma_0)^2$, $\sigma \in \kappa_1(v)$. Having in mind that $\{\tilde{H}_1(\sigma)[2]\}$ is zero mean and $\tilde{H}_1(\sigma_0)[2]$ is bounded w.p.1, for AlgDrift.c $\sum_\sigma \frac{1}{\sigma \zeta} E\{\Delta t(k)^2\} < \infty$ by Lemma 3 and that $\sum_\sigma \frac{1}{\sigma \zeta} < \infty$, we obtain (38).

Consequently,

$$V^g(k + 1) \leq C_2[1 + \max_{1 \leq \sigma \leq k} W^g(\sigma)],$$

where $C_2 > 0$, having in mind that $\sum_{k=1}^\infty \frac{1}{\sigma + q''} < \infty$.

Estimation of $W^g(k)$ is based on considering the recursion (20) as a set of recursions on the sets $\{\kappa_i^j(v)\}, i = 1, \ldots, n, j \in \mathcal{N}_i^-$. We rewrite (28) for $\{\sigma \in \kappa_1(v)\}$ in the following way

$$\tilde{g}(\sigma + 1)^2 = \Pi(\sigma, \sigma)^2\tilde{g}(\sigma)^2 + \frac{1}{\sigma \zeta} H_2(\sigma)[1] \tilde{g}(\sigma)[1],$$

where $\Pi(\sigma, \sigma) = I + \frac{1}{\sigma \zeta}[(B^* + F_2(\sigma))\Delta t(\sigma) + H_2(\sigma)[2]]$, while $H_2(\sigma)[1]$ and $H_2(\sigma)[2]$ follow from the decomposition $H_2(\sigma) = [H_2(\sigma)[1];H_2(\sigma)[2]]$. We start the analysis by observing that for any $n$-vector $x$ and any $\sigma$

$$x^T E\{\Pi(\sigma, \sigma)^2 R^g \Pi(\sigma, \sigma)[2]\} x \leq [1 - \frac{2}{\sigma \zeta} q \frac{\lambda_{\min}(Q^g)}{\lambda_{\max}(R^g)} + O\left(\frac{1}{\sigma^{2\zeta}}\right)] x^T R^g x,$$

where $0 < \lambda_{\min}(Q^g), \lambda_{\max}(R^g) < \infty$ and $q = \frac{L}{\max_{i,j}(\mu_{ij} p_{ij})}$ for AlgDrift.a, $q = \frac{1}{\mu}$ for AlgDrift.b and $q = \frac{1}{\mu}$ for AlgDrift.c. Because $q > 0$ (Lemma 3), after standard technicalities based on the classical results on stochastic approximation [Chen, 2002] [Kushner and Yin, 2003], it follows that

$$\prod_{\sigma \in \{\kappa_1(v)\}} \|\Pi(\sigma, \sigma)\| \to_{\sigma \to \infty} 0,$$

for $i = 1, \ldots, n, j \in \mathcal{N}_i^-$, in the mean square sense and w.p.1, for AlgDrift.a, AlgDrift.b and AlgDrift.c. Moreover, as $\{H_2(\sigma)[1]\}$ has the properties analogous to those of $\{H_1(\sigma)[1]\}$, we have for $k$ large enough

$$W^g(\sigma + 1) \leq [1 - c_1 \frac{1}{\sigma \zeta}] W^g(\sigma) + C_3 \frac{1}{\sigma \zeta} V^g(\sigma),$$

where $0 < c_1, C_3 < \infty$. 

27
Using the methodology of [Huang and Manton, 2010; Huang et al., 2010] we can, consequently, show that \( \sup_k V^g(k) < \infty \). This gives rise to the conclusion that \( \tilde{g}(k)^{[1]} \) tends to a random variable \( \chi^* (E\{\chi^*^2\} < \infty) \) and \( \tilde{g}(k)^{[2]} \) to zero in the mean square sense and w.p.1. Consequently

\[
\hat{g}_\infty = T \left[ \lim_{k \to \infty} \tilde{g}(k)^{[1]} \right] = \chi^* 1, \tag{43}
\]

which proves the theorem.

**D Proof of Theorem 2**

After introducing the expression for \( z(k) \) into (28), we use the approximation \((1 + \frac{1}{k})^\zeta_d \approx 1 + \zeta_d \frac{1}{k}\) and obtain, after neglecting higher order terms, that for \( k \) large enough

\[
z(k + 1) = z(k) + \frac{1}{k^\zeta} [\tilde{B} + F_2(k)] \Delta t(k) + \zeta d \frac{1}{k^\zeta} z(k) + \frac{1}{k^\zeta(1-d)} H_2(k) \tilde{g}(k).
\]

Applying the methodology of the proof of Theorem 1 to (44), we observe that for \( \zeta' < 1 \) the term proportional to \( \frac{1}{k^\zeta} \), introduced by the formulation of the recursion for \( z(k) \), can be neglected for \( k \) large enough with respect to the term proportional to \( \frac{1}{k^\zeta} \). As \( \tilde{B} \) is Hurwitz, \( \lim_{k \to \infty} z(k) = 0 \) in the mean square sense and w.p.1 provided: a) \( 2\zeta'(1-d) > 1 \) for AlgDrift.a, b) \( 2(1 + \zeta')(1-d) > 1 \) for AlgDrift.b and c) \( (1 + \zeta')(1-d) > \zeta' \) for AlgDrift.c, wherefrom the first part of the result follows. Notice that different conditions result from different definitions of \( \zeta \) and the properties of the corresponding sequence \( \{H_2(k)\} \). For \( \zeta' = 1 \), the terms proportional to \( \frac{1}{k^\zeta} \) and \( \frac{1}{k^\zeta} \) are of the same order of magnitude; as a result, the convergence conditions for (44) depend on the properties of the matrix \( \tilde{B} \). Hence the result follows.

**E Proof of Theorem 3**

Let \( \hat{h}(k) = [(\hat{f}(k) + \chi(k) \tilde{A} A^{-1} \mathbf{1})^T; (\hat{c}(k) - \chi(k) A(\eta^0 + \delta))^T]^T \). As in the proof of Theorem 1 we obtain from (21), (22) and (30), after applying Lemma 1
that
\[ \hat{h}(k+1) = \hat{h}(k) + \frac{1}{k\gamma} P_d^{-\zeta''} [M_1(k) \hat{h}(k) + u_1(k) + u_2(k)] + M_2(k) \hat{G}(k) - \frac{1}{k\gamma} M_3(k) \hat{g}(k), \]  

(45)

where
\[ u_1(k) = o(\frac{1}{k\gamma}) [(A^{-1} \bar{\zeta}^0)^T (A(\bar{\eta}^0 + \bar{\delta}))] \],
\[ u_2(k) = [(\hat{g}_d(k) A^{-1} \bar{\zeta}^0(k)) \hat{G}(k) + A(\bar{\eta}^0(k) + \bar{\delta}(k))] \hat{g}(k)] \]

\[ M_1(k) = M_1 + \hat{M}_1(k), \]
\[ M_2(k) = M_2 + \hat{M}_2(k), \]
\[ M_2 = \text{diag}\{ \bar{\Gamma} \}, \hat{M}_2(k) = \text{diag}\{ \bar{\Gamma} \} + \text{diag}\{ \bar{\Gamma} \} \]

\[ \hat{g}_d(k) = \text{diag} \hat{g}(k), \hat{G}(k) = [\hat{g}(k)^T \hat{g}(k)^T]^T \text{ and } P_d^{-\zeta''} = \text{diag}\{ P^{-\zeta''}, P^{-\zeta''} \}; \]

the last term in (45) follows from the term \( \Delta \hat{g}(k+1) = e^\alpha(k)[A \Gamma(k) \Delta t(k) + N_g(k)] \hat{g}(k) \) in (48) and Lemma 1 so that \( M_3(k) = \begin{bmatrix} P^{-\zeta} \text{[A} \Gamma(k) \Delta t(k) + N_g(k)] \end{bmatrix} \).

From (45) we realize that \( P_d^{-\zeta''} \hat{M}_1 \) has \( n \) eigenvalues at the origin and \( n \) eigenvalues in the left half plane. Therefore, there exists a nonsingular transformation \( S \) such that
\[ S^{-1} P_d^{-\zeta''} \hat{M}_1 S = \begin{bmatrix} 0; 0; \hat{M}^* \end{bmatrix}, \]

(46)

where \( \hat{M}^* \) is Hurwitz (Stanković et al., 2015). Introduce \( \tilde{h}(k) = S^{-1} \hat{h}(k) \), with \( \tilde{h}(k) = [\tilde{h}(k)^{[1]} \tilde{h}(k)^{[2]}]^T \), where \( \text{dim} \tilde{h}(k)^{[1]} = \text{dim} \tilde{h}(k)^{[2]} = n \). Like in Theorem 1 we obtain from (45) the following two recursions:
\[ \tilde{h}(k+1)^{[1]} = \tilde{h}(k)^{[1]} + \frac{1}{k\sigma} \{ \Psi(k) \tilde{h}(k)^{[1]} + p(k)^{[1]} + q(k)^{[1]} + r(k)^{[1]} \} \]
\[ \tilde{h}(k+1)^{[2]} = \tilde{h}(k)^{[2]} + \frac{1}{k\sigma} \{ \hat{M}^* \tilde{h}(k)^{[2]} + \Psi(k) \tilde{h}(k)^{[2]} + p(k)^{[2]} + q(k)^{[2]} + r(k)^{[2]} \}, \]

(47)

(48)

where
\[ S^{-1} P_d^{-\zeta''} \hat{M}_1(k) S = \begin{bmatrix} \Psi(k)^{[1]} \Psi(k)^{[2]} \end{bmatrix}, \]

29
where \( \Phi(E) \) of (47) and (48) to the inputs \( q_k \) and \( \tilde{q}_k \), we obtain

\[
S^{-1}P^{-1}E^T [\bar{M}_1(k)u_1(k) + M_1(k)u_2(k) + \bar{M}_2(k)\tilde{g}(k)] = \begin{bmatrix} \bar{p}(k)[1] \\ p(k)[2] \end{bmatrix},
\]

(49)

\[
S^{-1}P^{-1}E^T [\bar{M}_1(k)u_1(k) + \bar{M}_2\tilde{g}(k)] = \begin{bmatrix} q(k)[1] \\ q(k)[2] \end{bmatrix},
\]

\[
-S^{-1}M_3(k)\tilde{g}(k) = \begin{bmatrix} r(k)[1] \\ r(k)[2] \end{bmatrix}.
\]

We introduce two main Lyapunov functions \( V^h(k) = E\{\|\tilde{h}(k)[1]\|^2\} \) and \( W^h(k) = E\{\tilde{h}(k)[2]^TR^h\tilde{h}(k)[2]\} \), where \( R^h > 0 \) satisfies the Lyapunov equation \( R^hM^* + M^*R^h = -Q^h \), for any given \( Q^h > 0 \) (according to (46)).

At the first step, we set \( q(k)[1] = 0 \) and \( q(k)[2] = 0 \) and denote the corresponding solutions of (47) and (48) by \( \tilde{h}_1(k)[1] \) and \( \tilde{h}_1(k)[2] \), respectively. Then, we introduce

\[
V^h_1(k) = E\{\|\tilde{h}_1(k)[1]\|^2\}
\]

and

\[
W^h_1(k) = E\{\tilde{h}_1(k)[2]^TR^h\tilde{h}_1(k)[2]\}.
\]

It is straightforward to see that the results from (Huang and Manton, 2010) can be directly applied to (47) and (48) (Theorem 11 and Lemma 12 therein), leading to the conclusion that \( \sup_k V^h_1(k) < \infty \) and that \( W^h_1(k) \) tends to zero when \( k \to \infty \). It is essential for this conclusion that the sequences \( \{\Psi(k)[1]\} \), \( \{\Psi(k)[2]\} \), \( \{p(k)[1]\} \) and \( \{p(k)[2]\} \) are uncorrelated, that \( \sum_{k=1}^{\infty} \frac{1}{k^{2\alpha\eta}} < \infty \), and that \( \lambda_{\min}(Q^h) > 0 \).

At the second step, consider the zero state responses \( \tilde{h}_2(k)[1] \) and \( \tilde{h}_2(k)[2] \) of (47) and (48) to the inputs \( q(k)[1] \) and \( q(k)[2] \), respectively. Let \( V^h_2(k) = E\{\|h_2(k)[1]\|^2\} \) and \( W^h_2(k) = E\{\tilde{h}_2(k)[2]^TR^h\tilde{h}_2(k)[2]\} \). By (29) we first conclude that \( M_2\tilde{g}(k) = M_2\tilde{g}(k)[2] \) (having in mind Theorem 2 and (29)). From (47), we obtain

\[
\tilde{h}_2(k+1)[1] = [I + \frac{1}{k^{2\alpha\eta}}\Psi(k)[1]]\tilde{h}_2(k)[1] + \frac{1}{k^{2\alpha\eta}}q(k)[1],
\]

(50)

where \( \Psi(k)[1] \) is an \( (n \times n) \) submatrix of \( \Psi(k)[1] \). From (50) we have that

\[
E\{\tilde{h}_2(k+1)[1]\} = E\{\tilde{h}_2(k)[1]\} + \frac{1}{k^{2\alpha\eta}}q(k)[1];
\]

consequently,

\[
V^h_2(k+1) < (1 + c) V^h_2(k) + \left(\frac{1}{k^{2\alpha\eta}}q(k)[1]\right)^2 + E\{\tilde{h}_2(k)[1]\} \frac{1}{k^{2\alpha\eta}}q(k)[1],
\]

(51)
(\epsilon' < \infty). Since \(q(k)[1] = o\left(\frac{1}{k^{c}}\right)\) w.p.1, by Theorem \([2]\), we can derive that 
\[ \sup_{k} E\{h_2(k)[1]\}^2 < \infty. \]
Consequently, \(\sup_{k} V_2^h(k) < \infty\) for all \(\zeta'' > 1 - \zeta d\), because of the requirement that \(\sum_{k} \frac{1}{k^{c}} \frac{1}{k^{c}} < \infty\). According to Theorem \([2]\), when \(\zeta' < 1\), this result holds for all \(\zeta'' \in \left(\frac{1}{2}, 1\right]\) in the case of AlgDrift.b and AlgDrift.c and for \(\zeta'' > \frac{3}{2} - \zeta'\) in the case of AlgDrift.a. Analysis of \(h_2(k)[2]\) relies on classical results from stochastic approximation (Chen, 2002), wherefrom we obtain that \(\lim_{k \to \infty} W_2^h(k) = 0\).

At the third step, consider the zero state responses \(\tilde{h}_3(k)[1]\) and \(\tilde{h}_3(k)[2]\) of (47) and (48) to the inputs \(r(k)[1]\) and \(r(k)[2]\), respectively. Let \(V_3^h(k) = E\{\|h_3(k)[1]\|^2\}\) and \(W_3^h(k) = E\{h_3(k)[2]^T R h_3(k)[2]\}\). Let \(r_1(k)[1]\) be the part of \(r(k)[1]\) following from \(e^D(k) A\Gamma(k) \Delta t(k) \hat{g}(k)\), and \(r_2(k)[1]\) the part following from \(e^D(k) N_{y}(k) \hat{g}(k)\). Taking into account (29), one concludes that \(r_1(k)[1] \sim o\left(\frac{1}{k^{c'}}\right)\) and that \(r_2(k)[1]\) is a zero mean i.i.d. term on any subsequence \(k^{\nu'}\), multiplied by \(\frac{1}{k^{c'}}\). Therefore, \(V_3^h(k) < \infty\), provided \(\zeta' + \zeta d > 1\); this is fulfilled in the case of AlgDrift.a for \(\zeta' > \frac{3}{2}\), and for any \(\zeta' \in \left(\frac{3}{2}, 1\right]\) in the case of AlgDrift.b and AlgDrift.c. Reasoning similarly, we conclude that \(\lim_{k \to \infty} W_3^h(k) = 0\) under less restrictive conditions.

Therefore, we conclude that \(\sup_{k} V_3^h(k) < \infty\) and \(\lim_{k \to \infty} W_3^h(k) = 0\). Using the arguments exposed in (Huang and Manton, 2010), we further derive that \(\hat{h}(k)[1]\) tends to a random n-vector \(\hat{h}[1]^\ast\), and that \(h(k)[2]\) tends to zero in the mean square sense and w.p.1, implying that \(\hat{h}^\ast = \lim_{k \to \infty} \hat{h}(k) = S \tilde{h}^\ast\), where \(\hat{h}^\ast = [\hat{h}[1]^T S = [0; 0; \vdots; 0; 0)]^T\). The result of the theorem follows after taking into account that \(\chi(k) \to \chi^\ast\) w.p.1 and that

\[
\tilde{M}_1\tilde{h}^\ast = \tilde{M}_1S\tilde{h}^\ast = P_3^\ast S \begin{bmatrix} 0; 0; \cdots; 0; M^\ast \end{bmatrix} \tilde{h}^\ast = 0,
\]

according to (46) the definition of \(\hat{h}^\ast\).

\section*{F Proof of Theorem \([4]\)}

We shall pay attention only to the possible convergence points: the rest can be derived by following methodologically the proof of Theorem \([3]\). Namely, according to (Kushner and Yin, 1987), we formulate the ODE characterizing the asymptotic behavior of the algorithm, and obtain that

\[
\tilde{\Gamma} \tilde{J}^\ast - \chi^\ast [\tilde{\Gamma}_d(\delta + \eta^0) - \tilde{\Gamma}_d \xi^0] + \tilde{\Gamma}_d \tilde{\chi}^\ast \tilde{\omega}^{\text{cor}} = 0
\]

\[
\sum_{i=1}^{n} \phi_i \{\tilde{\Gamma}(i) \tilde{J}^\ast - \chi^\ast [\tilde{\Gamma}_d(\delta + \eta^0)_i - \tilde{\Gamma}_d \xi^0)_i] + (\tilde{\Gamma}_d 1)_i \tilde{\omega}^{\text{cor}}\} = 0,
\]

(52)
wherefrom the result directly follows.

References

Akyildiz, I. F. and Vuran, M. C. (2010). *Wireless sensor networks*. John Wiley & Sons.

Aysal, T. C., Yildiz, M. E., Sarwate, A. D., and Scaglione, A. (2009). Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Process.*, 57:2748–2761.

Bolognani, S., Carli, R., Lovisari, E., and Zampieri, S. (2012). A randomized linear algorithm for clock synchronization in multi agent systems. In *Proc. IEEE Conf. Decision and Control*, pages 20–25.

Carli, R., Chiuso, A., Zampieri, S., and Schenato, L. (2008). A PI consensus controller for networked clock synchronization. In *Proc. IFAC World Congress*.

Chaudhari, Q. M., Serpedin, E., and Qaraqe, K. (2008). On maximum likelihood estimation of clock offset and skew in networks with exponential delays. *IEEE Trans. Signal Process.*, 56:1685–1697.

Chen, H. F. (2002). *Stochastic approximation and its applications*. Kluwer Academic, Dordrecht, the Netherlands.

Choi, B. J., Liang, H., Shen, X., and Zhuang, W. (2012). DCS : distributed asynchronous clock synchronization in delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.*, 23:491–504.

Elson, J., Girod, L., and Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. In *Proc. Symp. Oper. Syst. Design and Implement.*, pages 147–163.

Fan, R. and Lynch, N. (2006). Gradient clock synchronization. *Distrib. Comput.*, 18:255–266.

Freris, N. M., Graham, S. R., and Kumar, P. R. (2011). Fundamental limits on synchronizing clocks over networks. *IEEE Trans. Autom. Control*, 56:1352–1364.

Garone, E., Gasparri, A., and Lamonaca, F. (2015). Clock synchronization protocol for wireless sensor networks with bounded communication delays. *Automatica*, 59:60–72.
Gupta, P. and Kumar, P. R. (2006). The capacity of wireless networks. IEEE Trans. Inf. Theory, 46(2):388–404.

He, J., Cheng, P., Chen, J., Shi, L., and Lu, R. (2014a). Time synchronization for random mobile sensor networks. IEEE Trans. Veh. Technol., 63:3935–3946.

He, J., Cheng, P., Shi, L., Chen, J., and Sun, Y. (2014b). Time synchronization in WSNs: a maximum value based consensus approach. IEEE Trans. Autom. Control, 59:660–675.

Holler, J., Tsiatsis, V., Mulligan, C., Avesand, S., Karnouskos, S., and Boyle, D. (2014). From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence. Academic Press.

Huang, M., Day, S., Nair, G. N., and Manton, J. H. (2010). Stochastic consensus over noisy networks with Markovian and arbitrary switches. Automatica, 46:1571–1583.

Huang, M. and Manton, J. H. (2010). Stochastic consensus seeking with noisy and directed inter-agent communications: fixed and randomly varying topologies. IEEE Trans. Autom. Control, 55:235–241.

Kim, K.-D. and Kumar, P. R. (2012). Cyberphysical systems: A perspective at the centennial. Proc. IEEE, 100(Special Centennial Issue):1287–1308.

Kushner, H. J. and Yin, G. (1987). Asymptotic properties of distributed and communicating stochastic approximation algorithms. SIAM J. Control Optim., 25:1266–1290.

Kushner, H. J. and Yin, G. (2003). Stochastic Approximation and Recursive Algorithms and Applications. Springer.

Leng, M. and Wu, Y.-C. (2011). Low complexity maximum likelihood estimator for clock synchronization of wireless sensor nodes under exponential delays. IEEE Trans. Signal Process., 59:4860–4870.

Li, Q. and Rus, D. (2006). Global clock synchronization in sensor networks. IEEE Trans. Comput., 55:214–226.

Liao, C. and Barooah, P. (2013a). Distributed clock skew and offset estimation from relative measurements in mobile networks with Markovian switching topologies. Automatica, 49:3015–3022.
Liao, C. and Barooah, P. (2013b). Disync: accurate distributed clock synchronization in mobile ad–hoc networks from noisy difference measurements. In Proc. American Control Conference, pages 3332–3337.

Maroti, M., Kusy, B., Simon, G., and Ledeczi, A. (2004). The flooding time synchronization protocol. In Proc. 2nd ACM Intern. Conf. Embedded Netw. Contr. Syst., SenSys'04, pages 39–49.

Nedić, A. (2011). Asynchronous broadcast-based convex optimization over a network. IEEE Trans. Autom. Control, 56:1337–1351.

Olfati-Saber, R., Fax, A., and Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. Proc. IEEE, 95:215–233.

Schenato, L. and Fiorentin, F. (2011). Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks. Automatica, 47(9):1878–1886.

Simeone, O., Spagnolini, U., Bar-Ness, Y., and Strogatz, S. H. (2008). Distributed synchronization in wireless networks. IEEE Signal Process. Mag., 25:81–97.

Sivrikaya, F. and Yener, B. (2004). Time synchronization in sensor networks: a survey. IEEE Network, 18:45–50.

Solis, R., Borkar, V., and Kumar, P. R. (2006). A new distributed time synchronization protocol for multi hop wireless networks. In Proc. IEEE Conf. Decision and Control, pages 2734–2739.

Sommer, P. and Wattenhofer, R. (2009). Gradient clock synchronization in wireless sensor networks. In Proc. Int. Conf. Inf. Proc. in Sensor Networks, pages 37–48.

Stanković, M. S., Stanković, S. S., and Johansson, K. H. (2012). Distributed time synchronization in lossy wireless sensor networks. In 3rd IFAC Workshop on Distr. Estim. Contr. Netw. Syst., NECSYS, volume 3, pages 25–30.

Stanković, M. S., Stanković, S. S., and Johansson, K. H. (2015). Distributed blind calibration in lossy sensor networks via output synchronization. IEEE Trans. Autom. Control, 60:3257–3262.

Stanković, M. S., Stanković, S. S., and Johansson, K. H. (2016). Distributed drift estimation for time synchronization in lossy networks. In 24th
Su, W. and Akyildiz, I. F. (2005). Time diffusion synchronization protocol for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13:384–397.

Sundaraman, B., Buyand, U., and Kshemkalyani, A. (2005). Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3:281–323.

Tian, Y.-P. (2015). LSTS: a new time synchronization protocol for networks with random communication delays. In *Proc. IEEE Conf. Decision and Control*, pages 7404–7409.

Tian, Y.-P., Zong, S., and Cao, Q. (2016). Structural modeling and convergence analysis of consensus-based time synchronization algorithms over networks: non-topological conditions. *Automatica*, 65:64–75.

Wu, Y.-C., Chaudhari, Q., and Serpedin, E. (2011). Clock synchronization in wireless sensor networks. *IEEE Signal Process. Mag.*, pages 124–138.

Xiong, G. and Kishore, S. (2009). Analysis of distributed consensus time synchronization with Gaussian delay over wireless sensor networks. *EURASIP J. on Wireless Comm. and Netw.*, (1):1–9.

Yildirim, K. S., Carli, R., and Schenato, L. (2015). Adaptive control-based clock synchronization in wireless sensor networks. In *Proc. European Control Conference*. 