A New Drift Correction Algorithm for Distributed Time Synchronization

Milos S. Stankovic, Srdjan S. Stankovic and Karl Henrik Johansson

Abstract—This paper proposes a new distributed asynchronous algorithm for drift correction in time synchronization for networks with random communication delays, measurement noise and communication dropouts. Three variants of the algorithm are proposed, 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 of the corrected drifts of all the nodes to the same value (consensus). Asymptotic rate of convergence of the algorithms is formulated. It is shown that it is possible to achieve convergence to the common virtual clock. Simulation results give an illustration of the properties of the algorithms.

I. 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 [1], [2]. 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., [3]. There are numerous approaches to time synchronization starting from different assumptions and using different methodologies, e.g., [3], [4]. An important class of time synchronization algorithms is based on full distribution of functions [5], [6]. A class of consensus based algorithms, called CBTS (Consensus-Based Time Synchronization) algorithms has attracted considerable attention, e.g., [7]–[10]. It has been treated in a unified way in a recent survey [11], providing figure of merit of the principal approaches. In [12] 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 [13] for basic issues.

In this paper we propose a new asynchronous distributed algorithm for drift correction, used for time synchronization in lossy networks characterized by random communication delays, measurement noise and communication dropouts. The algorithm is composed of a distributed recursion of asynchronous stochastic approximation type based on broadcast gossip and derived from predefined local error function. The recursion is aimed at achieving asymptotic consensus on the corrected drifts and, together with an offset correction algorithm, at obtaining common virtual clock for all the nodes in the network.

The proposed recursion for drift synchronization (presented in a preliminary form in [14]) 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. Compared to the existing analogous algorithms [8], [10], the proposed scheme is structurally different and simpler and provides the best convergence rate. Notice that the algorithm proposed in [8] cannot handle communication delays or measurement noise, while the paper [10] treats random delays, but not the case of measurement noise and communication dropouts.

The proposed drift correction algorithm is a good prerequisite for achieving finite differences between local virtual clocks in the means 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.

Finally, two illustrative simulation results are presented.

II. DRIFT CORRECTION ALGORITHMS

A. Time and Network Models

Assume a network consisting of nodes, formally represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, 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 $i$ has a local clock, whose output, defining local time, is given for any absolute time $t \in \mathcal{R}$ by

$$\tau_i(t) = a_\tau t + b_\tau + \xi_i(t),$$

where $a_\tau$ is the local drift (gain), $b_\tau$ is the local offset, while $\xi_i(t)$ is measurement noise, appearing due to equipment instabilities, round-off errors, thermal noise, etc. [8], [15]. 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, $g_i = a_i a_\tau$ is the corrected drift and $f_i = a_i b_\tau + b_i$ the corrected offset.
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., [16], 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 $t_j^i$, $b = 0, 1, 2, \ldots$), node $j$ broadcasts its current local time (together with its current estimates of correction parameters) to its out-neighbors $i \in \mathcal{N}_j^+$. Each node $i \in \mathcal{N}_j^+$ hears the broadcast with probability $p_{ij} > 0$. Let $\{t_j^i\}, i = 0, 1, 2, \ldots$, be the sequence of absolute time instants corresponding to the messages heard by node $i$. The message sent at $t_j^i$ is received at node $i$ at the time instant

\[
t_j^i = t_j^i + \delta_j^i,
\]

where $\delta_j^i$ represents the corresponding communication delay. See [17] for presentation of physical and technical sources of the delays. We assume in the sequel that the communication delay can be decomposed as

\[
\delta_j^i = \bar{\delta}_j^i + \eta_j(t_j^i),
\]

where $\bar{\delta}_j^i$ is assumed to be constant (depending only on the chosen arc $(j, i)$), while $\eta_j(t_j^i)$ 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 [16].

B. Algorithm

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

\[
\bar{\psi}_j(t_j^i) = \Delta \tilde{\tau}_j(t_j^i) - \Delta \tilde{\xi}_j(t_j^i),
\]

where $\Delta \tilde{\tau}_j(t_j^i)$ and $\Delta \tilde{\xi}_j(t_j^i)$ are increments of the corrected local times, given by

\[
\Delta \tilde{\tau}_j(t_j^i) = \tilde{\tau}_j(t_j^i) - \tilde{\tau}_j(t_j^m),
\]

\[
\Delta \tilde{\xi}_j(t_j^i) = \tilde{\xi}_j(t_j^i) - \tilde{\xi}_j(t_j^m),
\]

where $m \in \{0, \ldots, i - 1\}$.

\[
\Delta \tilde{\tau}_j(t_j^i) = \tilde{\tau}_j(t_j^i) - \tilde{\tau}_j(t_j^m) = a_j \Delta \tau_j(t_j^i),
\]

\[
\Delta \tilde{\xi}_j(t_j^i) = \tilde{\xi}_j(t_j^i) - \tilde{\xi}_j(t_j^m) = a_i \Delta \xi_j(t_j^i),
\]

\[
\Delta \xi_j(t_j^i) = \xi_j(t_j^i) - \xi_j(t_j^m),
\]

\[
\Delta \tau_j(t_j^i) = \tau_j(t_j^i) - \tau_j(t_j^m).
\]

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):

1) $m = i - L$, where $L$ is a predefined constant (AlgDrift.a);
2) $m = \lfloor v \rceil (0 < v < 1)$, where $\lfloor x \rceil$ denotes the largest integer less than or equal to $x$ (AlgDrift.b);
3) $m = l_0$, 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. The case b), when we have both $\lim_{m \to \infty} m = \infty$ and $\lim_{m \to \infty} (l-m) = \infty$, and the case c) are conceptually very important (see Theorem 2 below).

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

\[
\tilde{a}_i(t_j^i) = \bar{a}_i(t_j^i) + \bar{\psi}_j(t_j^i) \bar{\psi}_j^a(t_j^i),
\]

where $\bar{\psi}_j^a(t_j^i)$ are a priori adopted nonnegative weights expressing relative importance of communication links (their role will be discussed below). $\bar{\psi}_j^a(t_j^i) = \Delta \tilde{\tau}_j(t_j^i) - \Delta \tilde{\xi}_j(t_j^i)$,

\[
\Delta \tilde{\tau}_j(t_j^i) = \Delta \tilde{\tau}_j(t_j^i)_{a_j = \tilde{a}_j(t_j^i)},
\]

\[
\Delta \tilde{\xi}_j(t_j^i) = \Delta \tilde{\xi}_j(t_j^i)_{a_i = \tilde{a}_j(t_j^i)}.
\]

$\bar{a}_j(t_j^i)$ and $\tilde{a}_j(t_j^i)$ are the old estimates, $\hat{a}_j(t_j^i)$ the new estimate, while $\varepsilon_j(t_j^i)$ is a positive step size. 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 $\bar{a}_i(t_0^i) = 1$.

In terms of the corrected drift $\bar{\xi}_j(. \bar{a}_i(.a_i$, (5) gives:

\[
\hat{a}_i(t_j^i + \Delta t_j^i) = \bar{a}_i(t_j^i) + \varepsilon_j(t_j^i) \bar{\psi}_j^a(t_j^i),
\]

where $\bar{\psi}_j^a(t_j^i) = \alpha_j \Delta \tilde{\tau}_j(t_j^i) - \Delta \tilde{\xi}_j(t_j^i)$, $\Delta \tilde{\tau}_j(t_j^i) = \tilde{\tau}_j(t_j^i) - \tilde{\tau}_j(t_j^m)$, $\Delta \tilde{\xi}_j(t_j^i) = \tilde{\xi}_j(t_j^i) - \tilde{\xi}_j(t_j^m)$, $\Delta \tau_j(t_j^i) = \tau_j(t_j^i) - \tau_j(t_j^m)$, $\Delta \xi_j(t_j^i) = \xi_j(t_j^i) - \xi_j(t_j^m)$, $\Delta \tilde{\tau}_j(t_j^i) = \tilde{\tau}_j(t_j^i) - \tilde{\tau}_j(t_j^m)$, $\Delta \tilde{\xi}_j(t_j^i) = \tilde{\xi}_j(t_j^i) - \tilde{\xi}_j(t_j^m)$, $\Delta \tilde{\xi}_j(t_j^i + 1) = \tilde{\xi}_j(t_j^i + 1) - \tilde{\xi}_j(t_j^i)$.

Remark 2: The algorithm does not belong to the class of the so-called CBTS algorithms [11]: it is structurally different and simpler, not requiring the step of relative drift estimation, which introduces unnecessary dynamics and additional non-linearities.

C. Global Model

Next we derive a global model of the whole 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., [16], [18]). Starting from this fact, a global model for the whole network has been defined in [14] 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 \( \hat{t}_{ij} \) for some \( j \) and \( i \)). 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), \( N(j) \leq \lfloor N_j \rfloor \). Following analogous approaches in [11], [16], we replace (with some abuse of notation) the variable \( \hat{t}_{ij} \) by \( k \) in all the above defined functions of time, so that we have \( \tau_k(\hat{t}_{ij}) = \tau_k(k), \tau_k(\hat{t}_{ij}) = \bar{\tau}_k(k), \bar{\xi}_k(\hat{t}_{ij}) = \bar{\xi}_k(k), \) etc.; accordingly, we also write \( \bar{\tau}_k(\hat{t}_{ij}) = \bar{\tau}_k(k), \tau_k(\hat{t}_{ij}) = \tau_k(k), \bar{\xi}_k(\hat{t}_{ij}) = \bar{\xi}_k(k), \) etc. In the case of delays, we write \( \hat{\delta}_k = \delta_k(k) \) and \( \eta_k(\hat{t}_{ij}) = \eta_k(k) \). Assume that \( k \) is connected to an update at node \( i \), initiated by a tick of node \( j \). Let \( \bar{g}(k) = [\bar{g}_1(k) \ldots \bar{g}_n(k)]^T \), \( \bar{f}(k) = [\bar{f}_1(k) \ldots \bar{f}_n(k)]^T \) and \( \bar{c}(k) = [\bar{c}_1(k) \ldots \bar{c}_n(k)]^T \), where \( \bar{g}_k(k) = \bar{a}_k(k) \alpha_k, \bar{a}_k(k) = \hat{a}_k(k), \bar{f}_k(k) = \bar{a}_k(k) \bar{B}_0 + \bar{b}_k(k), \bar{b}_k(k) = \bar{b}_k(k), \bar{c}_k(k) = \hat{c}_k(k), \bar{c}_k(k) = \hat{c}_k(k), \mu = 1, \ldots, n. \) Then, (8) gives
\[
\bar{g}(k+1) = \bar{g}(k) + e^\theta(k)Z(k)\bar{g}(k),
\]
where \( e^\theta(k) = [e_1^\theta(k) \ldots e_n^\theta(k)]^T \), \( e^\theta(k) = \text{diag}(e_1^\theta(k), \ldots, e_n^\theta(k)), e_i^\theta(k) = e_i(\hat{t}_{ij}) \) (see (5)),
\[
Z(k) = A(\Gamma(k)\Delta(k) + N(k)),
\]
\( \Delta = \text{diag}(\alpha_1, \ldots, \alpha_n), \Gamma(k) = [\Gamma(k)_{ij}], \) with \( \Gamma(k)_{ii} = -\gamma_i, \Gamma(k)_{ij} = \gamma_{ij} \) for \( \gamma_{ij} \neq 0, \Gamma(k)_{ij} = 0 \) otherwise, \( \Delta(k) = \bar{t}_{ij} - \bar{t}_{ij} \), while the noise term is defined as
\[
N(k) = -A\Gamma(k)\Delta(k)\Delta(k) + A(\Gamma(k)\Delta(k)\bar{\xi}_k(k)A)^{-1},
\]
where \( \Gamma_d(k) = \text{diag}(\text{diag}(\gamma_1, \ldots, \gamma_n)\omega(k)), \omega(k) = [\omega_1(k) \ldots \omega_n(k)]^T, \omega_1(k) = 1, \omega_k(k) = 0 \) for \( \mu \neq i, \Delta(k) = \Delta(\bar{g}_k(k) \ldots \Delta_k(k), \Delta_{\xi}(k) = \text{diag} \Delta(\bar{g}_k(k) \ldots \Delta_{\xi}(k)), \Delta_{\xi}(k) = [\Delta \xi(k) \ldots \Delta_{\xi}(k)]^T, \Delta_{\xi}(k) = \text{diag} \Delta(\bar{g}_k(k) \ldots \Delta_{\xi}(k)).
\]

### III. Convergence Analysis

#### A. Preliminaries

Within the exposed general setting, we additionally assume:

(A1) Graph \( \mathcal{G} \) has a spanning tree.

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

(A3) The step sizes \( e_i^\theta(k) \) and \( e_i^\theta(k) \) are defined in the following way:
\[
e_i^\theta(k) = e_i(k)|_{t = t^\theta}, \quad \text{for AlgDrift.a,}
\]
\[
e_i^\theta(k) = e_i(k)|_{t = t^\theta}, \quad \text{for AlgDrift.b and AlgDrift.c,}
\]
where \( e_i(k) = v_i(k), v_i(k) = \sum_{i=1}^n 1(\text{node } i \text{ received a message}) \), representing the number of updates of node \( i \) up to the instant \( k \) (\( I \{ \cdot \} \) denotes the indicator function), while \( \frac{1}{2} \leq e_i^\theta, e_i^\theta \leq 1.

**Remark 3:** (A1) implies that graph \( \mathcal{G} \) has a center node from which all the remaining nodes are reachable [19], [20]. (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 size for all the nodes as a function of \( k \). The choice of the exponent in the expression for \( e_i^\theta(k) \) 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).

Asymptotic behavior of the step size is given by the following lemma.

**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 \( \xi_0 > 0 \). Then, for a given \( q \in (0, \frac{1}{2}) \), there exists \( \bar{k} \) such that w.p.1 for all \( k \geq \bar{k}
\[
e_i(k) = \frac{1}{k^{\bar{k}}} e_i^\theta(k),
\]
where \( \bar{N} = E_i[\{N(j)\}] \) represents the average number of updates per one tick of the global virtual clock and \( |e_i(k)| \leq \bar{e}_i \), \( \bar{e}_i < 0, i = 1, \ldots, n. \)

Properties of the matrix \( \Gamma(k) \) defined in the previous section are essential for convergence of the algorithm; its expectation \( \bar{\Gamma} = E[\Gamma(k)] \) has a central role, 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 \( \mathcal{G} \):

\[
\bar{\Gamma} = \begin{bmatrix}
\gamma_{1} \pi_{11} & \gamma_{12} \pi_{12} & \cdots & \gamma_{1n} \pi_{1n} \\
\gamma_{21} \pi_{21} & \gamma_{22} \pi_{22} & \cdots & \gamma_{2n} \pi_{2n} \\
\vdots & \ddots & \cdots & \vdots \\
\gamma_{n1} \pi_{n1} & \gamma_{n2} \pi_{n2} & \cdots & \gamma_{nn} \pi_{nn}
\end{bmatrix},
\]
(11)
where \( \gamma_{ij} > 0 \) when \( j \notin \mathcal{N}_-(i) \), where \( \pi_{ij} \) is unconditional probability that the node \( j \) broadcasts and node \( i \) updates its parameters as a consequence (\( \pi_{ij} = \pi_i p_{ij} \)), where \( p_{ij} \) is the unconditional probability for node \( j \) to broadcast).

According to (9) and Lemma 1, we shall consider \( B(k) = P^\theta \bar{\Gamma}(k) \) and \( \bar{B} = E(B(k)) = P^\theta \bar{\Gamma}(k) \) \( P^\theta = \bar{N}^\theta \text{diag}(p_1^\theta, \ldots, p_n^\theta) \). The latter matrix has the following properties.

**Lemma 2:** [20] Matrix \( \bar{B} \) has one eigenvalue at the origin, and the remaining ones in the left half plane. Let \( T = [1 \cdots T_n(n-1)] \), where \( T_n(n-1) \) is such that span\( \{T_n(n-1)\} = \text{span}(\bar{B}) \), while \( 1 = [1 \ldots 1]^T \). Then,
\[
T^{-1}\bar{B}T = \begin{bmatrix}
0 & 0 & \cdots & 0 \\
0 & B_1 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & B_n
\end{bmatrix},
\]
where \( B^* \) is Hurwitz.

Consequently, there exists \( R^b > 0 \) satisfying
\[
R^b \bar{B} + \bar{B}^T R^b = -Q^b,
\]
for any given \( Q^b > 0 \). It also follows from the derivation of (10) that \( T^{-1}B(k)T = \begin{bmatrix}
0 & \bar{B}_1(k) \\
0 & \bar{B}_2(k)
\end{bmatrix} \), with \( \text{E}\{B_1(k)\} = 0 \) and \( \text{E}\{B_2(k)\} = \bar{B}^* \).
Lemma 3: $E\{\Delta_t(k)\} = \frac{1}{\mu_j}(l - m)$, $\text{var}\{\Delta_t(k)\} = \frac{1}{\mu_j^2}(l - m)$, where $l - m = L$ for AlgDrift.a, $l - m = \lceil (1 - v)l \rceil$ for AlgDrift.b and $l - m = l$ for AlgDrift.c; for large $l$, we have $l \sim \pi_h k$.

B. Convergence and Convergence Rate

After coming back to (9), we first insert $e^a(k)$ from (10). Then, we introduce $\tilde{g}(k) = T^{-1}g(k)$ and decompose $\tilde{g}(k)$ as $\tilde{g}(k) = [g(k)]^T \cdot \tilde{g}(k) = \tilde{g}_1(k)$ and $\tilde{g}(k) = \tilde{g}_2(k) \ldots \tilde{g}_n(k)]^T$. After neglecting the higher order terms from (10), we obtain

$$\begin{align*}
\tilde{g}(k + 1)^1 &= \tilde{g}(k)^1 + \frac{1}{k^2}F_1(k)\Delta_t(k)\tilde{g}(k)^2 \\
\tilde{g}(k + 1)^2 &= \{I + \frac{1}{k^2}[\tilde{B}^\ast + F_2(k)]\Delta_t(k)\}\tilde{g}(k)^2 + \frac{1}{k^2}H_2(k)\tilde{g}(k),
\end{align*}$$

where matrices $F_1(k)$ and $F_2(k)$ are defined by $T^{-1}[B(k) - \tilde{B}^\ast]T = \begin{bmatrix} 0 & F_1(k) \\ 0_{(n-1)\times 1} & F_2(k) \end{bmatrix}$, while $H_1(k)$ and $H_2(k)$ are defined by $T^{-1}P^{-c}N_g(k)T = \begin{bmatrix} H_1(k) \\ H_2(k) \end{bmatrix}$.

**Theorem 1:** Let assumptions (A1)–(A3) be satisfied. Then, $\tilde{g}(k)^1$ from (14) converges to a random variable $\chi^\ast$ with bounded second moment, and $\tilde{g}(k)^2$ from (15) to zero in the mean square sense and w.p.1; in other words, $\tilde{g}(k)$ generated by (9) converges for all three choices of $m$ to $\tilde{g}_\infty = \chi^\ast 1$ 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 (9) will be studied through the behavior of $\tilde{g}(k)^2$ in (15), using the methodology of [21, Chapter 3].

**Theorem 2:** Let (A1)–(A3) hold. Then, $\tilde{z}(k) = k\xi_d\tilde{g}(k)^2$, where $d > 0$ and $\tilde{g}(k)^2$ is defined by (15), converges to zero in the mean square sense and w.p.1, when $\xi^\ast < 1$ for:

1. $\xi_d < \xi^\ast - \frac{1}{2}$ (AlgDrift.a),
2. $\xi_d < \frac{1}{2} + \frac{1}{\xi^\ast}$ (AlgDrift.b) and $\xi_d < \frac{1}{2}$ (AlgDrift.c),

and when $\xi^\ast < 1$ for:

1. $d < \min\left\{ \frac{1}{4}, \frac{1}{2}qr \right\}$ (AlgDrift.a),
2. $d < \min\left\{ \frac{1}{4}, \frac{1}{2}qr \right\}$ (AlgDrift.b) and $d < \min\left\{ \frac{1}{4}, \frac{1}{2}qr \right\}$ (AlgDrift.c),

where $r = \frac{\max\{\eta(k)\}}{q}$, $q = \max\{v, h, p_j\}$ for AlgDrift.a, $q = \frac{1 - v}{\mu}$ for AlgDrift.b and $q = \frac{1}{\mu}$ for AlgDrift.c.

**Remark 4:** The results hold asymptotically, for $k$ large enough. They indicate that the AlgDrift.b provides the best results: in the case when $\xi^\ast < 1$ the condition $\xi_d > 1$ is achieved, enabling convergence to a common virtual clock. The requirements for $m$ 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 $[m,l]$. 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. Practically the best results can be obtained by AlgDrift.a for $L$ moderately high.

Notice that the CBTS algorithms discussed in [11] cannot achieve convergence rate $\xi_d > 1$, important for achieving convergence to a global virtual clock.

IV. 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 [22]. 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 $\tilde{d}_{ij}$ have been chosen to be 0.1, while $\{\tilde{\eta}(k)\}$ and $\{\tilde{\xi}(k)\}$ have been simulated as zero-mean Gaussian white noise sequences with specified standard deviation $\sigma$. It has been adopted that $\xi_d = 0.99$ and that the communication dropouts occur according to the probability $p_{ij} = 0.9$.

Typical behavior of the corrected drifts generated by AlgDrift.a ($L = 100$) and AlgDrift.b ($v = \frac{1}{2}$) in the presence of stochastic delays and measurement noise with $\sigma = 0.05$ is presented in Figs. 1 and 2 for a network with ten nodes. Convergence to consensus can be clearly observed in all cases. Analogous schemes from the literature (e.g., [11]) cannot achieve such a performance. It should be noticed that the best results are achieved by AlgDrift.a with $L = 100$; 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 AlgDrift.a with a suitably selected L large enough; it represents the best compromise for practice.

V. CONCLUSION

In this paper a new distributed asynchronous algorithm have been proposed for drift correction within time synchronization for networks with random communication delays, measurement noise and communication dropouts. A new algorithm is proposed based on an error function derived from 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. Also, the algorithm is substantially simpler than all the existing schemes.

VI. PROOF OF THEOREM 1

Introduce Lyapunov functions $V^g_{\sigma}(k) = E\{\tilde{g}(k)^2\}$ and $W^g_{\sigma}(k) = E\{\tilde{g}(k)^2 | R^g_{\sigma}(k)^2\}$, where $R^g_{\sigma} > 0$ satisfies (13) for a given $Q^g > 0$.

In order to obtain an estimate of $V^g_{\sigma}(k)$, we decompose $\tilde{g}(k+1)^{[1]}$ from (14) into the sum of zero input and zero state responses, defined by

$$\tilde{g}(k+1)^{[1]} = \Pi(k,1)^{[1]} \tilde{g}(1)^{[1]}$$

and

$$\tilde{g}(k+1)^{[1]} = \sum_{\sigma_1=1}^{k} \frac{1}{\sigma_1 \sigma_2} \Pi(k,\sigma_1 + 1)^{[1]} [F_{\sigma_2}(\sigma) \Delta(\sigma)$$

$$+ H_{\sigma_2}(\sigma)^{[1]} \tilde{g}(\sigma)^{[2]}]$$

respectively, where $\Pi(k,\sigma)^{[1]} = \prod_{\sigma_1=1}^{k} (1 + \frac{1}{\sigma_1 \sigma_2} H_{\sigma_2}(\sigma)^{[1]}), \Pi(k, k + 1)^{[1]} = 1$, and $H_{\sigma_2}(\sigma)^{[1]}$ follows from the decomposition $H_{\sigma_2}(\sigma) = [H_{\sigma_2}(\sigma)^{[1]} H_{\sigma_2}(\sigma)^{[2]}].$ Therefore,

$$V^g_{\sigma}(k) \leq 2V^g_{\sigma}(k) + 2V^g_{\sigma}(k),$$

where $V^g_{\sigma}(k) = E\{\tilde{g}(k)^2\}$ and $V^g_{\sigma}(k) = E\{(\tilde{g}(k)^{[1]}\}^2\}$.

Introduce $\sum_{\sigma=1}^{N_{\sigma}} |\kappa^{j}(\nu)\}$ infinite subsequences $\{\kappa^{j}(\nu)\}$ of the set of nonnegative integers $\mathcal{N}^{j}$, $j = 1,\ldots,n$, where $\kappa^{j}(\nu)$ for a given $\nu$ defines an instant $k$ corresponding to an update at node $i$ realized as a consequence of a tick of node $j (\kappa^{j}(\nu) < \kappa^{j}(\nu_2)$ for $\nu_1 < \nu_2$ and $\cup_{j=1}^{N_{\sigma}}(\kappa^{j}(\nu)) = \mathcal{N}^{j}$). Define $\Pi(k,\sigma)^{[1]} = \prod_{\sigma_1=1}^{k} [1 + (1 - \sigma_1 \sigma_2) H_{\sigma_2}(\sigma)^{[1]}], s = 1,\ldots,\prod_{\sigma_1=1}^{k} |\kappa^{j}(\nu)|, \nu = 0,1,2,\ldots,$ in which $\kappa^{j}(\nu)$ for a given $\nu$ defines an instant $k$ corresponding to an update at node $i$ realized as a consequence of a tick of node $j (\kappa^{j}(\nu) < \kappa^{j}(\nu_2)$ for $\nu_1 < \nu_2$ and $\cup_{j=1}^{N_{\sigma}}(\kappa^{j}(\nu)) = \mathcal{N}^{j}$). Therefore, it follows that $E\{(\Pi(k,\sigma)^{[1]}\} < \infty$, since $\Pi(k,\sigma)^{[1]}$ are mutually independent. For $\text{AlgDrift.c}$, we have that $H_{\sigma}(\sigma)^{[1]} = H_{\sigma}(\sigma)^{[1]} - H_{\sigma}(\sigma_0)^{[1]}$, where $H_{\sigma}(\sigma_0)^{[1]}$ is zero mean i.i.d. and $H_{\sigma}(\sigma)^{[1]}$ a finite w.p.1 random variable ($\sigma_0 = \kappa^{j}(0)^{[1]}$). Therefore, we have

$$E\{ \frac{1}{\sigma_0} H_{\sigma}(\sigma)^{[1]} \} < \infty.$$
\[ H_2(\sigma) = [H_2(\sigma)^{[1]}; H_2(\sigma)^{[2]}]. \]

We start the analysis by observing that for any \( n \)-vector \( x \) and any \( \sigma \) large enough

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

where \( 0 < \lambda_{\min}(Q^\Pi), \lambda_{\max}(R^\Pi) < \infty \) and \( q = \frac{1}{\max\{1/j: j \in \mathbb{N}_+\}} \) for AlgDrift.a, \( q = \frac{1}{2} \) for AlgDrift.b and \( q = \frac{1}{2} \) for AlgDrift.c. Because \( q > 0 \) (Lemma 3), after standard technicalities based on the classical results on stochastic approximation [21], [23], it follows that \( \Pi_{\sigma \in [k^t(i)]} \|\Pi(\sigma, \sigma)\| \to \sigma \to \infty 0, i = 1, \ldots, n, j \in \mathcal{F}, \) 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

\[ W^x(\sigma + 1) \leq \left[ 1 - c_1 \frac{1}{\sigma^5} \right] W^x(\sigma) + C_1 \frac{1}{\sigma^5} V^x(\sigma), \]

where \( 0 < c_1, C_3 < \infty \).

Using the methodology of [24], [25] we can, consequently, show that \( \sup_{t} V^x(t) < \infty \). This gives rise to the conclusion that \( \bar{g}(k)^{[1]} \) tends to a random variable \( X^* \) \( (E\{X^*\} < \infty) \) and \( \bar{g}(k)^{[2]} \) to zero in the mean square sense and w.p.1. Consequently

\[ \bar{g}_w = T \left[ \lim_{k \to \infty} \bar{g}(k)^{[1]} \right] = X^*, \]

which proves the theorem.

### VII. PROOF OF THEOREM 2

After introducing the expression for \( z(k) \) into (15), we use the approximation \((1 + \frac{1}{k^5})^{\Delta (k)} \approx 1 + \frac{\Delta (k)}{k^5}\) and obtain, after neglecting higher order terms, that for \( k \) large enough

\[ z(k + 1) = z(k) + \frac{1}{k^5} [B^* + F_2(k)] \Delta (k) + \frac{1}{k^5} [F_2(k)] \Delta (k) \]

Applying the methodology of the proof of Theorem 1 to (25), we observe that for \( \zeta' < 1 \) the term proportional to \( \frac{1}{k^5} \) introduced by the formulation of the recursion for \( z(k) \), can be neglected for \( k \) large enough with respect to the term proportional to \( \frac{1}{k^5} \). As \( B^* \) 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 \( \frac{1}{k^5} \) and \( \frac{1}{k^5} \) are of the same order of magnitude; as a result, the convergence conditions for (25) depend on the properties of the matrix \( B^* \). Hence the result follows.