Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 528161, 9 pages
doi:10.1155/2009/528161

Research Article

Analysis of Distributed Consensus Time Synchronization with Gaussian Delay over Wireless Sensor Networks

Gang Xiong and Shalinee Kishore

Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA

Correspondence should be addressed to Shalinee Kishore, skishore@lehigh.edu

Received 28 June 2008; Accepted 6 May 2009

Recommended by Visa Koivunen

This paper presents theoretical results on the convergence of the distributed consensus timing synchronization (DCTS) algorithm for wireless sensor networks assuming general Gaussian delay between nodes. The asymptotic expectation and mean square of the global synchronization error are computed. The results lead to the definition of a *time delay balanced network* in which average timing consensus between nodes can be achieved despite random delays. Several structured network architectures are studied as examples, and their associated simulation results are used to validate analytical findings.

1. Introduction

Wireless sensor networks are typically comprised of inexpensive, small-sized, power-limited terminals. In a variety of applications, sensor nodes are required to maintain accurate time synchronization, for example, moving object tracking, reconnaissance and surveillance, environmental monitoring, and so forth [1]. This necessitates algorithms that achieve and maintain *global time synchronization* at all network nodes, that is, algorithms that align all nodes to a common notion of time.

Due to imperfections in low-cost hardware nodes and the decentralized nature of wireless sensor networks, global time synchronization has been recognized as a particularly challenging task. Recently, several distributed time synchronization algorithms have been proposed; one such class is *distributed consensus time synchronization* (DCTS) [2]. In the DCTS approach, a global time consensus can be sufficiently reached within a connected network by averaging pairwise local time information. In [3], Olfati-Saber et al. established a theoretical framework for the analysis of consensus synchronization algorithms. Later, a fully distributed, asynchronous DCTS algorithm was proposed in [4]; this scheme was designed to reach agreement on time offset and skew offset between network nodes using media access control (MAC) layer time-stamped packet exchanges. As an alternative, a physical layer-based DCTS algorithm was introduced in [5] by modeling sensor nodes as coupled discrete time oscillators. Based on our knowledge, the existing body of literature on the DCTS approach does not examine the effects of time delay uncertainty between network nodes. In this paper, we study the convergence of the DCTS algorithm when uncertain delays impact local pairwise time information exchange.

In [6], Xiao et al. considered distributed average consensus with additive noise and investigated the design of network link weights to minimize the mean-square deviation in steady state. In this paper, we analyze the convergence characteristics of the DCTS algorithm under Gaussian delay uncertainties. First, we determine the asymptotic expectation of the global synchronization error. Our results lead to the definition of a *time delay balanced network*, and we claim that under such network topologies average timing consensus between nodes can be achieved despite the presence of random delays. Additionally, we show that the asymptotic mean square synchronization error is lower and upper bounded by several values related to network parameters. As examples, we analyze the global synchronization error of the DCTS algorithm for several structured networks.

This paper is outlined as follows. Section 2 provides background and system model for the DCTS algorithm studied here. Section 3 presents convergence results on

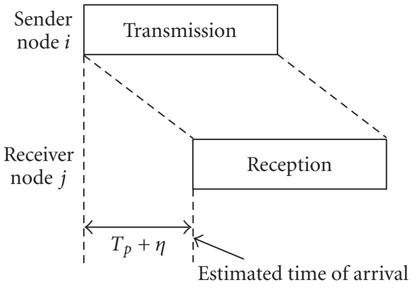

Figure 1: Physical layer-based time delay model.

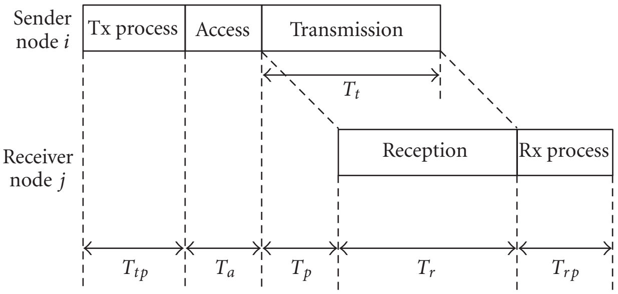

Figure 2: MAC layer-based time delay model.

the synchronization error of the DCTS algorithm due to Gaussian random delays between nodes. Section 4 discusses the convergence characteristics of the global synchronization error for several structured networks. Simulation results are presented in Section 5, and we conclude our discussion in Section 6.

2. Background and System Model

2.1. Time Delay for Local Time Information Exchange. The DCTS algorithm requires local time information exchange between two or more nodes in a wireless sensor network. This exchange can occur using either MAC layer time-stamped packets or via physical layer pulse signals. In either case, the delay between two network nodes is defined as the interval between when the time information is generated by the sender node and when this information is determined by the receiver node. Furthermore, in either case, this delay can be comprised of a deterministic and a random portion. In the following, we discuss the delay sources at the two layers and argue that, in both cases, a common underlying model of Gaussian delay uncertainty can be adopted. (We have separately examined the performance of the DCTS algorithm considering alternate delay distributions, e.g., exponential delay distribution [7]. Results show similar performance bounds as those presented in this paper for the Gaussian assumption. For this reason, we constrain our discussion here to the more common Gaussian delay model.)

2.1.1. Physical Layer-Based Time Delay. Sender nodes using physical layer synchronization algorithms convey local time information to receiver nodes by transmitting pulse signals according to their local clocks. The receiver node, however, estimates the arrival time of the pulse signal as the clock of the sender node. As shown in Figure 1, there is an offset between the transmit time of the pulse at the sender and the arrival time estimate at the receiver.

One source of this lag is T_p, the propagation delay between the sender and receiver nodes. The propagation delay is related to the distance between the two nodes such that $T_p = \ell_{ij}/c$, where ℓ_{ij} is the distance between nodes i and j and c is the speed of light. Once the pulse signal propagates to the receiver, the receiver node takes some time to reliably detect the pulse signal and to make an arrival time estimate. We assume the arrival time estimation procedure at the receiver will automatically compensate for this detection and estimation delay. However, since the pulse signal is received in noise (and may additionally experience fading over the wireless link), the actual arrival time estimate produced at the receiver will have an associated error. It is known from parameter estimation theory that any maximum likelihood (ML) estimator is asymptotically unbiased, and an ML estimate is asymptotically Gaussian distributed [8]. Thus, if an ML arrival time estimator is employed at the receiver, the arrival time estimation error can be modeled as a Gaussian random variable, ν_{PHY}, with zero mean and variance σ^2_{PHY} (the variance of arrival time estimator). In the physical layer delay model used here, we assume such an estimation error and write the total delay between the transmit time and estimated arrival time of a pulse signal as

$$T_{\text{PHY-delay}} = T_p + \nu_{\text{PHY}}. \tag{1}$$

2.1.2. MAC Layer-Based Time Delay. At the MAC layer, local time information at a sender node is clocked and incorporated into a packet during packet formation. The overall delay between two nodes exchanging such time-stamped packets is, therefore, the time interval between when the sender time is clocked and when the receiver node decodes this time information from its received packet [9]. The sources of delay during this interval are shown in Figure 2.

The major sources of random delay at the MAC layer are T_{tp}, the transmission processing time; T_a, the channel access time; and T_{rp}, the receiver processing time. The delay in processing a packet (at either the transmitter or receiver) depends on several factors such as the protocol processing time, the CPU load, and delays in the operating system. T_a, on the other hand, is the time the sender node that must wait to access the transmit channel, which is determined by the MAC protocol in use as well as the current network traffic. Here, we assume the overall delay, $T_{tp} + T_a + T_{rp}$ results from the additive effect of delays introduced by several independent random processes (e.g., the instantaneous workload on the sender/receiver CPU, packet generation processes at other network nodes, etc.). Using the central limit theorem, we model this delay as a Gaussian random variable with mean $\mu_{\text{MAC}} = \mathbb{E}(T_{tp}) + \mathbb{E}(T_a) + \mathbb{E}(T_{rp})$ and variance $\sigma^2_{\text{MAC}} = \text{Var}(T_{tp}) + \text{Var}(T_a) + \text{Var}(T_{rp})$. Additionally, the packet experiences a propagation delay of T_p; the overall MAC layer delay is therefore given as

$$T_{\text{MAC-delay}} = T_p + \nu_{\text{MAC}}. \tag{2}$$

In the following, we use a general delay model that incorporates the two delay calculations for the physical and MAC layers, that is, we assume

$$T_{\text{delay}} = T_c + T_p + \nu, \tag{3}$$

where T_c is a constant equal to zero for physical layer-based schemes and μ_{MAC} for MAC layer-based schemes; and ν is a zero mean Gaussian random variable. The variance of ν, σ^2, is equal to σ^2_{PHY} for physical layer schemes and to σ^2_{MAC} for MAC layer-based schemes.

2.2. DCTS Algorithm With Gaussian Delay.

In each iteration of the DCTS algorithm, each node processes and decodes the time-stamped message from its neighbors in the MAC layer-based approach or estimates the arrival time of its neighbors' pulse signals in the physical layer scheme. Each node then updates its local clock time using the weighted average of the time differences with its neighbor nodes. It is well known that in a connected network with nonrandom delay between nodes, this DCTS algorithm can reach *average consensus* [10]; that is, all nodes converge to the average of the initial timing differences between the nodes.

Our study focuses on the operation of the DCTS algorithm when there are both deterministic and random (Gaussian) delays during local time information exchange, as described above. In this case, the timing update rule of the DCTS algorithm at each node i is given as

$$t_i(k+1) = t_i(k) + \varepsilon \sum_{j \in \mathcal{N}_i} \left[\hat{t}_j(k) - t_i(k)\right], \tag{4}$$

where $t_i(k)$ is the local time at node i during iteration k; $\mathcal{N}_i$ is the set of neighboring nodes that can communicate reliably with node i; $\hat{t}_j(k) = t_j(k) + T_{\text{delay}} = t_j(k) + T_c + \ell_{ij}/c + \nu_j(k)$; T_c is the constant delay defined above; ε is the constant step size for each iteration; $\nu_j(k)$ are i.i.d Gaussian random variables, with zero mean and variance σ^2. Local time information exchange between nodes i and j under this delay model is shown in Figure 3.

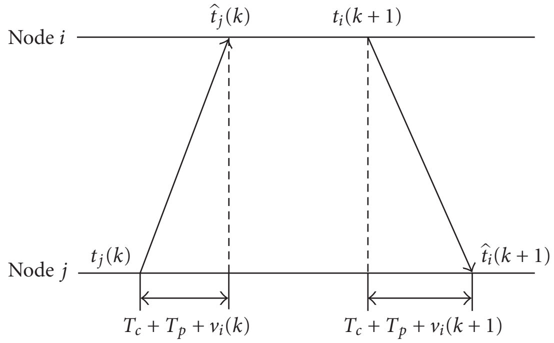

FIGURE 3: DCTS algorithm with Gaussian delay during local time information exchange.

The DCTS algorithm in (4) can be rearranged as

$$t_i(k+1) = t_i(k) + \varepsilon \sum_{j \in \mathcal{N}_i} \left[t_j(k) - t_i(k)\right] + n_i(k), \tag{5}$$

where $n_i(k) = \varepsilon \sum_{j \in \mathcal{N}_i} [T_c + \ell_{ij}/c + \nu_j(k)]$. It should be noted that $n_i(k)$ and $n_j(k)$ might not be independent between nodes i and j since the two nodes might have identical noise coming from some potentially overlapping neighbors.

2.3. Network Model and Some Preliminaries.

In the following, we model a wireless sensor network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a set of n nodes $\mathcal{V} = \{1, 2, \ldots, n\}$ and a set of edges $\mathcal{E}$. (The convergence properties presented here can be easily extended for a directed graph. We omit this extension here.) Each edge is denoted as $e = (i, j) \in \mathcal{E}$ where $i \in \mathcal{V}$ and $j \in \mathcal{V}$ are two nodes connected by edge e. We assume that the presence of an edge (i, j) indicates that nodes i and j can communicate with each other reliably. We assume here a connected graph; that is, there exists a path connecting any pair of distinct nodes in the network.

Given this network model, we denote A as the adjacency matrix of $\mathcal{G}$ such that

$$A(i,j) = \begin{cases} 1, & (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

Next, we let L be the graph Laplacian matrix of $\mathcal{G}$ which is defined as

$$L = D - A, \tag{7}$$

where $D = \text{diag}(d_1, d_2, \ldots, d_n)$ is the degree matrix of $\mathcal{G}$. Specifically, d_i is equal to the number of neighbors of node i with which it can communicate reliably, that is, $d_i = |\mathcal{N}_i|$. Given this matrix L, we can show that $L\mathbf{1} = \mathbf{0}$ and $\mathbf{1}^{\text{T}} L = \mathbf{0}^{\text{T}}$, where $\mathbf{1} = [1, 1, \ldots, 1]^{\text{T}}$ and $\mathbf{0} = [0, 0, \ldots, 0]^{\text{T}}$. Additionally, L is a symmetric positive semidefinite matrix (implying its eigenvalues are all nonnegative), and for a connected graph, the rank of L is $n - 1$ and its eigenvalues can be arranged in increasing order as $0 = \lambda_1(L) < \lambda_2(L) \leq \cdots \leq \lambda_n(L)$ [11]. We now define vectors $\mathbf{t}(k) = [t_1(k), t_2(k), \ldots, t_n(k)]^{\text{T}}$ and $\mathbf{n}(k) = [n_1(k), n_2(k), \ldots, n_n(k)]^{\text{T}}$. Based on these definitions, the evolution of DCTS algorithm in (5) can be written as

$$\mathbf{t}(k+1) = H\mathbf{t}(k) + \mathbf{n}(k), \tag{8}$$

where $H = I_n - \varepsilon L$ is called a Perron matrix of a graph with parameter ε [3]. Here, I_n denotes the $n \times n$ identity matrix. The eigenvalues of H are $\lambda_i(H) = 1 - \varepsilon\lambda_i(L)$ and can be ordered in decreasing order: $1 = \lambda_1(H) > \lambda_1(H) \geq \cdots \geq \lambda_n(H)$. It is worth mentioning that the constant step size ε_{opt} which minimizes convergence time is given as $2/[\lambda_2(L) + \lambda_n(L)]$ [12]. (Note that the optimal ε is generally difficult to obtain as it involves computing the eigenvalues of the Laplacian matrix L. However, in practical applications, a numerical solution can be obtained offline based on node deployment within a given wireless sensor network, and this ε_{opt} can then be flooded to all nodes before they run the DCTS algorithm.) Let us define $\mathbf{v}(k) = [v_1(k), v_2(k), \ldots, v_n(k)]^{\mathrm{T}}$ and $\mathbf{u} = [u_1, u_2, \ldots, u_n]^{\mathrm{T}}$, where $u_i = \sum_{j\in\mathcal{N}_i}(T_c + \ell_{ij}/c)$. Then the noise vector in (8) is given as $\mathbf{n}(k) = \varepsilon[\mathbf{u} + A\mathbf{v}(k)]$.

When there is no Gaussian delay between nodes, it can be shown that [10, 12], for a time-invariant, connected, undirected network, when $\varepsilon \in (0, 2/\lambda_n(L))$, *average consensus* can be asymptotically achieved by the DCTS algorithm, that is, $\lim_{k\to\infty} H^k = (1/n)\mathbf{1}\mathbf{1}^{\mathrm{T}}$. In our discussion, we also assume an undirected, connected network with a constant step size $0 < \varepsilon < 2/\lambda_n(L)$ unless otherwise stated.

In the following analysis, we use the following matrices: $K = (1/n)\mathbf{1}\mathbf{1}^{\mathrm{T}}$, $P = H - K$ and $Q = I_n - K$. For matrices P and Q, it is straightforward to show that (1) the eigenvalues of P agree with those of H except that $\lambda_1(H) = 1$ is replaced by $\lambda_1(P) = 0$; (2)$P^k = H^k - K$ such that $\lim_{k\to\infty} P^k = \mathbf{0}_n$; and (3)$QP^kQ = P^k$ and $Q^k = Q$.

3. Convergence Analysis of DCTS Algorithm with Gaussian Delay

Let us define the average value in each iteration as $m(k) = (1/n)\mathbf{1}^{\mathrm{T}}\mathbf{t}(k)$. Then, mean of the average value $m(k)$ in each iteration of the DCTS algorithm is $m(0) + k\varepsilon/n\sum_{i=1}^{n} u_i$, and the variance of the average value is $k\varepsilon^2\sigma^2/n^2\sum_{i=1}^{n} d_i^2$. It can be seen that as iteration time increases, both mean and variance increase linearly with the time index k. Furthermore, the variance of $m(k)$ increases linearly with the variance of the random Gaussian delay, σ^2.

3.1. Expectation and Second Central Moment of Disagreement Vector. We now define the disagreement vector as $\boldsymbol{\delta}(k) = \mathbf{t}(k) - K\mathbf{t}(k)$; that is, $\boldsymbol{\delta}(k)$ is the difference between the updated times and the actual average times of the network nodes. Then, the disagreement vector evolves as $\boldsymbol{\delta}(k) = P\boldsymbol{\delta}(k-1) + Q\mathbf{n}(k-1)$.

Lemma 1. *For the DCTS algorithm in* (8)*, the expectation of disagreement vector is*

$$\mathbb{E}[\boldsymbol{\delta}(k)] = P^k\boldsymbol{\delta}(0) + \varepsilon\sum_{l=0}^{k-1} P^l Q\mathbf{u}. \tag{9}$$

The proof of this lemma is straightforward and thus omitted from the paper. Let us define the second central moment of disagreement vector as $\kappa_\delta(k) = \mathbb{E}\{(\boldsymbol{\delta}(k) - \mathbb{E}[\boldsymbol{\delta}(k)])^{\mathrm{T}}(\boldsymbol{\delta}(k) - \mathbb{E}[\boldsymbol{\delta}(k)])\}$. We next note the following.

Lemma 2. *For the DCTS algorithm in* (8)*, the second central moment of disagreement vector is given as*

$$\kappa_\delta(k) = \boldsymbol{\delta}(0)^T P^{2k}\boldsymbol{\delta}(0) + \varepsilon^2\sigma^2\mathbf{tr}\left(Q\sum_{l=0}^{k-1} P^{2l}QA^2\right), \tag{10}$$

where $\mathbf{tr}(\cdot)$ *denotes the trace of a matrix.*

Proof. Please see Appendix. □

3.2. Asymptotic Expectation of Global Synchronization Error. Using Lemma 1, we see that the steady state of expectation of disagreement vector is

$$\boldsymbol{\mu}(\infty) \doteq \lim_{k\to\infty}\mathbb{E}[\boldsymbol{\delta}(k)] = \varepsilon(I_n - P)^{-1}Q\mathbf{u}. \tag{11}$$

Let us define $W_1 = (I_n - P)^{-1}$; then the eigenvalues of W_1 are $\lambda_1(W_1) = 1$ and $\lambda_i(W_1) = 1/[\varepsilon\lambda_i(L)]$, $i = 2, \ldots, n$. For this $\boldsymbol{\mu}(\infty)$, we can show that.

Theorem 1. *In a network with fixed, connected topology,* $\boldsymbol{\mu}(\infty)$ *in* (11) *is a constant vector independent of the constant value of* ε.

Proof. Let us denote the eigenvectors of W_1 as $\boldsymbol{\omega}_i$. It is easy to check that the eigenvector corresponding to $\lambda_1(W_1) = 1$ is $\boldsymbol{\omega}_1 = \mathbf{1}$. $\boldsymbol{\mu}(\infty)$ in (11) can thus be written as

$$\begin{aligned}\boldsymbol{\mu}(\infty) &= \varepsilon\mathbf{1}\mathbf{1}^{\mathrm{T}}Q\mathbf{u} + \left[\sum_{i=2}^{n}\frac{1}{\varepsilon\lambda_i(L)}\boldsymbol{\omega}_i\boldsymbol{\omega}_i^{\mathrm{T}}\right]Q\varepsilon\mathbf{u} \\ &= (L+K)^{-1}Q\mathbf{u}.\end{aligned} \tag{12}$$

Thus, $\boldsymbol{\mu}(\infty)$ does not depend on ε. □

Thus, for a constant step size ε, the steady state of expectation of disagreement vector is a constant vector regardless of ε. In other words, in a network with fixed topology, the expectation of global synchronization error is the same regardless of the speed of synchronization.

In general, we see that the DCTS algorithm with Gaussian delay cannot achieve average consensus since $\boldsymbol{\mu}(\infty)$ is a linear function of $\mathbf{u}$ (is not equal to $\mathbf{0}$). This global synchronization error can be viewed as the accuracy of time synchronization algorithm. If this synchronization error is tolerable or small compared to time resolution of the system, we say that this DCTS algorithm still achieves the average consensus but with "*tolerable synchronization error*". Let us now define the asymptotic expectation of pairwise synchronization error as

$$\Delta t_{i,j} = \lim_{k\to\infty}\mathbb{E}\left[t_i(k) - t_j(k)\right] = \mu_i(\infty) - \mu_j(\infty), \quad i, j \in \mathcal{V}. \tag{13}$$

Hence, the maximum asymptotic expectation of global synchronization error between any two nodes is $\Delta t_{\max} = \max\{|\Delta t_{i,j}|\}$. It is worth mentioning that, under certain network topologies (e.g., the ring network studied in Section 4), average consensus can still be asymptotically achieved when using the DCTS approach under Gaussian delays.

Recall that $\boldsymbol{\mu}(\infty) = (L+K)^{-1}Q\mathbf{u}$. In this equation, $Q\mathbf{u} = \mathbf{u} - K\mathbf{u}$ is the disagreement vector of $\mathbf{u}$. When $\mathbf{u} = K\mathbf{u}$, we see that $\sum_{j\in\mathcal{N}_i}(T_c + \ell_{ij}/c) = \sum_{m\in\mathcal{N}_k}(T_c + \ell_{km}/c)$, for $(i, j) \in \mathcal{E}$ and $(k, m) \in \mathcal{E}$. More specifically, when $d_i = d_j$ and $\ell_{ij} = \ell_{km}$, then $\boldsymbol{\mu}(\infty) = \mathbf{0}$ and $\Delta t_{\max} = 0$, implying that the DCTS algorithm achieves average consensus asymptotically. The condition above indicates that the time delay between nodes can be canceled if each node receives the same amount of time delay from all neighbors; networks that meet this condition are defined as follows.

Definition 1. A network is called "time delay balanced network" if $\sum_{j\in\mathcal{N}_i}(T_c + \ell_{ij}/c) = \sum_{m\in\mathcal{N}_k}(T_c + \ell_{km}/c)$, for $(i,j)\in\mathcal{E}$ and $(k,m)\in\mathcal{E}$, or equivalently, $\Delta t_{\max} = 0$.

Otherwise we refer to the network as "*time delay unbalanced*". It is worth mentioning that a similar definition of "equal delay networks" was discussed in [13] for continuous time network synchronization. Based on the definition above, we see that time delay balance may be readily (but not exclusively) achieved in well-structured networks.

3.3. Asymptotic Mean Square Synchronization Error. Using Lemma 2, the steady state of second central moment of disagreement vector is

$$\kappa_\delta(\infty) \doteq \lim_{k\to\infty}\kappa_\delta(k) = \varepsilon^2\sigma^2\mathbf{tr}\Big\{\Big[\left(I_n - P^2\right)^{-1} + Q - I_n\Big]A^2\Big\}. \tag{14}$$

Let us define $W_2 = (I_n - P^2)^{-1} + Q - I_n$. Thus, the eigenvalues of W_2 are $\lambda_1(W_2) = 0$ and $\lambda_i(W_2) = 1/[2\varepsilon\lambda_i(L) - \varepsilon^2\lambda_i^2(L)]$, $i = 2,\dots,n$. We now define the asymptotic mean square time synchronization error as

$$\sigma_{\Delta t}^2 = \lim_{k\to\infty}\sum_{i=1}^{n}\mathbb{E}\Big[\left|t_i(k) - m(k)\right|^2\Big], \tag{15}$$

which indicates the amount of error by which the updated time at each node differs from the average value over all n nodes. We see that

$$\sigma_{\Delta t}^2 = \mathbf{u}^{\mathrm{T}}Q(L+K)^{-2}Q\mathbf{u} + \varepsilon^2\sigma^2\mathbf{tr}\left(W_2A^2\right). \tag{16}$$

Theorem 2. *For a connected, time delay unbalanced network, $\sigma_{\Delta t}^2$ in* (15) *is bounded by*

$$\begin{gathered}\sigma_{\Delta t}^2 \geq \frac{\mathbf{u}^TQ\mathbf{u}}{\xi_1} + \varepsilon\sigma^2\lambda_{\min}\left(A^2\right)\sum_{i=2}^{n}\lambda_i,\\ \sigma_{\Delta t}^2 \leq \frac{\|\mathbf{u}\|^2}{\xi_2} + \varepsilon\sigma^2\min\left\{\mathcal{D}_n\max\{\lambda_i\}, \lambda_{\max}\left(A^2\right)\sum_{i=2}^{n}\lambda_i\right\},\end{gathered} \tag{17}$$

where $\xi_1 = \lambda_n^2(L)$, $\xi_2 = \min\{\lambda_2^2(L), 1\}$, $\lambda_i = 1/[2\lambda_i(L) - \varepsilon\lambda_i^2(L)]$, $i = 2,\dots,n$, $\mathcal{D}_n = \sum_{i=1}^{n}d_i$ is the total degree in the networks, and $\|\cdot\|$ denotes the ℓ_2 norm of a vector.

Proof. Please see the Appendix. □

Based on this result, it can be seen that the lower and upper bounds of $\sigma_{\Delta t}^2$ are determined by several values related to network parameters: eigenvalues of L and A^2, total degree of network, step size, and delay time vector.

4. DCTS Algorithm with Gaussian Delay in Structured Networks

In this section, we apply the DCTS algorithm under Gaussian delay for several structured networks. In particular, we study the structured networks as they are analytically tractable, provide some valuable insights, and can be used to validate our analytical findings. (Typical sensor network deployments may in fact have a random topology. We study how our results extend to such random network scenarios using simulation in Section 5.) Specifically, we analyze at the impact of Gaussian delay when using DCTS in the following networks.

Definition 2 (A Ring Network with Equal Distance (R_n)). A ring network is a network that consists of a single cycle. The ring network with equal distance is a ring network that has n nodes, n edges, and $\ell_c = \ell_{ij} = \ell_{km}$ for $(i,j)\in\mathcal{E}$ and $(k,m)\in\mathcal{E}$.

Definition 3 (A Star Network with Equal Distance (S_n)). A star network is a network that consists of edge set $\{(i,n), 1\leq i<n\}$. The star network with equal distance is a star network that has n nodes, $n-1$ edges, and $\ell_c = \ell_{ij} = \ell_{km}$ for $(i,j)\in\mathcal{E}$ and $(k,m)\in\mathcal{E}$.

Definition 4 (A Hypercube Network with Equal Distance Degree (H_n)). A hypercube network with equal distance degree is a hypercube network that has n nodes, $n\log_2 n$ edges and $\sum_{j\in\mathcal{N}_i}\ell_{ij} = \sum_{i\in\mathcal{N}_j}\ell_{ji}$.

Figure 4 illustrates several examples of such networks. In the following, we simply present convergence results for these structured networks without proof.

4.1. Convergence Properties for Ring Networks. For a ring network R_n, the DCTS algorithm in (8) produces a global synchronization error with the following properties:

$$\begin{gathered}\Delta t_{\max} = 0,\\ \sigma_{\Delta t}^2 \geq \frac{\varepsilon\sigma^2}{2}\left[1 + \cos\left(\frac{4\pi\lfloor(n+1)/4\rfloor}{n}\right)\right]\sum_{i=1}^{n-1}\lambda_i,\\ \sigma_{\Delta t}^2 \leq \varepsilon\sigma^2\min\left\{\max\left\{\frac{n\lambda_i}{2}\right\}, \sum_{i=1}^{n-1}\lambda_i\right\},\end{gathered} \tag{18}$$

where $\lambda_i = 1/[1-\varepsilon+(2\varepsilon-1)\cos(2\pi i/n) - \varepsilon\cos^2(2\pi i/n)]$, $i = 1,\dots,n-1$. Since $\Delta t_{\max} = 0$, we see that the ring network R_n is a time delay balanced network.

4.2. Convergence Properties for Star Networks. For a star network S_n, the DCTS algorithm in (8) produces a global synchronization error with the following properties:

$$\begin{aligned}\sigma_{\Delta t}^2 &\geq \frac{\mathbf{u}^{\mathrm{T}}Q\mathbf{u}}{n^2},\\ \sigma_{\Delta t}^2 &\leq \|\mathbf{u}\|^2 + (n-1)\varepsilon\sigma^2\\ &\quad\cdot\min\{\max\{2\lambda_1, 2\lambda_2\}, (n-2)\lambda_1 + \lambda_2\},\end{aligned} \tag{19}$$

where $\lambda_1 = 1/(2-\varepsilon)$ and $\lambda_2 = 1/(2n-\varepsilon n^2)$.

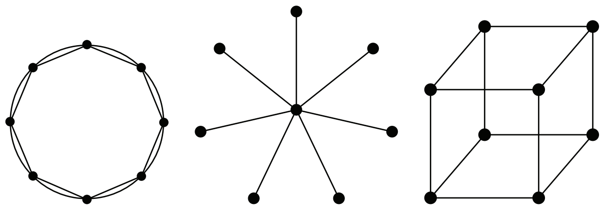

Figure 4: Structured networks: (a) R_8, (b) S_8, and (c) H_8.

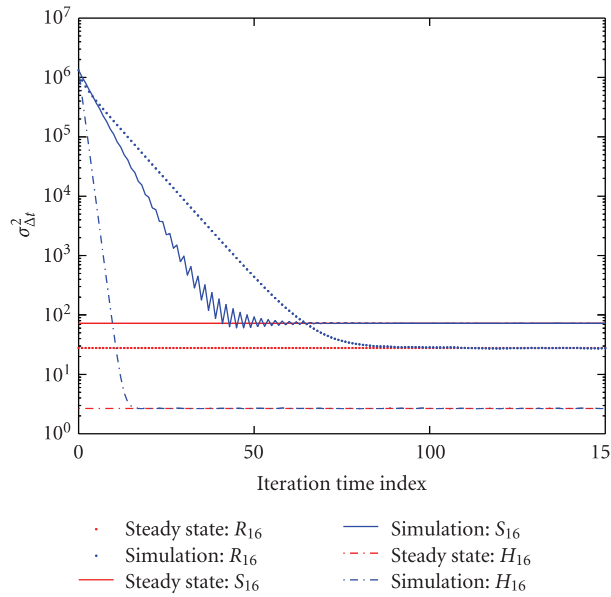

Figure 5: $\sigma^2_{\Delta t}$ as a function of the iteration time index for the DCTS algorithm in structured networks with Gaussian delay between network nodes.

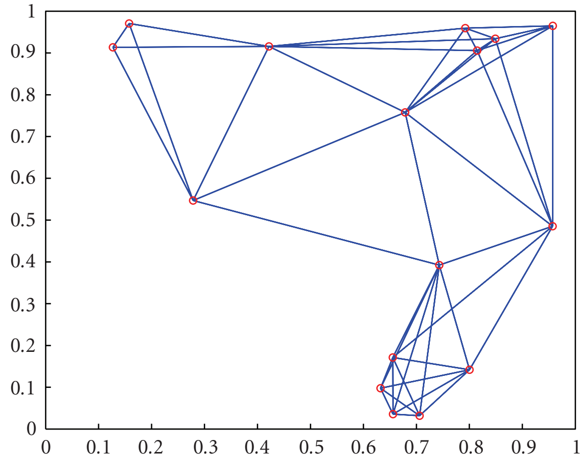

Figure 6: Random network with 16 nodes.

The star network S_n is time delay unbalanced. Furthermore, it should be noted that when operating the DCTS algorithm with ε_{opt}, we get that $\kappa_\delta(\infty) = (n-1)\sigma^2/n$. This is because W_2 can be simplified in this case to $((n+1)^2/4n)Q$. As a result, we see that as n becomes large, $\kappa_\delta(\infty) \approx \sigma^2$.

4.3. Convergence Properties for Hypercube Networks. For a hypercube network H_n, the DCTS algorithm in (8) produces a global synchronization error with the following properties:

$$\Delta t_{\max} = 0,$$

$$\sigma^2_{\Delta t} \ge \begin{cases} 0, & \text{when } \vartheta_n \text{ is even}, \\ \varepsilon\sigma^2 \displaystyle\sum_{i=1}^{\vartheta_n} \binom{\vartheta_n}{i} \lambda_i, & \text{when } \vartheta_n \text{ is odd}, \end{cases} \qquad (20)$$

$$\sigma^2_{\Delta t} \le \varepsilon\sigma^2 \vartheta_n \min\left\{ \max\{n\lambda_i\}, \vartheta_n \sum_{i=1}^{\vartheta_n} \binom{\vartheta_n}{i} \lambda_i \right\},$$

where $\vartheta_n = \log_2 n$ and $\lambda_i = 1/(4i - 4\varepsilon i^2)$, $i = 1, \ldots, \vartheta_n$. Since $\Delta t_{\max} = 0$, the hypercube network is also time delay balanced.

5. Simulation Results

The simulation parameters are described as follows: initial time phase of node i is $(i - 1/2)T/n$, $i = 1, \ldots, n$, where $T = 1000\,\mu$s, and the standard deviation of delay variance is $\sigma = 1\,\mu$s. The simulation results are based on 5000 runs. (Trends similar to the ones noted below were observed when initial time offsets between nodes were arbitrary (e.g., when they were uniformly distributed over $[0, T]$). We use this fixed offset assumption here for comparison purposes.)

5.1. Structured Networks. In our simulations of structured networks, we assume $u_{cp} = T_c + \ell_c/c = 10\,\mu$s and the optimal constant step size is ε_{opt}. The simulation results and asymptotic mean square time synchronization errors for structured networks with 16 nodes are shown in Figure 5. The asymptotic mean square (steady-state) time synchronization errors $\sigma^2_{\Delta t}$ are calculated from (16). It can be seen that as the time index increases, the mean square time synchronization errors approach their respective steady state values when using DCTS with Gaussian delay. As expected, DCTS algorithm in a hypercube network achieves the smallest variance of synchronization error and the fastest convergence among those structured networks. This is primarily due to the high degree of connectivity in the hypercube network, which also results in the smallest value of ε_{opt}.

5.2. Random Networks. We also present here simulation results for a random network comprised of n nodes that were randomly generated with uniform distribution over a unit square kilometer; two nodes were assumed connected if the distance between them was less than η, a predefined threshold. One realization of such a network with 16 nodes is shown in Figure 6. We assume that the average distance between two nodes is 0.5 km.

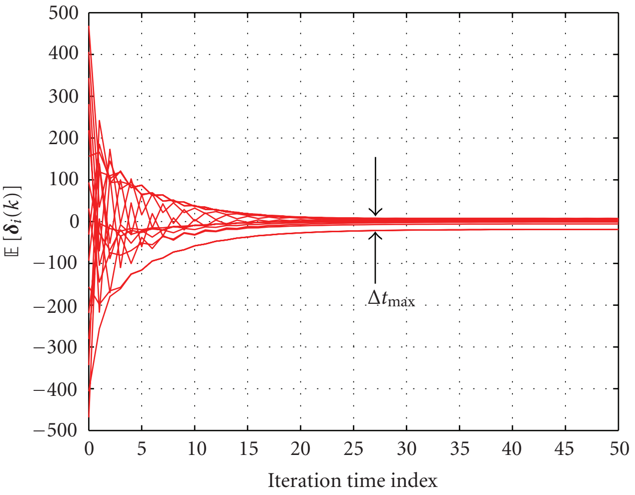

FIGURE 7: Disagreement evolution of the DCTS algorithm in a random network with Gaussian delay.

Figure 7 shows the simulation results of $\mathbb{E}[\delta_i(k)]$ in a particular realization of the random network. We choose ε_{opt} for this simulation. It can be seen that, asymptotically, there exists global synchronization error between some pairs of nodes, and $\Delta t_{\max} = 26.4130\,\mu$s for this random network. If we specify a threshold Δt_{Th} to be greater than or equal to this $\Delta t_{\max}$, we call this network as "*average consensus achievable with tolerable synchronization error*" as described in Section 3.2.

As mentioned above, the generation of random networks depends on two parameters: the number of users n and the predefined threshold η. In order to better understand the performance of DCTS algorithm with Gaussian delay in such networks, we show the asymptotic values of $\sigma^2_{\Delta t}$ and $\Delta t_{\max}$ in random networks as functions of n and η in Figures 8 and 9. For a given value of n and η, the presented asymptotic values were averaged over 1000 realizations of the random network, where we excluded disconnected network realizations. From the plots, we see that with the same threshold η, the asymptotic values of $\sigma^2_{\Delta t}$ and $\Delta t_{\max}$ decrease as the number of nodes increases. Similarly, with the same number of nodes in the networks, $\sigma^2_{\Delta t}$ and $\Delta t_{\max}$ decrease as threshold η increases. This is primarily due to the fact that when the number of nodes or threshold increases, the random network behaves more like a "*time delay balanced network*".

6. Conclusions

In this paper, we present theoretical results on the convergence of the DCTS algorithm for wireless sensor networks with general Gaussian delay between nodes. Specifically, we compute the asymptotic expectation and mean square of the global synchronization error of the DCTS algorithm. The results lead to the definition of a *time delay balanced network* in which average timing consensus between nodes can be achieved despite random delays. Furthermore, several structured network architectures are studied as examples, and their associated simulation results are used to validate analytical findings. In the future, we intend to investigate the effects of skew, link failure, and other practical conditions when utilizing the DCTS algorithm in wireless sensor networks.

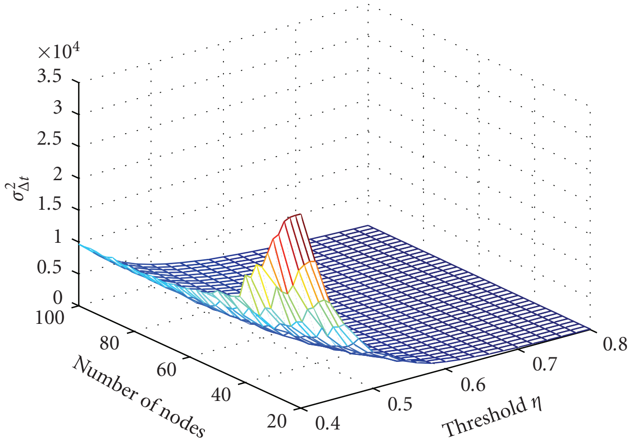

FIGURE 8: $\sigma^2_{\Delta t}$ as a function of the number of nodes and threshold η for the DCTS algorithm in random networks with Gaussian delay.

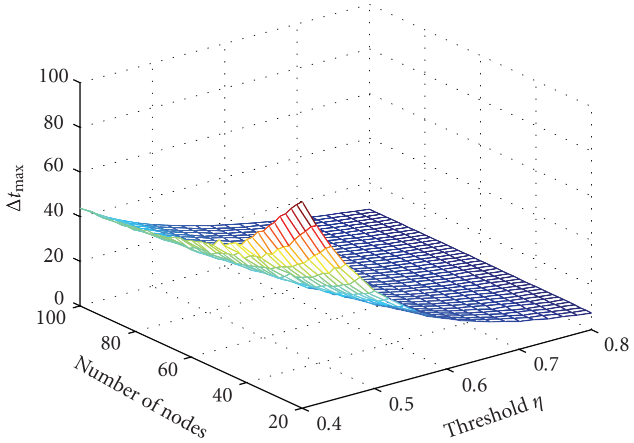

FIGURE 9: $\Delta t_{\max}$ as a function of the number of nodes and threshold η for the DCTS algorithm in random networks with Gaussian delay.

Appendices

A. Proof of Lemma 2

Proof. Define $\tilde{\boldsymbol{\delta}}(k) = \boldsymbol{\delta}(k) - \mathbb{E}[\boldsymbol{\delta}(k)]$. Then, the dynamics of this vector is given as follows:

$$\tilde{\boldsymbol{\delta}}(k) = P\tilde{\boldsymbol{\delta}}(k-1) + \varepsilon QA\mathbf{v}(k-1). \tag{A.1}$$

To prove this lemma, we can consider the evolution of covariance matrix of disagreement vector $\Sigma_\delta(k)$ instead since

$$\mathbb{E}\left[\tilde{\boldsymbol{\delta}}(k)^{\mathrm{T}}\tilde{\boldsymbol{\delta}}(k)\right] = \mathbf{tr}[\Sigma_\delta(k)] = \mathbf{tr}\left\{\mathbb{E}\left[\tilde{\boldsymbol{\delta}}(k)\tilde{\boldsymbol{\delta}}(k)^{\mathrm{T}}\right]\right\}. \tag{A.2}$$

Then, the proof of the lemma is equivalent to proving the following statement:

$$\Sigma_{\delta}(k) = P^{k}\boldsymbol{\delta}(0)\boldsymbol{\delta}(0)^{\mathrm{T}}P^{k} + \varepsilon^{2}\sigma^{2}\sum_{l=0}^{k-1}P^{l}QA^{2}QP^{l}, \quad (k \geq 1). \tag{A.3}$$

The statement is obviously true for $k = 1$. Now let us assume that the statement is true when $k = m$, $(m > 1)$, that is,

$$\Sigma_{\delta}(m) = P^{m}\boldsymbol{\delta}(0)\boldsymbol{\delta}(0)^{\mathrm{T}}P^{m} + \varepsilon^{2}\sigma^{2}\sum_{l=0}^{m-1}P^{l}QA^{2}QP^{l}. \tag{A.4}$$

When $k = m + 1$, we have

$$\begin{aligned}\Sigma_{\delta}(m+1) &= \mathbb{E}\left\{\left[P\widetilde{\boldsymbol{\delta}}(m)+\varepsilon QA\mathbf{v}(m)\right]\left[P\widetilde{\boldsymbol{\delta}}(m)+\varepsilon QA\mathbf{v}(m)\right]^{\mathrm{T}}\right\}\\ &= P^{m+1}\boldsymbol{\delta}(0)\boldsymbol{\delta}(0)^{\mathrm{T}}P^{m+1}\\ &\quad + \varepsilon^{2}\sigma^{2}\sum_{l=1}^{m}P^{l}QA^{2}QP^{l} + \varepsilon^{2}\sigma^{2}QA^{2}Q\\ &= P^{m+1}\boldsymbol{\delta}(0)\boldsymbol{\delta}(0)^{\mathrm{T}}P^{m+1} + \varepsilon^{2}\sigma^{2}\sum_{l=0}^{m}P^{l}QA^{2}QP^{l}.\end{aligned} \tag{A.5}$$

Therefore, $\Sigma_{\delta}(m + 1)$ has the exact same form as (A.3) for $k = m + 1$. Thus, (10) is valid, and we can conclude the proof. □

B. Proof of Theorem 2

Before proving the theorem, first we present some known results.

Theorem 3. *For any matrix A_1 and any symmetric matrix A_2, let $\overline{A}_1 = (A_1 + A_1^T)/2$, then one has [14]*

$$\sum_{i=1}^{n}\lambda_{n-i+1}\left(\overline{A}_1\right)\lambda_i(A_2) \leq \mathbf{tr}(A_1A_2) \leq \sum_{i=1}^{n}\lambda_i\left(\overline{A}_1\right)\lambda_i(A_2), \tag{B.1}$$

where $\lambda_i(\cdot)$ denotes the ith smallest eigenvalue of a matrix. In particular, if A_2 is a positive semidefinite matrix, one has

$$\lambda_1\left(\overline{A}_1\right)\mathbf{tr}(A_2) \leq \mathbf{tr}(A_1A_2) \leq \lambda_n\left(\overline{A}_1\right)\mathbf{tr}(A_2). \tag{B.2}$$

If A_1 is a positive semidefinite matrix, replacing A_1 with A_2 in (B.2), we have [15]

$$\lambda_1\left(\overline{A}_2\right)\mathbf{tr}(A_1) \leq \mathbf{tr}(A_1A_2) \leq \lambda_n\left(\overline{A}_2\right)\mathbf{tr}(A_1). \tag{B.3}$$

Combining (B.2) with (B.3), we have the following theorem.

Theorem 4. *If A_1 and A_2 are two positive semidefinite matrices, one has*

$$\begin{aligned}&\max\{\lambda_1(A_1)\mathbf{tr}(A_2), \lambda_1(A_2)\mathbf{tr}(A_1)\}\\ &\quad\leq \mathbf{tr}(A_1A_2)\\ &\quad\leq \min\{\lambda_n(A_1)\mathbf{tr}(A_2), \lambda_n(A_2)\mathbf{tr}(A_1)\}.\end{aligned} \tag{B.4}$$

We can now prove Theorem 2.

Proof. We know that the eigenvalues of $(L + K)^{-2}$ are 1 and $1/\lambda_i^2(L)$, $i = 2, \ldots, n$. Also, $\lambda_{\max}(Q) = 1$ and $\lambda_{\min}(Q) = 0$. Recall that

$$\begin{aligned}\lambda_i(W_2) &= \frac{1}{2\varepsilon\lambda_i(L) - \varepsilon^2\lambda_i^2(L)}\\ &= \frac{1}{2}\left[\frac{1}{\varepsilon\lambda_i(L)} + \frac{1}{2 - \varepsilon\lambda_i(L)}\right], \quad i = 2, \ldots, n.\end{aligned} \tag{B.5}$$

Since $\varepsilon \in (0, 2/\lambda_n(L))$, the eigenvalues of W_2 are nonnegative. Thus, $\lambda_{\min}(W_2) = 0$. In addition, W_2 and A^2 are positive semidefinite matrices with $\mathbf{tr}(A^2) = \mathcal{D}_n$. For a time delay unbalanced network, $Q\mathbf{u} \neq \mathbf{0}$. Based on (16) and (B.4), $\sigma_{\Delta t}^2$ is upper bounded by

$$\begin{aligned}\sigma_{\Delta t}^2 &\leq \frac{\|\mathbf{u}Q\|^2}{\min\{\lambda_2^2(L), 1\}} + \varepsilon^2\sigma^2\mathbf{tr}(W_2A^2)\\ &\leq \frac{\mathbf{u}^{\mathrm{T}}Q\mathbf{u}}{\min\{\lambda_2^2(L), 1\}}\\ &\quad + \varepsilon^2\sigma^2\min\{\lambda_{\max}(W_2)\mathbf{tr}(A^2), \lambda_{\max}(A^2)\mathbf{tr}(W_2)\}\\ &\leq \frac{\|\mathbf{u}\|^2}{\min\{\lambda_2^2(L), 1\}}\\ &\quad + \min\left\{\max\left\{\frac{\varepsilon\sigma^2\mathcal{D}_n}{2\lambda_i(L) - \varepsilon\lambda_i^2(L)}\right\}, \sum_{i=2}^{n}\frac{\varepsilon\sigma^2\lambda_{\max}(A^2)}{2\lambda_i(L) - \varepsilon\lambda_i^2(L)}\right\}.\end{aligned} \tag{B.6}$$

From [16], we know that $\lambda_n(L) \geq (n/(n-1))\max\{d_i\} > max\{d_i\} > 1$, $\forall i \in \mathcal{V}$. Then, $\sigma_{\Delta t}^2$ is lower bounded by

$$\begin{aligned}\sigma_{\Delta t}^2 &\geq \frac{\|\mathbf{u}Q\|^2}{\max\{\lambda_n^2(L), 1\}} + \varepsilon^2\sigma^2\mathbf{tr}(W_2A^2)\\ &\geq \frac{\mathbf{u}^{\mathrm{T}}Q\mathbf{u}}{\lambda_n^2(L)} + \varepsilon^2\sigma^2\max\{\lambda_{\min}(W_2)\mathbf{tr}(A^2), \lambda_{\min}(A^2)\mathbf{tr}(W_2)\}\\ &= \frac{\mathbf{u}^{\mathrm{T}}Q\mathbf{u}}{\lambda_n^2(L)} + \sum_{i=2}^{n}\frac{\varepsilon\sigma^2\lambda_{\min}(A^2)}{2\lambda_i(L) - \varepsilon\lambda_i^2(L)}.\end{aligned} \tag{B.7}$$

This completes the proof. □

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[2] A. Scaglione and R. Pagliari, "Non-cooperative versus cooperative approaches for distributed network synchronization," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW '07)*, pp. 537–541, White Plains, NY, USA, March 2007.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 2289–2294, New Orleans, La, USA, December 2007.

[5] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, Article ID 57054, 13 pages, 2007.

[6] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

[7] H. S. Abdel-Ghaffar, "Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays," *IEEE Transactions on Communications*, vol. 50, no. 10, pp. 1652–1661, 2002.

[8] J. Proakis, *Digital Communications*, McGraw-Hill, Boston, Mass, USA, 4th edition, 2000.

[9] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pp. 147–163, Boston, Mass, USA, December 2002.

[10] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[12] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC '03)*, vol. 5, pp. 4997–5002, Maui, Hawaii, USA, December 2003.

[13] W. C. Lindsey, F. Ghazvinian, W. C. Hagmann, and K. Dessouky, "Network synchronization," *Proceedings of the IEEE*, vol. 73, no. 10, pp. 1445–1467, 1985.

[14] J. B. Lasserre, "A trace inequality for matrix product," *IEEE Transactions on Automatic Control*, vol. 40, no. 8, pp. 1500–1501, 1995.

[15] W. Xing, Q. Zhang, and Q. Wang, "A trace bound for a general square matrix product," *IEEE Transactions on Automatic Control*, vol. 45, no. 8, pp. 1563–1565, 2000.

[16] B. Mohar, "Some applications of laplace eigenvalues of graphs," in *Graph Symmetry: Algebraic Methods and Applications*, G. Hahn and G. Sabidussi, Eds., vol. 497 of *NATO ASI Series C*, pp. 225–275, 1997.