



HAL
open science

Diffusive clock synchronization in highly dynamic networks

Matthias Függer, Thomas Nowak, Bernadette Charron-Bost

► **To cite this version:**

Matthias Függer, Thomas Nowak, Bernadette Charron-Bost. Diffusive clock synchronization in highly dynamic networks. 49th Annual Conference on Information Systems and Sciences (CISS 2015), Mar 2015, Baltimore, United States. 10.1109/CISS.2015.7086841 . hal-01231513

HAL Id: hal-01231513

<https://hal.science/hal-01231513v1>

Submitted on 20 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Diffusive Clock Synchronization in Highly Dynamic Networks

Matthias Függer
TU Wien
Vienna, Austria
fuegger@ecs.tuwien.ac.at

Thomas Nowak
École normale supérieure
Paris, France
thomas.nowak@ens.fr

Bernadette Charron-Bost
CNRS, École polytechnique
Palaiseau, France
charron@lix.polytechnique.fr

Abstract—This paper studies the clock synchronization problem in highly dynamic networks. We show that diffusive synchronization algorithms are well adapted to environments in which the network topology may change unpredictably. In a diffusive algorithm, each node repeatedly (i) estimates the clock difference to its neighbors via broadcast of zero-bit messages, and (ii) updates its local clock according to a weighted average of the estimated differences. The system model allows for drifting local clocks, running at possibly different frequencies.

We show that having a rooted spanning tree in the network at every time instance suffices to solve clock synchronization. We do not require any stability of the spanning tree, nor do we impose that the links of the spanning tree be known to the nodes. Explicit bounds on the convergence speed are obtained. In particular, our results settle an open question posed by Simeone and Spagnolini to reach clock synchronization in dynamic networks in the presence of nonzero clock drift. We also identify certain reasonable assumptions that allow for a significant higher convergence speed, e.g., bidirectional networks or random graph models.

I. INTRODUCTION

Maintaining a common notion of time between distributed network nodes is of great importance in distributed control and monitoring. It allows to globally order events, e.g., sensor readings, that locally occur at nodes, and relate their interarrival times to physical real-time. Application domains are numerous and reach from applications in sensor fusion [1] to setting up efficient inter-node communication layers, e.g., between base-stations [2] or moving vehicles [3]. Clock synchronization algorithms that create a common notion of time in such systems by synchronizing local hardware clocks face major challenges: *dynamicity* of the underlying communication network, e.g., due to transient link or node failures and moving nodes, and *drift*, i.e., frequencies of the nodes' hardware clocks that differ from each other and vary over time.

There is a large body of work on clock synchronization, see e.g., [4], [5], [6], [7]. Generally, clock synchronization algorithms can be partitioned into centralized approaches and decentralized approaches. The idea for the first kind is to determine a master node that distributes its clock over (a spanning subgraph of) the network, making all other nodes follow the master. Clearly, however, these approaches are susceptible to network changes, e.g., failure of the master, and at least require time to adapt to network changes. By

contrast, decentralized clock synchronization algorithms aim to overcome these restrictions.

In this work we focus on clock synchronization in highly dynamic networks, where no assumptions can be made on the time between network changes. In particular, classical methods, such as electing a leader node, constructing a spanning subgraph of the network etc. fail in such networks. Further, highly dynamic networks typically comprise of resource limited nodes and communication infrastructure, e.g., wireless battery-powered nodes attached to moving objects. Clearly, low communication complexity and low complexity/overhead of the clock synchronization algorithm itself are an issue in these systems. We will next give an overview of the work addressing such systems.

Related Work. Schneider [8] gave an overview on the work on clock synchronization in distributed computing that deals with fully connected communication networks in presence of different kinds of node failures. The idea of the presented algorithms is to periodically broadcast messages containing the nodes' current clock reading and to reach (approximate) agreement on a value within the range of the proposed clock readings. Each node then sets its clock on its local outcome of the agreement procedure. Dolev et al. [9] presented an algorithm that can cope with dynamic networks. However, they require that each correct node can communicate with each other correct node in bounded time; a property that is violated, e.g., in leader-based topologies like a directed star-graph.

Of specific interest in the context of low-resource networks is the study of non packet-based synchronization algorithms. By contrast, these algorithms communicate by broadcasting a zero-bit pulse only, conveying only the time of occurrence to its current neighboring nodes. Li and Rus [10] proposed to use diffusive clock synchronization algorithms where each node repeatedly sets its clock value to a weighted average of the neighboring clock values. Their algorithm can be seen as an instance of an algorithm that communicates by means of pulses only. However, Li and Rus analyzed the algorithm's behavior in case of fixed networks only. Simeone and Spagnolini [11] presented a synchronization algorithm subsuming several algorithms that communicate by means of pulses, including those in [2], [3], [10]. Their algorithm round-wise broadcasts pulses and adapts the time it generates a new

pulse according to a correction term equal to the weighted average of the received pulse times. They proposed to view and analyze this algorithm as coupled discrete-time oscillators of phase locked loops (PLLs). This point of view also lends itself to an efficient implementation in hardware. Simeone and Spagnolini considered the case of (i) static networks with potentially non-equal but constant node frequencies, and (ii) dynamic networks with constant and homogeneous clock frequencies. They further assumed bidirectional communication links between nodes. The analysis of dynamic networks with potentially non-equal constant node frequencies is stated as an open problem in their work.

Contributions. In this work we present a clock synchronization algorithm that communicates by means of pulses only. We show that the algorithm makes the nodes generate pulses within a bounded phase difference window, the skew, thus solving the clock synchronization problem. We do not only show solvability, but also derive precise analytic upper bounds on the skew size and the rate at which an initially large phase difference decreases. Our solution covers the case of dynamic networks with not necessarily bidirectional links and drifting hardware clocks, by that solving the open problem stated in [11]. Note that the study of solutions for non-bidirectional dynamic networks is especially interesting in the context of leader-based networks, e.g., those containing dedicated nodes with higher communication range. For the special case of bidirectional networks, the proposed solution reduces to the assumption that the network stay connected over time. We further present a method to drastically reduce convergence time of initially large phase differences. The analytic results are supported by simulations.

II. DIFFUSIVE CLOCK SYNCHRONIZATION WITH PULSES

We consider a dynamic network with a fixed set of nodes $[n] = \{1, \dots, n\}$. Computation evolves in rounds: In each round $k \geq 0$, every node $i \in [n]$ broadcasts a zero-bit pulse that is perceived by a subset of nodes. A node always receives its own pulses. Let us denote the real-time node i broadcasts its round k pulse by $t_i(k) \in \mathbb{R}$ and by $t(k)$ the column vector $^t(t_1(k), \dots, t_n(k))$. We further assume negligible communication delays in comparison to the times between successive pulses sent by a node, i.e., if a node receives the round k pulse sent by node i it does so at time $t_i(k)$.

The evolution of the dynamic network over time is modeled by a *communication pattern*, i.e., a sequence of directed graphs with self-loops at every node, each one of which we call a *communication graph*: The fact that the round k pulse generated by i is received by node j is modeled by a link from i to j in the k^{th} communication graph.

We assume that each node has access to a local clock by which it can measure progress of (real) time. Local clocks may, however, be inexact due to a *drift* $\varrho \in [0, 1)$, i.e., we only assume that a node's local clock requires at least time $(1 - \varrho)R$ and at most time $(1 + \varrho)R$ to advance by some

amount $R > 0$. Since the pulses do not carry any explicitly encoded information, a node i 's *algorithm* only determines the time i broadcasts its next pulse on the times of the received pulses. Note, however, that nodes do not have access to real time, but only to local clocks: i 's algorithm is not able to trigger a broadcast after some real time $R > 0$ passed, but only after i 's local clock advanced by some value R .

We do not make any assumptions on the stability of the underlying dynamic network: Communication patterns may be highly dynamic and change unpredictably. Let \mathcal{G} be a set of communication graphs. An algorithm *solves clock synchronization with skew* $\sigma \geq 0$ in \mathcal{G} if, for all communication patterns with communication graphs from \mathcal{G} , the sequence $(t(k))_{k \geq 0}$ induced by the communication pattern and the nodes' algorithm fulfills $\forall k \geq 0 : \max_{i,j} |t_i(k) - t_j(k)| \leq \sigma$. To rule out useless solutions, we further require that sequences $(t_i(k))_{k \geq 0}$ be increasing within some cone of real-time.

In this work, we study the ability of so-called *diffusive algorithms* to solve clock synchronization: A diffusive algorithm triggers the generation of the round $k + 1$ pulse of node i depending on the received round k pulses only. More specifically, after node i broadcast its round k pulse, it waits until its local clock has increased by a constant R adjusted by a correcting offset. This term ensures an offset between successive pulses that determines a node's pulse generation frequency in case it is not influenced by other nodes. It must be chosen large enough, in accordance with the propagation delay and the resulting clock synchronization skew, to ensure that round k and $k + 1$ pulses are well-separated, i.e., all nodes receive all round k pulses of their incoming neighbors before they broadcasts their round $k + 1$ pulses. The correcting term is a weighted average of the time differences of received round k pulses. We may thus write,

$$t_i(k + 1) = t_i(k) + T_i(k + 1) + \text{corr}_i(k + 1)$$

where $(1 - \varrho)R \leq T_i(k + 1) \leq (1 + \varrho)R$, and

$$\text{corr}_i(k + 1) = [\varepsilon_{k+1}]_i \sum_{j \in \text{In}_i(k+1)} [W_{k+1}]_{i,j} (t_j(k) - t_i(k)) \quad (1)$$

Weight $[\varepsilon_{k+1}]_i \in [0, 1)$ and weights $[W_{k+1}]_{i,j} \in [0, 1]$ for each $j \in \text{In}_i(k + 1)$ are locally chosen by i 's algorithm, such that $\sum_{j \in \text{In}_i(k+1)} [W_{k+1}]_{i,j} = 1$. For ease of notation, we set $[W_{k+1}]_{i,j} = 0$ for $j \notin \text{In}_i(k + 1)$. Note that without loss of generality we may assume that i 's algorithm always sets $[W_{k+1}]_{i,i} = 0$ if i receives at least one round k pulse from a remote node, and $[W_{k+1}]_{i,i} = 1$ otherwise.

Fig. 1 shows an example system of three nodes generating their round $k + 1$ pulses.

Example 1 (Switching communication graphs). Fig. 2 depicts the two communication graphs of an example dynamic network defined by the set of graphs $G = \{G_1, G_2\}$. Fig. 3 on page 4 shows the 10 round prefix of an execution of the diffusive algorithm where each node i chooses $[\varepsilon_{k+1}]_i = 1/2$ and $[W_{k+1}]_{i,j} = 1/(|\text{In}_i(k + 1)| - 1)$ if i received a round k pulse from $j \neq i$. Nodes 1 and 2 are assumed to have fast

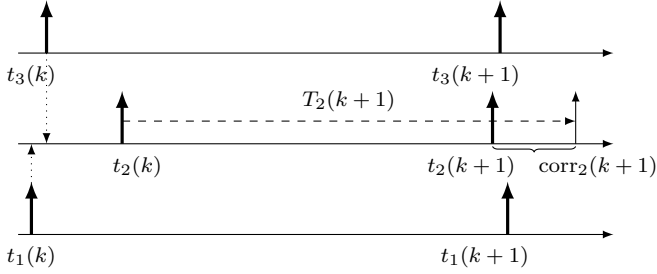


Fig. 1. Round k and $k+1$ pulses of a three node system. Node 2 adapts the time of occurrence of its round $k+1$ pulse by a negative offset $\text{corr}_2(k+1)$ due to the early pulses received from nodes 1 and 3.

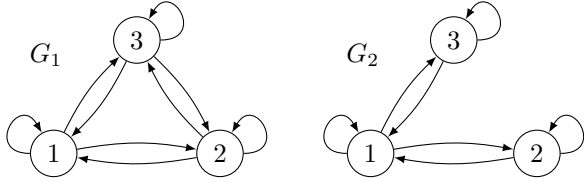


Fig. 2. Dynamic network with possible link loss between nodes 2 and 3.

local clocks, i.e., $T_1(k) = T_2(k) = (1 - \varrho)R$, while node 3 has a slow clock with $T_2(k) = (1 + \varrho)R$, where $R = 10$ and $\varrho = 0.2$. The execution in this example is induced by the above algorithm and the communication pattern $G(k) = G_1$ for $1 \leq k \leq 5$, and $G(k) = G_2$ for $k > 5$. Note that although the nodes' local clocks were assumed not to change frequency, the pulse generation frequency of all nodes changes with the loss of the link between nodes 2 and 3.

III. MATRIX FORMULATION

A. Linear Matrix Recurrence

Using basic algebraic manipulations, we may rewrite (1) in terms of a non-homogeneous linear matrix recurrence

$$t(k+1) = A_{k+1}t(k) + T(k+1), \quad (2)$$

where

$$[A_{k+1}]_{i,j} = \begin{cases} [\varepsilon_{k+1}]_i [W_{k+1}]_{i,j} & \text{if } i \neq j \\ 1 + [\varepsilon_{k+1}]_i ([W_{k+1}]_{i,i} - 1) & \text{if } i = j \end{cases}.$$

Note that each matrix A_k is stochastic, i.e., $\sum_j [A_k]_{i,j} = 1$ for all nodes i . Diffusive algorithms can equally be defined via the matrices A_k instead of ε_k and W_k . Note that $[A_k]_{i,j} > 0$ if and only if there is a link from j to i in the communication graph $G(k)$. We denote by γ the minimal positive entry of all matrices A_k during the execution. For the algorithm described in Example 1, we always have $\gamma \geq 1/2n$.

B. Dobrushin Semi-Norm

To assess the maximum phase differences of nodes in a given round, we use the Dobrushin semi-norm defined on real vectors x by $\delta(x) = \max_{i,j} |x_i - x_j|$ and apply it to $x = t(k)$.

An important tool is the matrix semi-norm associated to δ , mimicking the definition of the operator norm, i.e., setting

$$\delta(A) = \sup_{\substack{x \in \mathbb{R}^n \\ \delta(x) \neq 0}} \frac{\delta(Ax)}{\delta(x)}.$$

Indeed, both for vectors and for matrices, this defines a semi-norm. We recall several of its properties:

Lemma 2 ([12], [13]). *Let $A, B \in \mathbb{R}^{n \times n}$.*

- 1) *If A is stochastic, then we have the formula $\delta(A) = 1 - \min_{i_1, i_2} \sum_j \min\{A_{i_1, j}, A_{i_2, j}\}$.*
- 2) *$\delta(AB) \leq \delta(A)\delta(B)$*

We thus have $\delta(A) \leq 1$ whenever A is stochastic. Those stochastic matrices for which the strict inequality holds are called *scrambling* matrices. They are equivalently defined via the scrambling property, which demands that for all pairs (i_1, i_2) of indices there exist an index j such that both $A_{i_1, j} > 0$ and $A_{i_2, j} > 0$. The next lemma follows immediately from the definition of the scrambling property and Lemma 2(1) and is a central ingredient in our proofs.

Lemma 3. *Let A be a stochastic matrix with the scrambling property and minimal positive entry γ . Then, $\delta(A) \leq 1 - \gamma$.*

IV. RESULTS

A. Performance in Rooted Networks

For a fixed execution, denote the maximum phase difference of pulses in round k by $\delta(k) = \delta(t(k))$. The skew of a specific execution is hence $\sigma = \sup_k \delta(k)$.

We will next analyze the behavior of the maximum phase differences $\delta(k)$ evolving over rounds k and valid conditions on the skew in presence of two kinds of networks: (i) those where all communication graphs are *non-split*, and (ii) those where they are *rooted*. Hereby a communication graph is called non-split if all pairs of nodes have a common incoming neighbor, and it is rooted if there exists a node i such that there is a directed path from i to all nodes. A stochastic matrix has the scrambling property if and only if the corresponding communication graph is non-split. While being non-split is a strong property, which is typically not fulfilled in dynamic networks, it plays an important role in the analysis for rooted networks.

Given two graphs G and H on the same node set, we define their composition graph $G \circ H$ by including a link (i, j) if there is a node k such that the link (i, k) is in G and the link (k, j) is in H . The following lemma shows that a sufficiently large number of rooted communication graphs always compose to a non-split graph and explicitly bounds this number.

Lemma 4 ([14, Proposition 8]). *The composition of any $n-1$ rooted communication graphs on n nodes is non-split.*

We will thus start with the non-split case and then reduce arbitrary rooted networks to the non-split case. It is important to note that while the set of roots in Fig. 2 is the same for G_1 and G_2 , we do not require this in our analysis: roots may

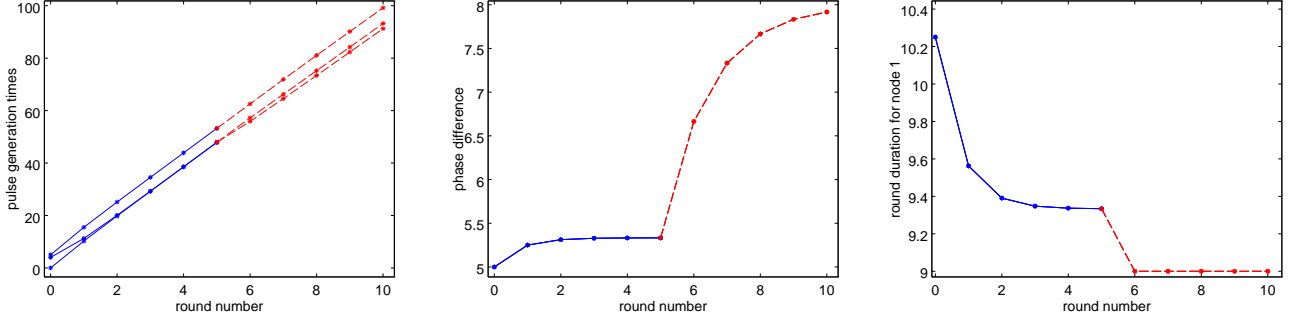


Fig. 3. Prefix of an execution of the example dynamic network for rounds $0 \leq k \leq 10$. The communication graph changes from G_1 (solid lines) to G_2 (dotted lines) at round 5. The figure shows (i) pulse generation times $t_i(k)$, (ii) maximum phase differences $\max_{i,j} |t_i(k) - t_j(k)|$, and (iii) round durations $t_1(k+1) - t_1(k)$.

change arbitrarily and without any stability assumption during an execution.

Theorem 5. *If every communication graph has the non-split property, then every diffusive algorithm solves clock synchronization with skew $\sigma = \max\{\delta(0), 2\rho R/\gamma\}$. Moreover*

$$\delta(k) \leq (1 - \gamma)^k \delta(0) + 2\rho R/\gamma$$

for all $k \geq 0$. In particular, the maximum phase difference is arbitrarily close to $2\rho R/\gamma$ for large round numbers k .

Assuming communication graphs to be non-split is quite restrictive, and typically not the case if nodes are physically distributed within a larger area. The following result shows that even in the general case of rooted communication graphs with unpredictable root changes, clock synchronization can be solved. The main point is to show that those systems can implicitly simulate non-split communication graphs and then apply Theorem 5.

Theorem 6. *If every communication graph is rooted, then every diffusive algorithm solves clock synchronization with skew $\sigma = \max\{\delta(0) + 2\rho Rn, 2\rho Rn/\gamma^n\}$. Moreover*

$$\delta(k) \leq (1 - \gamma^n)^{\lfloor k/n \rfloor} \delta(0) + 2\rho Rn/\gamma^n$$

for all $k \geq 0$. In particular, the maximum phase difference is arbitrarily close to $2\rho Rn/\gamma^n$ for large round numbers k .

B. Stable Common Frequency and Improved Convergence in Bidirectional Networks

Certain diffusive algorithms executed in networks with bidirectional communication graphs cannot only be shown to converge faster (Theorem 7), but also not to lead to different round durations due to network changes (Theorem 8). We study the class of *fixed weight algorithms*, that is, diffusive algorithms where i weighs all incoming pulses from $j \neq i$ with a constant weight $c > 0$.

Theorem 7. *If every communication graph is bidirectional and connected, then the fixed weight averaging algorithm with parameter c solves clock synchronization with $\sigma = \max\{n^{1/2}\delta(0), 4\rho Rn^{5/2}/c\}$. Moreover*

$$\delta(k) \leq n^{1/2}(1 - c/2n^2)^k \delta(0) + 4\rho Rn^{5/2}/c$$

for all $k \geq 0$. In particular, the maximum phase difference is arbitrarily close to $4\rho Rn^{5/2}/c$ for large round numbers k .

We next show that unless the nodes' local clocks change their frequency, running a fixed weight averaging algorithm in bidirectional networks leads to no round duration alternations due to network changes.

Theorem 8. *If every communication graph is bidirectional and connected, local clocks do not change frequency, i.e., $T(k) = T$ for all $k \geq 1$, and all nodes execute the fixed weight averaging algorithm with parameter c , then each node i 's average round duration $t_i(k)/k$ converges to $\sum_{j=1}^n T_j/n$.*

Example 9 (Stable common frequency). Fig. 4 on page 5 shows the 10 round prefix of an execution of the fixed weight diffusive algorithm presented at the beginning of the section. Local clock speeds and the communication pattern were chosen as in Fig. 3. Observe that although the communication graph changes from round 5 to 6, node 1's round duration does not stabilize to a different value, unlike it is the case in Fig. 3.

C. Fast Stabilization in Stable Environments

In this section we will show that under the assumption of small local drift and stable networks, the nodes' phase differences converge fast to a slowly varying offset, whose maximum value can be bounded by Theorems 5 and 6.

While Theorem 6 shows that clock synchronization is solvable in network models whose only constraint is that communication graphs are rooted, without any restrictions on (i) the stability of the network or (ii) the nodes' frequencies, in the context of many applications assumptions on (i) or (ii) can be made. Concerning (ii) it is often the case that the amount by which the frequencies of distinct nodes differ is significantly larger than the amount by which a single node changes frequency from one round to the next; both effects are currently subsumed under the notion of drift, yielding $0 \leq |T_i(k+1) - T_i(k)| \leq 2\rho R$. To allow for a finer analysis, we strengthen this assumption by requiring $0 \leq |T_i(k+1) - T_i(k)| \leq 2\rho'R$ for some non-negative local drift $\rho' < \rho$.

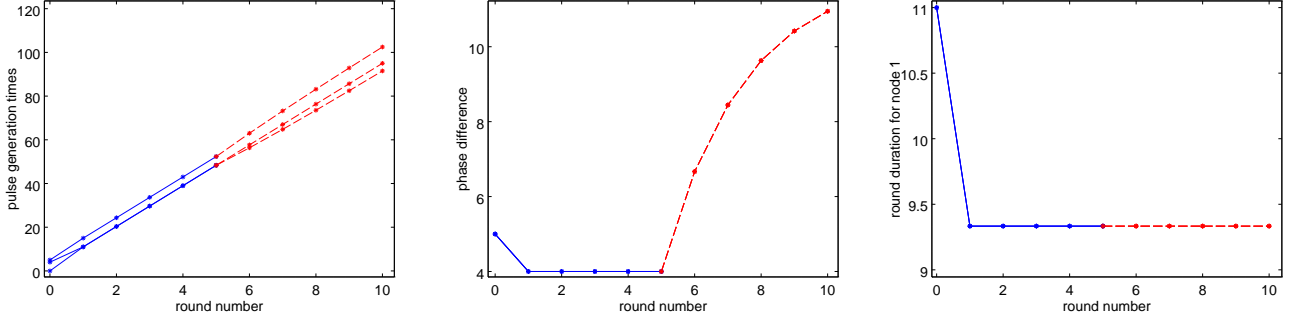


Fig. 4. Prefix of an execution of a diffusive algorithm with fixed weights run in the example dynamic network of Fig. 3. Solid lines represent rounds k at which $G(k) = G_1$, and dotted lines those where $G(k) = G_2$.

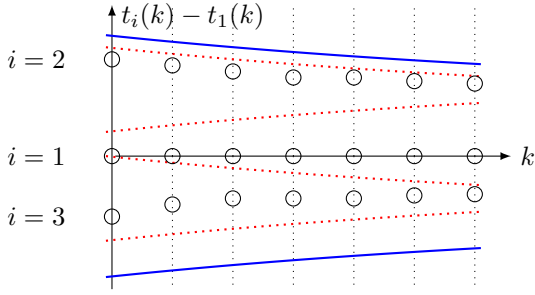


Fig. 5. Phase differences of three nodes to node 1 over rounds k in stable environments. The outer solid hull follows from Theorem 5. The inner dotted hulls follow from Theorem 10, showing that phase differences cannot vary arbitrarily within the outer hull.

For every stochastic vector π and every vector X , we write $\mathbb{E}_\pi X = \sum_i \pi_i X_i$. For a matrix M , we denote by M^\dagger its pseudo-inverse. In particular, it satisfies $MM^\dagger M = M$. Set $\Delta T(k) = T(k) - \mathbb{E}_{\pi_k} T(k) \cdot \mathbf{1}$, $\tau(k) = t(k) - \sum_{\ell=1}^k \mathbb{E}_{\pi_\ell} T(\ell)$, and $\tau'(k) = \tau(k) - L_{k+1}^\dagger \Delta T(k+1)$.

Theorem 10. *Assume that every communication graph is non-split and strongly connected, and local drift is bounded by ρ' . Further assume that A_k does not change during rounds $1 \leq k \leq K$, i.e., neither the network nor the algorithm's weights change during those rounds. Then*

$$\delta(\tau'(k)) \leq (1 - \gamma)^k \delta(\tau'(0)) + 2\rho' R n / \gamma^{n+1}$$

for all $k < K$.

We can use Theorem 10 to effectively assess the speed of convergence of its phase differences after k rounds and Theorem 5 to bound its maximum phase differences $\delta(k)$ after k rounds. Fig. 5 visualizes the effect of both bounds.

The reduction arguments given in the proof of Theorem 6 can be applied to Theorem 10 as well, yielding analogous results on the convergence speed in the general strongly connected case.

D. Expander Networks

If, in addition to be bidirectional, the communication graphs are expanders, we can achieve an even better skew of the synchronized clocks. In random networks models, communication

graphs are typically expanders; in particular in the Erdős-Rényi model.

The *Cheeger constant* of a bidirectional directed graph $G = (V, E)$ is equal to

$$h(G) = \min_{\substack{S \subset V \\ 0 < |S| \leq |V|/2}} \frac{|E(S, S^c)|}{|S|}$$

where $E(A, B)$ denotes the set of links between the sets of nodes A and B and S^c denotes the complement of S in V . A bidirectional directed graph G is a λ -*expander* if $h(G) \geq \lambda$.

Theorem 11. *If every communication graph is bidirectional, connected and a λ -expander, then the fixed weight averaging algorithm with parameter c solves clock synchronization with $\sigma = \max \{n^{1/2} \delta(0), 4\rho R / \lambda^2 c^2\}$. Moreover*

$$\delta(k) \leq n^{1/2} (1 - \lambda^2 c^2 / 2)^k \delta(0) + 4\rho R / \lambda^2 c^2$$

for all $k \geq 0$. In particular, the maximum phase difference is arbitrarily close to $4\rho R / \lambda^2 c^2$ for large round numbers k .

V. SIMULATIONS

To back up our analytical results, we performed simulations in Matlab. For that purpose we generated sequences of random communication graphs $G(k)$ that are not necessarily bidirectional, and where each link (i, j) , with $i \neq j$, appears in $G(k)$ with probability 0.6. Fig. 6 on page 6 shows the prefix of an example execution for a system of $n = 5$ nodes running the fixed weight diffusive algorithm with parameter $c = 1/10$; and Fig. 6 for a larger network of $n = 20$ nodes and parameter $c = 1/20$. In both scenarios, $R = 10$, $\rho = 0.2$, and $T_i(k)$ were uniformly chosen within $[(1 - \rho)R, (1 + \rho)R]$.

From our experiments we observed that despite the relatively large drift the convergence rates of the phase differences and the actual skew obtained in all performed experiments were significantly better than the derived worst case bounds. We hope to be able to close this gap in future work.

VI. CONCLUSION

This paper dealt with clock synchronization in networks whose topology may change unexpectedly at every communication round. We showed that diffusive algorithms are able to solve clock synchronization if, at every instant, the network is

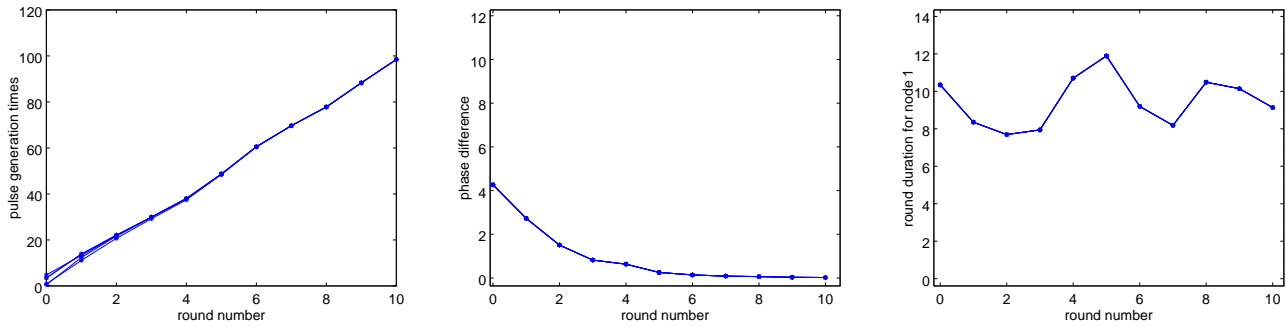


Fig. 6. Prefix of an execution of the example dynamic network of size $n = 5$ for rounds $0 \leq k \leq 10$. Communication graph $G(k)$ are directed random graphs, with probability 0.6 of a link to be in $G(k)$. We run the fixed weight algorithm with parameter $c = 1/10$.

rooted, i.e., contains a directed spanning tree. The root of this tree does not need to be known to processes, nor does it need to be unique or stable. Our model also allows for clock drift. We provided analytic upper bounds on the clock skew and the stabilization time. Moreover, we identified special cases in which we were able to bound them more tightly and showed that, in simulations, convergence is very fast.

For future work, we plan to study the discrepancy between the obtained analytic bounds and the simulation results, as well as perform real-world experiments with sensor motes.

REFERENCES

- [1] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Comput. Surv.*, vol. 39, no. 3, 2007.
- [2] F. Tong and Y. Akaiwa, "Theoretical analysis of interbase-station synchronization systems," *Communications, IEEE Transactions on*, vol. 46, no. 5, pp. 590–594, May 1998.
- [3] E. Sourour and M. Nakagawa, "Mutual decentralized synchronization for intervehicle communications," *Vehicular Technology, IEEE Transactions on*, vol. 48, no. 6, pp. 2015–2027, Nov 1999.
- [4] D. Mills, "Internet time synchronization: the network time protocol," *Communications, IEEE Transactions on*, vol. 39, no. 10, pp. 1482–1493, Oct 1991.
- [5] F. Cristian, "Probabilistic clock synchronization," *Distributed Computing*, vol. 3, no. 3, pp. 146–158, 1989.
- [6] H. Kopetz and W. Ochsreiter, "Clock synchronization in distributed real-time systems," *Computers, IEEE Transactions on*, vol. C-36, no. 8, pp. 933–940, Aug 1987.
- [7] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [8] F. B. Schneider, "Understanding protocols for byzantine clock synchronization," Ithaca, NY, USA, Tech. Rep., 1987.
- [9] D. Dolev, J. Y. Halpern, B. Simons, and R. Strong, "Dynamic fault-tolerant clock synchronization," *J. ACM*, vol. 42, no. 1, pp. 143–185, Jan. 1995.
- [10] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–226, Feb. 2006.
- [11] 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, no. 1, p. 057054, 2007.
- [12] R. L. Dobrushin, "Central limit theorem for non-stationary Markov chains I," in *Theory of Probability and its Applications*. American Mathematical Society, 1956, vol. 1, no. 1, pp. 65–80.
- [13] B. Charron-Bost, "Orientation and connectivity based criteria for asymptotic consensus," 2013, <http://arxiv.org/abs/1303.2043>.
- [14] M. Cao, A. S. Morse, and B. D. O. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM Journal on Control and Optimization*, vol. 47, no. 2, pp. 575–600, 2008.