UDC 519.248: 62–192

Agreement algorithms for synchronization of clocks in nodes of stochastic networks

L. Manita[*], **A. Manita**[†]

[*] *National Research University Higher School of Economics, Moscow Institute of Electronics and Mathematics, Russia*
[†] *Lomonosov Moscow State University, Faculty of Mathematics and Mechanics, Russia*

Abstract. We propose deterministic and stochastic models of clock synchronization in nodes of large distributed network locally coupled with a reliable external exact time server.

Keywords: load balance, design of reliable networks, steady state.

1. Introduction

We study a special class of multi-dimensional stochastic processes motivated by the problem of synchronization of local clocks in asynchronous networks. Proposed models are based on network consensus algorithms. Such algorithms have a variety of applications in computer science and social network dynamics [1–5]. Stochastic models, in the presence of random noise in nodes, have much in common with Markovian queueing networks or with probabilistic models of communication networks. We prove that values of the node clocks converge in distribution and find some properties of the steady state.

2. Clock synchronization models

Network. Let indices $j \in \{1,\ldots,N\} := \mathcal{N}$ mark nodes of a network. The nodes communicate in a manner to be precised below. Possibility of direct communications between pairs of nodes is described by an $N \times N$-matrix $W = (w_{jk})_{j,k\in\mathcal{N}}$ with nonnegative off-diagonal entries. The meaning of the entry w_{jk}, $j \neq k$, is the measure of importance of an opinion (or a state) of node k for node j. Consider a digraph $\mathcal{G} = (\mathcal{N},\mathcal{E})$ with the set of vertices $\mathcal{N}$ and the set $\mathcal{E}$ of directed edges (m_1, m_2) such that $w_{m_1 m_2} > 0$, $m_1 \neq m_2$. Denote by $\mathcal{O}_j := \{k : w_{jk} > 0\}$ the set of neighbours of a node j in the graph $\mathcal{G}$. If $|\mathcal{O}_j| \ll |\mathcal{N}|$ for all j then the network is called distributed. Similarly, any algorithm that uses only local exchance of information between neighbouring nodes is called distributed.

In the sequel elements of $\mathcal{N}$ will be called client nodes. There is also a stand-alone time server formally not belonging to the network but being reffered in the sequel as the node 0. The system is observed at discrete times $t = 0,1,\ldots$. The variable t will also count steps of distributed algorithms. For this reason we introduce notation $\mathbb{T} := \{1,2,\ldots\}$.

Clocks. Any client node $j \in \mathcal{N}$ is equipped with an unperfect clock which current value is $\tau_j \in \mathbb{R}$. The time server (the node 0) has a perfect clock providing exact time $\tau_0 \in \mathbb{R}$. Let $v > 0$ be the rate of the clocks with respect to t. Evolutions of isolated clocks are

$$\tau_0(t) = \tau_0(t-1) + v, \qquad \tau_j^\circ(t) = \tau_j^\circ(t-1) + v + \delta_j(t), \quad j \in \mathcal{N},$$

where $\delta(t) = (\delta_j(t),\ j = 1, \ldots, N) \in \mathbb{R}^N,\ t \in \mathbb{T}$, are i.i.d. random vectors representing random noise related to unperfect clocks of client nodes. We assume that $\mathsf{E}\,\delta_j(t) = 0$. Let $B = \mathsf{Var}\,\delta(t) = \left(\mathsf{cov}(\delta_j(t), \delta_k(t))\right)_{j,k\in\mathcal{N}}$ denote the corresponding covariance matrix. Since rates of clients are equal to the server rate v the above system of clocks is drift-free.

Algorithm for internal synchronization of the network. To synchronize clocks in nodes of the distributed system one can run a linear iterating algorithm (LIA) similar to that is used for the *network consensus* problem (NC problem). Namely, for any $j \in \mathcal{N}$

$$\tau_j(t) = \tau_j(t-1) + v + \delta_j(t) + \sum_{k \neq j} w_{jk} \left(\tau_k(t-1) - \tau_j(t-1)\right). \qquad (1)$$

The sum in (1) is in fact taken over $k \in \mathcal{O}_j$ and so (1) can be regarded as distributed algorithm. The last summand in (1) is a correction made by the node j by using information on local time values $\tau_k(t-1)$ obtained from the neighbourhood $\mathcal{O}_j$. It is convenient to put $w_{jj} = 1 - \sum\limits_{k\neq j} w_{jk}$ and to rewrite equations in matrix form $\boldsymbol{\tau}(t) = W\boldsymbol{\tau}(t-1) + v\mathbf{1} + \boldsymbol{\delta}(t)$ where $\boldsymbol{\tau}$, $\mathbf{1}$ and $\boldsymbol{\delta}$ are column vectors of length N. In the study of iterations (1) the behavior of powers W^t as $t \to \infty$ plays a crucial role. While it is not necessary for general LIAs, it is natural in the context of NC problem to assume that w_{jj} are *all nonnegative*. We will adopt this assumption throughout the rest of the paper. Thus the matrix W is stochastic. Hence the Markov chain theory as well as the Perron-Frobenius theory are very useful here. Recall that a matrix A is called *primitive*[1] if there exists $t_0 \in \mathbb{N}$ such that $A^{t_0} > 0$, i.e., all entries of A^{t_0} are positive.

Theorem 1. *Let the matrix W be primitive, $\boldsymbol{\tau}(t)$ be evolving according to the agreement algorithm (1) and j_0 and k_0 be a pair of nodes. Then*

i) in the deterministic case ($B = 0$)

$$\tau_{j_0}(t) - \tau_{k_0}(t) \to 0 \qquad (t \to \infty), \qquad (2)$$

ii) for the stochastic model ($B \neq 0$)

$$\limsup_t \mathsf{E}\,\left|\tau_{j_0}(t) - \tau_{k_0}(t)\right|^2 \leq C(B, W) \qquad (3)$$

[1] This corresponds to the notion of ergodic matrix in the Markov chain theory.

where $C(B,W) > 0$ does not depend on $\boldsymbol{\tau}(0)$ and vanishes as $B \to 0$.

Clearly, the results (2) and (3) mean that after a large number t of steps in (1) all clocks τ_j show, in some sense, a common time. Differences of clock values at different nodes of the distributed network vanish as in (2) or become satisfactory small as in (3). After introducing a new vector $\boldsymbol{\tau}'(t) = \boldsymbol{\tau}(t) - vt\mathbf{1}$ the iterating scheme (1) turns into

$$\boldsymbol{\tau}'(t) = W\boldsymbol{\tau}'(t-1) + \boldsymbol{\delta}(t) \qquad (4)$$

which is a stochastic version of the network consensus algorithm [1,3,6,7]. Now the both statements of Theorem 1 easily follows from already known results on the NC problem. The item i) is just reformulation of the main result from [1] and the item ii) can be extracted from [6]. The use of the word *consensus* here is related to the convergence $\boldsymbol{\tau}'(t) \to c\mathbf{1}$ taking place under assumptions $B = 0$ and primitivity of W. For self-evident reasons, $\{c\mathbf{1},\, c \in \mathbb{R}\}$ is called a consensus subspace of $\mathbb{R}^n$.

To check primitivity of W for the network models we need sufficient conditions, which can be easily verified.

Assumption WI: The matrix W is irreducible or, equivalently, the digraph $\mathcal{G}$ is strongly connected (see [8, Sect. 6.2]).

Assumption WA: There exists $j_0 \in \mathcal{N}$ such that $w_{j_0 j_0} > 0$.

It is well known that validity of WI+WA implies the primitivity of W. The Perron theorem states that $\lambda_1 = 1$ is a simple eigenvalue of any primitive stochastic matrix W. Moreover, it states that $\lim_t W^t = \mathbf{1}\pi^W$. Here a row π^W is the left eigenvector of W corresponding to $\lambda_1 = 1$ and normalized as $\pi^W\mathbf{1} = 1$. We know from the Perron theorem that π^W has all components positive. Hence $\mathbf{1}\pi^W$ is a positive $N \times N$-matrix of rank one. Finally, we conclude that for any column vector $\boldsymbol{\rho} \in \mathbb{R}^N$

$$W^t\boldsymbol{\rho} \to A(\boldsymbol{\rho})\mathbf{1} \qquad (t \to \infty) \qquad (5)$$

where $A(\boldsymbol{\rho}) := \pi^W\boldsymbol{\rho} = \sum_{j=1}^N \pi_j^W \rho_j$.

Deviation from the time server. Nevertheless, in the both situations i) and ii) of Theorem 1 the client clocks are far from the *accurate time* provided by the time server 0. To see this consider deviations $x_j = \tau_j - \tau_0$. The vector $\boldsymbol{x}(t) = (x_1(t), \ldots, x_N(t))$ evolves in the same way as (4),

$$\boldsymbol{x}(t) = W\boldsymbol{x}(t-1) + \boldsymbol{\delta}(t), \qquad t \in \mathbb{T},$$

so all above arguments are applicable. For example, applying (5) to the deterministic case $\boldsymbol{\delta} = 0$ we get $\tau_j(t) - \tau_0(t) \to A(\boldsymbol{\tau}(0)) - \tau_0(0)$ as $t \to \infty$. Note that initial values $\boldsymbol{\tau}(0)$ of the clients clocks are unknown.

3. Interaction with the time server

To solve a problem of synchronization of client clocks $\tau_j(t)$ with the accurate time $\tau_0(t)$ we propose a *modification* of the model (1). The *first* novelty is that the time server 0 can address messages to some (but not to all) client nodes. The schedule of this messaging will be precised later. A message $m^{0\to j'}$ sent on some step $t = t'$ from 0 to j' contains the value $\tau_0(t')$. It is assumed that $m^{0\to j'}$ instantly reaches the destination node. After receiving this message the node j' immediately adjusts its clock to the value recorded in $m^{0\to j'}$: $\tau_{j'}(t') = \tau_0(t')$. This is the usual message passing mechanism with zero delays. If on step t there is no message from 0 to j then the clock value $\tau_j(t)$ is adjusted according to the jth row of (1). The *second* novelty reflects the assumption that a node just received a message from the time server *is aware* that its newly adjusted clock value is more precious than ones of its neigbours. If on step t_1 a client clock τ_{j_1} was set to the value $\tau_0(t_1)$ then during the time interval $t_1+1,\dots,t_1+\Delta_{j_1}$ the node j_1 decides to ignore opinions of its neighbours. Denote by $T^{0,j} = \left\{t_n^{(j)}\right\}_{n\in\mathbb{N}}$ the sequence of steps t when the node j receives messages from the node 0. Hence to define the modified model one need to specify a set of message flows $T :=(T^{0,j},\ j\in\mathcal{N})$ and a set $\Delta :=(\Delta_j,\ j\in\mathcal{N})$. Consider $\mathcal{T} = \left\{j:\ T^{0,j}\neq\varnothing\right\}$, the set of nodes receiving messages from the time server node 0. So $\mathcal{T}\subset\mathcal{N}$ and the most interesting case is that where the set $\mathcal{T}$ is only a small part the whole distributed network.

The evolution of $x_j = \tau_j - \tau_0$ can be written as

$$\boldsymbol{x}(s) = W(s)\,\boldsymbol{x}(s-1) + D(s)\boldsymbol{\delta}(s), \qquad s\in\mathbb{T}. \tag{6}$$

Here $W(s) = W(s;T,\Delta)$ is a time-dependent $N\times N$-matrix with entries $w_{jk}(s)$ such that $w_{jk}(s)\in\{0,w_{jk},1\}$. The concrete value of $w_{jk}(s)$ is uniquely determined by the above description of the algorithm. Notation $D(s) = D(s;T)$ stands for the diagonal matrix of 0s and 1s, $D(s) := \mathrm{diag}(\mathbb{I}_{\{s\notin T^{0,j}\}},\ j\in\mathcal{N})$, indicating that the random noise term $\delta_j(s)$ are not added to $\tau_j(s)$ on the steps $s = t_n^{(j)}$. From the general point of view the model (6) is a special subclass of distributed LIAs with time-dependent topologies and the time-nonhomogeneous random noise [2].

Scheduling sequence. Let the set Δ be fixed. For any $s\in\mathbb{T}$ define $\mathcal{R}(s) := \left\{j \mid T^{0,j}\ni s\right\}\subset\mathcal{N}$, the subset of client nodes receiving messages from the time server node 0 on the step s. Evidently, $T = (T^{0,j},\ j\in\mathcal{N})$ and $\mathcal{R} = \{\mathcal{R}(s),\ s\in\mathbb{T}\}$ uniquely determine each other. From the viewpoint of the node 0 the set $\mathcal{R}$ defines a prescribed sequence of recipients $\mathcal{R}(s)$ to whom it should consequently send messages.

Assumption SP: The set $\mathcal{T}$ (and hence $\mathcal{R}$) is not empty. The scheduling sequence $\mathcal{R}$ is periodic with period d, i.e., $\mathcal{R}(s+d) = \mathcal{R}(s)$, $s \in \mathbb{T}$.

If $\mathcal{R}$ is periodic with period d, then the sequence $\{D(s)\}_{s\in\mathbb{T}}$ is periodic too. Moreover, $\{W(s)\}_{s\geq s_0}$ is d-periodic for sufficiently large $s_0 = s_0(\mathcal{R}, \Delta)$.

Theorem 2. *Consider the deterministic model $\boldsymbol{x}(s) = W(s)\,\boldsymbol{x}(s-1)$, $s \in \mathbb{T}$, with periodic scheduling (Assumption SP). Then under Assumptions WI and WA all clients synchronize with the node* 0

$$x_j(s) = \tau_j(s) - \tau_0(s) \to 0 \qquad (s \to \infty).$$

This theorem is similar to results on the NC problem in the presence of leaders [2, 5]. Now let $\boldsymbol{x}(s)$ be the stochastic model (6) started from nonrandom $\boldsymbol{x}(0)$. In Theorems 3 and 4 we make the additional assumption that W is double stochastic. Consider $\boldsymbol{m}(s) = \mathsf{E}\,\boldsymbol{x}(s)$ and the covariance matrix $\mathsf{Var}\,\boldsymbol{x}(s)$. Let $\|A\|_2$ denode the spectral norm of a matrix A (see [8]).

Theorem 3. *The mean vector $m(s)$ follows the equation for the deterministic model studied in Theorem 2, in particular, $\boldsymbol{m}(s) - \tau_0(s)\mathbf{1} \to 0$. The covariance matrix is uniformly bounded, i.e., $\sup_{s\in\mathbb{T}} \|\mathsf{Var}\,\boldsymbol{x}(s)\|_2 \leq C\,\|B\|_2$.*

Theorem 4. *For any $i \in \{0, \ldots, d-1\}$ the subsequence $\{\boldsymbol{x}(nd+i)\}_{n\in\mathbb{N}}$ has a limit in distribution as $n \to \infty$. These limit distributions can be characterized explicitly. In general, they differs for different i.*

References

1. *DeGroot M. H.* Reaching a consensus // Journal of the American Statistical Association. — 1974. — Vol. 69, no. 345. — P. 118–121.
2. *Jadbabaie A., Lin J., Morse A. S.* Coordination of groups of mobile autonomous agents using nearest neighbor rules // IEEE Transactions on automatic control. — 2003. — Vol. 48, no. 6. — P. 988–1001.
3. *Olshevsky A., Tsitsiklis J. N.* Convergence speed in distributed consensus and averaging // SIAM Review. — 2011. — Vol. 53, — P. 747–772.
4. *Estrada E., Vargas-Estrada E.* How peer pressure shapes consensus, leadership, and innovations in social groups // Scientific reports. — 2013. — Vol. 3.
5. *Lin F., Fardad M., Jovanovic M. R.* Algorithms for leader selection in stochastically forced consensus networks // IEEE Transactions on Automatic Control. — 2014. — Vol. 59, no. 7. — P. 1789–1802.
6. *Cybenko G.* Dynamic load balancing for distributed memory multiprocessors. // J. Parallel Distrib. Comput. — 1989. — Vol. 7, no. 2. — P. 279–301.
7. *Xiao L., Boyd S. P., Kim S.-J.* Distributed average consensus with least-mean-square deviation. // J. Parallel Distrib. Comput. — 2007. — Vol. 67, no. 1. — P. 33–46.
8. *Horn R. A., Johnson C. R.* Matrix Analysis. — Cambridge University Press, 1986.