Methodology and Computing in Applied Probability (2021) 23:1023–1056
https://doi.org/10.1007/s11009-020-09792-z

Upper and Lower Bounds for the Synchronizer Performance in Systems with Probabilistic Message Loss

Martin Zeiner[1] · Ulrich Schmid[1]

Received: 10 July 2019 / Revised: 22 April 2020 / Accepted: 28 April 2020 /
Published online: 20 June 2020

Abstract

In this paper, we revisit the performance of the α-synchronizer in distributed systems with probabilistic message loss as introduced in Függer et al. [Perf. Eval. 93(2015)]. In sharp contrast to the infinite-state Markov chain resp. the exponential-size finite-state upper bound presented in the original paper, we introduce a polynomial-size finite-state Markov chain for a new synchronizer variant α', which provides a new upper bound on the performance of the α-synchronizer. Both analytic and simulation results show that our new upper bound is strictly better than the existing one. Moreover, we show that a modified version of the α'-synchronizer provides a lower bound on the performance of the α-synchronizer. By means of elaborate simulation results, we show that our new lower bound is also strictly better than the lower bound presented in the original paper.

Keywords Distributed systems · Synchronizer · Performance analysis · Probabilistic message loss · Markov chain

Mathematics Subject Classification (2010) 60J20 · 60J10 · 68Q87 · 68W15

1 Introduction

Simulating synchronous executions in a distributed message-passing system is a well-known and powerful design approach. Synchronizers like the well-known α-synchronizer by Awerbuch (Awerbuch 1985) allow to establish a virtual (lock-step) round structure, which greatly simplifies the design of higher-level distributed algorithms. Moreover, it

This work has been supported the Austrian Science Fund (FWF) project S11405 (RiSE) and P28182 (ADynNet).

✉ Martin Zeiner
mzeiner@ecs.tuwien.ac.at

Ulrich Schmid
s@ecs.tuwien.ac.at

[1] TU Wien, ECS, Treitlstraße 3/II, 1040 Wien, Austria

makes it easy to reason about the time complexity of an atop-running algorithm, which is just the number of rounds needed until termination.

The underlying idea of the α-synchronizer is to let processes continuously exchange round numbers and to allow a process to proceed to the next round only after it has witnessed that all processes have already started the current round.

Given the exploding number of distributed systems that are interconnected by wireless networks, ranging from Bluetooth over WLANs to 4G/5G broadband communication, the question of simulating synchronous executions in such systems arises. Unfortunately, though, the communication properties of a wireless link are typically unstable and highly time-variant (Cerpa et al. 2005b), due to limited transmission ranges, near-far problems (Ware et al. 2000), fading (Schilcher et al. 2012), interference (Fussen et al. 2005) and other phenomenons. There is hence no alternative to statistical modeling lossy links, which has been done in various different models, for simple sensor networks (Cerpa et al. 2005a) to elaborate *signal-to-interference-plus-noise* (SINR) ratio (Dousse et al. 2005) and even fading models (Bettstetter and Hartmann 2005; Schilcher et al. 2016). Most of this work focuses on individual links; some papers also deal with broadcasting protocols (Clementi et al. 2007).

We use a very simple model based on these results, which just assumes independent and identically distributed message loss per communication link. A similar assumption also underlies the edge-Markovian model (Clementi et al. 2008). It is appealing because of tractability and, despite its simplicity, not unreasonable in practice, at it provides (probabilistic) lower bounds on the performance of real networks for suitably chosen message loss probabilities. We note that, depending on the type of the underlying wireless network, both a constant value of p and a value that decreases with the number N of participants may make sense here: In wireless networks where a collision, i.e., a simultaneous attempt of two senders to broadcast a message at the same time, may lead to the destruction of both messages, some form of transmission scheduling needs to applied. Maximizing the overall throughput or similar performance measures in such networks (Gupta and Kumar 2000) requires to reduce the sending probability, and hence also p, down to something like $1/\log N$ or even $1/N$, see e.g. Moscibroda and Wattenhofer (2006) for more information.

In Függer et al. (2015), Függer et. al. analyzed the expected round duration of the α-synchronizer in a synchronous distributed system of N processes that execute in lock-step unit-time rounds in such a model. The model just assumes that every message sent from process i to process j in a round may be lost with some fixed probability $1-p$. The expected round duration is crucial for determining the running time of a synchronous distributed algorithm running atop of the α-synchronizer: its expected value is just the time complexity of the algorithm (measured in rounds, as already mentioned) times the expected round duration.

It turned out that the operation of the α-synchronizer, and variants thereof that sometimes forget part of their state, in such a system can be modeled by an infinite-state Markov chain. Whereas (Függer et al. 2015) also provided a reduction to a finite-state Markov chain, it has a state space that is exponential in the number of processes in the system, however. Owing to the inherent complexity involved in the numerical or analytical solution of this chain, the authors had to resort to coarse lower and upper bounds for analyzing the synchronizer performance, in particular, the expected duration of a synchronized round.

Main results:

(1) We provide a synchronizer α' and show that it provides an upper bound on the performance of the α-synchronizer.

(2) We prove analytically that the upper bound guaranteed by α' is not worse than the exponential-size upper bound presented in Függer et al. (2015), and strictly better for $p \to 0$. These results are backed-up by simulation results, which show that the accuracy of the approximation of the expected round duration of α provided by our new upper bound is considerably better than the original one.

(3) We model α' by a finite-state Markov chain, which has only polynomial state space.

(4) We provide a variant of our α'-synchronizer, and prove that it provides a lower bound for the performance of α. Albeit its complexity did not allow us to find an analytical proof, we demonstrate by means of elaborate simulations that our new upper bound better approximates the expected round duration of α than the existing lower bound from Függer et al. (2015).

These results also contribute to a better understanding of the Markov chain underlying the original problem, which may eventually pave the way to computationally more efficiently computable bounds.

Related work Early work on synchronizer performance in probabilistic systems considered varying message delays and computation times: Bertsekas and Tsitsiklis (1989) proved performance bounds for the case of constant processing times and exponentially distributed message delays on communication links without message loss.

This model has been augmented by exponentially distributed processing times in Rajsbaum (1994a). On the other hand, Rajsbaum and Sidi (1994b) analyzed synchronizer performance in the case of exponentially distributed processing times and negligible transmission delays.

In contrast to the above work, we assume bounded message delays. Varying delays between sending and successfully receiving a message are due to message loss and repeated retransmission. The performance of the α-synchronizer in certain lossy environments has been considered by Nowak et al. (2013). The authors calculated the expected round duration of a retransmission-based synchronizer in systems, where every message is successfully transmitted with constant probability p, subject to the additional constraint that a message that was retransmitted at least M times is guaranteed to arrive. Nowak et al. (2013) assumed M to be finite, however, which Függer et al. (2015) (and we) do not.

The dominant computational complexity in solving Markov chains like the ones arising in Függer et al. (2015) is due to calculating the steady states. Instead of exactly determining those, there exist also techniques that allow to just sample the steady state: However, while standard simulation techniques allow to sample the Markov chain's state at some time $t = T$, there is no guarantee that these samples resemble the distribution of the steady state for $t \to \infty$. By contrast, Propp and Wilson (1996) proposed backward coupling techniques to obtain exact steady state samples for Markov chains. In the case of monotonic Markov chains, these techniques are computationally efficient. Unfortunately, while our infinite state Markov chains are monotonic, our reduced finite chains are not. Their method thus requires to explore the complete finite state space, rendering this method computationally infeasible.

Paper organization Section 2 introduces our system model and the performance measure of interest, as well as the α-synchronizer and its corresponding Markov chain. In Section 3, we introduce our novel upper bound α'-synchronizer and its Markov chain; Section 3.3 shows that it indeed provides an upper bound for the α-synchronizer, Section 3.4 evaluates the asymptotics for $p \to 0$, and in Section 3.5 we compare this bound with the existing upper bound. Section 4 finally provides a modification of the α'-synchronizer and the proof that it indeed provides a lower bound for the performance of the α-synchronizer. The paper is rounded off in Section 5 by our simulation results, a discussion of our findings, and some future work; a glossary of our notation is appended in Section 6.

2 Model of the Synchronizer

2.1 System Model and Algorithm

In this paper, we study the performance of the α-synchronizer (Awerbuch 1985) running in a fully-connected message passing system with processes $1, 2, \ldots, N$. Processes take steps simultaneously at all integral times $t \geq 0$, but messages may be lost with some fixed probability $1 - p$. Messages that do arrive have a transmission delay of 1, i.e., a message sent at time t arrives at time $t + 1$, or not at all. A step consists in (a) receiving messages from other processes, (b) performing local computations, and (c) broadcasting a message to the other processes.

The synchronizer has two local variables, specified for every process i at time t: The *local round number* $R_i(t)$ and the *knowledge vector* $\big(K_{i,1}(t),\ K_{i,2}(t), \ldots, K_{i,N}(t)\big)$. Processes broadcast their local round number $R_i(t)$ in every step t. The knowledge vector contains information on other processes' local round numbers, accumulated via received messages. A process increments its local round number, and thereby starts the next round in step $(t+1)$, after it has gained knowledge that all other processes have already started the current round by step t. This round increment rule hence assures a precision of 1, i.e., $|R_i(t) - R_j(t)| \leq 1$ for all t. We write $R(t) = \min_i R_i(t)$ and call it the *global round number* at time t. When $R(t)$ increases, we say a global round switch occurs.

Formally, let $(\mathcal{P}(t))_{t\in\mathbb{N}^*}$ be a sequence of $(N \times N)$-matrices whose entries are pairwise independent random variables with

$$\mathbb{P}\big(\mathcal{P}_{i,j}(t) = 1\big) = p \text{ if } i \neq j \text{ and } \mathbb{P}\big(\mathcal{P}_{i,i}(t) = 1\big) = 1\ , \tag{1}$$

where $\mathcal{P}_{i,j}(t+1) = 0$ means that process j's message to process i sent at time t via channel (i, j) was lost, and $\mathcal{P}_{i,j}(t+1) = 1$ that it arrives (at time $t + 1$). Therefore we call the parameter p the *probability of successful transmission*. Note that in our notation of a channel, process j is the sender and process i is the receiver, i.e., the channel (i, j) leads from j to i. Moreover, row i in $\mathcal{P}(t)$ corresponds to the point of view of the *receiver* i.

Initially, $R_i(0) = 0$ and $K_{i,j}(0) = -1$ (i.e., no messages are received at time 0).

At every time step $t \geq 1$, process i's computation consists of the following:

1. *Update knowledge according to received messages:* $K_{i,j}(t) \leftarrow R_j(t-1)$ if $\mathcal{P}_{i,j}(t) = 1$, and $K_{i,j}(t) \leftarrow K_{i,j}(t-1)$ otherwise.
2. *Increment round number if possible:* $R_i(t) \leftarrow R_i(t-1) + 1$ if $K_{i,j}(t) \geq R_i(t-1)$ for all j, and $R_i(t) \leftarrow R_i(t-1)$ otherwise.

In the remainder of this paper, when we refer to $K_{i,j}(t)$ and $R_i(t)$, we mean its value after step (2).

Figure 1 shows part of an execution of the α-synchronizer. Times are labeled t_0 to t_{10}. Processes 1 and 3 start their local round r at time t_4 while process 2 has already started its local round r at time t_3. The arrows in the figure indicate the time until the first successful reception of a message sent in round r: The tail of the arrow is located at time t a process i

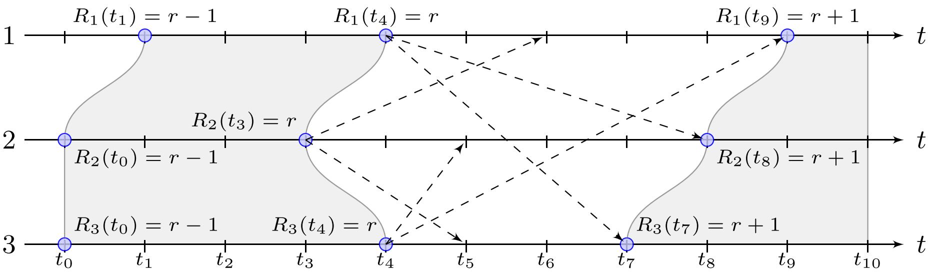


Fig. 1 An execution of the synchronizer

starts round r and thus broadcasts r for the first time. The head of the arrow marks the smallest time after t at which a process j receives a message from i. Messages from processes to themselves are always received at the next time step and thus are not explicitly shown in the figure. For example, processes 1 and 3 start round r at time t_4 sending r for the first time. While process 2 receives the message from 3 in the next step, it needs an overall amount of 4 time steps and consecutive retransmissions to receive a message from process 1 at time t_8.

2.2 Performance Measure

For a system with N processes and probability p of successful transmission, we define the *expected round duration* of process i by $\lambda_i(N, p) = \mathbb{E} \lim_{t\to\infty} t/R_i(t)$. Since our synchronization algorithm guarantees precision 1, it directly follows that $\lambda_i(N, p) = \lambda_j(N, p)$ for any two processes i and j. We will henceforth refer to this common value as $\lambda(N, p)$, or simply λ if the choice of parameters N and p is clear from the context.

2.3 Markov Chain and Definitions

The α-synchronizer can easily be modeled as a Markov chain (see Függer et al. (2015) for details): Let $A(t)$ be the sequence of matrices with $A_{i,i}(t) = R_i(t)$ and $A_{i,j}(t) = K_{i,j}(t)$ for $i \neq j$. It is easy to see that $A(t)$ is a Markov chain, i.e., the distribution of $A(t+1)$ depends only on $A(t)$. Since both $R_i(t)$ and $K_{i,j}(t)$ are unbounded, the state space of Markov chain $A(t)$ is infinite.

We therefore introduce the sequence of *normalized* states $a(t)$, defined by $a(t) = A(t) - \min_k A_{k,k}(t)$.

Normalized states belong to the finite set $\{-1, 0, 1\}^{N\times N}$. This is still a Markov chain.

Clearly, the computation steps defined above can be translated directly in terms of $A(t)$ and $a(t)$: For $A(t)$ they read

1. *Update knowledge according to received messages:* For $j \neq i$: $A_{i,j}(t) \leftarrow A_{j,j}(t-1)$ if $\mathcal{P}_{i,j}(t) = 1$, and $A_{i,j}(t) \leftarrow A_{i,j}(t-1)$ otherwise.
2. *Increment round number if possible:* $A_{i,i}(t) \leftarrow A_{i,i}(t-1) + 1$ if $A_{i,j}(t) \geq A_{i,i}(t-1)$ for all $j \neq i$, and $A_{i,i}(t) \leftarrow A_{i,i}(t-1)$ otherwise.

For $a(t)$ we have the following:

1. *Update knowledge according to received messages:* For $j \neq i$: $a_{i,j}(t) \leftarrow a_{j,j}(t-1)$ if $\mathcal{P}_{i,j}(t) = 1$, and $a_{i,j}(t) \leftarrow a_{i,j}(t-1)$ otherwise.
2. *Increment round number if possible:* $a_{i,i}(t) \leftarrow a_{i,i}(t-1) + 1$ if $a_{i,j}(t) \geq a_{i,i}(t-1)$ for all $j \neq i$, and $a_{i,i}(t) \leftarrow a_{i,i}(t-1)$ otherwise.
3. *Normalizing:* $a_{i,j}(t) \leftarrow a_{i,j}(t) - 1$ $\forall i, j$ if $\min_{1\leq k\leq N} a_{k,k}(t) = 1$.

In the following, we will switch between these representations as needed. For simplicity, when we extend the computation steps later on, we will do it for only one of these representations since the adaptations are straightforward.

As before, $A(t)$ resp. $a(t)$ refer to the values at the end of time step t (here: after step (2) resp. (3)). If we need to specifically refer to the values after step (i) ($i \in \{1, 2, 3\}$), we denote this matrices by $A^{(i)}(t)$ and $a^{(i)}(t)$. Moreover, let $md^{(i)}(t)$ denote the minimum of the diagonal of $A^{(i)}(t)$ and $w^{(i)}(t)$ the number of $md^{(i)}$-entries in the diagonal of $A^{(i)}(t)$.

A channel is called *relevant* (in $a(t)$ resp. $A(t)$) if a successfully transmitted message on this channel in $\mathcal{P}(t+1)$ would increase knowledge (in $a(t+1)$ resp. $A(t+1)$). Similarly, a message is called *relevant* if its arrival would increase knowledge.

We call a processes i an *r-process* in state $a(t)$ (or $A(t)$) if $a_{i,i}(t) = r$ (or $A_{i,i}(t) = r$).

Example 1 To illustrate the two Markov chains defined above, we give the following example:

$$t\colon \quad 0 \qquad 1 \qquad 2 \qquad 3 \qquad 4$$

$$A(t)\colon \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix} \to \cdots$$

$$\mathcal{P}(t)\colon \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \to \cdots$$

$$a(t)\colon \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} \to \cdots$$

At time $t = 0$ all channels are relevant, at time $t = 1$ the relevant channels (messages) are $(1, 2)$, $(1, 3)$, $(2, 3)$ and $(2, 1)$, whereas at $t = 3$ we have $(1, 2)$, $(2, 1)$, $(3, 1)$, and $(3, 2)$ as relevant messages.

Global round switches occur at times $t = 2$ and $t = 4$.

In $a(t)$, at times $t \in \{0, 2, 4\}$ we have three 0-processes, at $t = 1$ we have two 0-processes and one 1-process, and at time $t = 3$ we have one 0-process and two 1-processes.

Since it is very expensive to calculate the expected round duration based on this Markov chain, Függer et al. (2015) presented easy computable but quite conservative upper and lower bounds. The main purpose of this paper is to develop a new upper bound approximation, which will be shown both analytically and by means of simulations to improve the known upper bound. It will also be stated as a Markov chain and therefore still expensive to calculate, but reduces the state space from exponential to polynomial size in N.

3 New Upper Bound

3.1 Algorithm of our Upper Bound

We will now present the algorithm that generates our new upper bound, which will be called the α'-synchronizer. The main idea is to insert a reordering step between steps (1) and (2) of the computation, which reorders the entries in the matrix $a(t)$ in such a way that generating

1-entries in the diagonal is avoided as long as possible. Roughly speaking, we take all non-diagonal entries of the matrix, sort them in descending order of their value, and fill them in column by column.

Formally, we introduce a step $(1a)$ after updating the knowledge:

(1a) *Reordering of the knowledge*: Let $(k_i)_{i=1}^{n(n-1)}$ be a decreasing sorting of the multiset $\{A_{i,j}^{(1)}(t) \ : \ i \neq j\}$. Now fill in this sequence into the matrix column by column (omitting the diagonal) until position $(N-2, N-1)$ has been filled. Then go to $(1, N)$ and fill the last column. Finally, fill position $(N, N-1)$.

However, in order to be able to prove that this indeed results in an upper bound, we slightly need to modify the above simple strategy of updating the knowledge to slow down the synchronizer even further: If we are in a state with exactly one r-process i [and $(N-1)$ $(r+1)$-processes], then some of i's $(r-1)$-knowledge will be updated to an r-knowledge, rather than to the $(r+1)$-knowledge sent [by one of the other $(r+1)$ processes] if i switches to an $(r+1)$-process in step (2). This can be implemented into our algorithm by introducing an additional step (3):

(3) If $w^{(2)}(t) - w^{(2)}(t-1) = N-1$, then replace the last non-diagonal $md^{(2)}(t)$-entry (according to our filling rule) by $md^{(2)}(t) - 1$.

Step (1a) means one fills in the 1s first, then the 0s, and finally the (-1)s. For example if we have a (4×4)-matrix then the matrix is filled up in the following order as stated in the matrix S below. On the right-hand side we give an example of reordering a matrix resulting from computation step (1) and incrementing round numbers afterwards. Note that due to the reordering 1-processes will be on the top of the matrix and 0-processes on the bottom.

$$S = \begin{array}{cccc} & 4 & 7 & 9 \\ 1 & & 8 & 10 \\ 2 & 5 & & 11 \\ 3 & 6 & 12 & \end{array} \qquad \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \overset{(1a)}{\to} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix} \overset{(2)}{\to} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Note that in this example reordering indeed has an effect: Without reordering processes 3 and 4 would become 1-processes whereas now only process 1 is a 1-process.

Example 2 This example illustrates the α'-synchronizer and compares it to the α-synchronizer:

$$\begin{array}{lccccccccc}
t: & 0 & & 1 & & 2 & & 3 & & 4 \\
a(t): & \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} \to \cdots \\
\mathcal{P}(t): & & \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \to \cdots \\
a'(t): & \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \to \cdots
\end{array}$$

Here the α-synchronizer does two global round steps (at times $t = 2$ and $t = 4$), but the α'-synchronizer only does one (at $t = 4$). So the α'-synchronizer is indeed slower than the original α-synchronizer. We will see later on, however, that—when provided with the same sequence of matrices $\mathcal{P}(t)$—this slowdown is not always the case. Note that at time $t = 4$ we applied step (3). Otherwise the rightmost entry in the first row would be 0.

Observe that several different sequences of matrices $\mathcal{P}(t)$ can lead to the same $a'(t)$ (but different $a(t)$):

$$\begin{array}{rccccccccccc}
t\colon & 0 & & 1 & & 2 & & 3 & & 4 & \\
a(t)\colon & \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 0 & -1 & -1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \to \cdots \\
\mathcal{P}(t)\colon & & \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \to \cdots & \\
a'(t)\colon & \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \to \cdots
\end{array}$$

Note that, without the additional modification (step (3)) of the primary reordering of knowledge step (1a) needed to make our proofs working, one can achieve $a(t) = a'(t)$ if the matrices $\mathcal{P}(t)$ are properly chosen, as the following example shows:

$$\begin{array}{rccccccccccc}
t\colon & 0 & & 1 & & 2 & & 3 & & 4 & \\
a(t) = a'(t)\colon & \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} & \to \cdots \\
\mathcal{P}(t)\colon & & \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} & \to & \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \to \cdots &
\end{array}$$

Example 3

In this case, another view of the basic upper bound construction is not to employ the reordering step $(1a)$, but to choose specific matrices $\mathcal{P}(t+1)$ depending on the state $a(t)$. Unfortunately, though, this does not work if the additional modification is employed.

The proof that this algorithm leads indeed to an upper bound is given in Section 3.3. However, first we give a detailed description of the corresponding Markov chain.

3.2 Markov Chain of our Upper Bound

We now describe the Markov chain that models our upper bound. For this purpose, note that, due to our reordering step $(1a)$, it is not necessary anymore to store the whole matrix: It would be sufficient to know the number of 1s and 0s we have to fill in. However, we will use another representation, which is even more appropriate for our purposes: We represent a matrix by a pair (x, a), where x denotes the number of 0-entries in the diagonal (i.e., 0-processes) and a is the number y of non-diagonal 0-entries if $x = N$, whereas a is the number z of non-diagonal 1-entries if $1 \leq x \leq N - 1$.

Note that these pairs contain all the information we need: Firstly, there can be only 0- or 1-entries in the diagonal — so it is sufficient to know the number of one of them. Secondly, if $x = N$, then the only non-diagonal elements are (-1) or 0 – so again one of its numbers is sufficient. Thirdly, if $1 \leq x \leq N - 1$ we can have 1s, 0, and (-1)s outside the diagonal. But, due to our filling rule, x 0-processes imply that we have exactly x (-1)-entries! Hence knowing the number either of 1s or 0s is sufficient again.

For simplicity, we introduce the following abbreviations:

$$\Delta(x) = x_1 - x_2, \qquad \nu = N(N-1), \qquad \text{and} \qquad \nu(y) = (N-1)(N-y).$$

Our Markov chain can now be described as follows: The state space is given by pairs (x, a), where

$$(x,a) = \begin{cases} (N, y) & 0 \leq y \leq N(N-2) \\ (x, z) & 0 \leq z \leq (N-1)(N-x) - \delta_{1x} \text{ for } 1 \leq x \leq N-1, \end{cases} \tag{2}$$

where δ_{ij} is the Kronecker delta, i.e., $\delta ij = 1$ if $i = j$ and 0 otherwise. Clearly, we can have $1 \leq x \leq N$ 0-processes. For y, observe that with N 0-processes there can be up to $N(N-2)$ non-diagonal 0-entries (since there must be one remaining (-1) in each row). For the number of possible 1s we have to be a little bit more careful: If $x \geq 2$, then for each 1-process there can be up to $(N-1)$ 1-entries in the corresponding columns. On the other hand, if $x = 1$, we have to subtract one because the last process still needs a (-1)-entry. So the number of states of the Markov chain equals $N(N^2 - 1)/2$.

Let $s_1 = (x_1, y_1, z_1)$ and $s_2 = (x_2, y_2, z_2)$ be the uniform representation of the states s_1 and s_2, where, depending on e.g. x_1, either y_1 or z_1 is not used. Then, the transition probabilities $p_{s_1 s_2}$ from state s_1 to s_2 are given by $p_{s_1 s_2} = \tilde{p}_{s_1 s_2} + \hat{p}_{s_1 s_2}$, where $\tilde{p}_{s_1 s_2}$ is the probability of a transition without making a global round switch and $\hat{p}_{s_1 s_2}$ is the probability of a transition with a global round switch, with

$$\tilde{p}_{s_1 s_2} = \begin{cases} \dbinom{\nu - y_1}{-\Delta(y)} p^{-\Delta(y)} (1-p)^{\nu - y_2} & \\ \qquad \text{if } x_1 = x_2 = N,\ y_1 \leq y_2 \leq N(N-2) & \text{(3a)} \\ \dbinom{\nu(x_1) - \delta_{1x_1} - z_1}{-\Delta(z)} p^{-\Delta(z)} (1-p)^{\nu(x_1) - \delta_{1x_1} - z_2 + x_1} & \\ \qquad \text{if } x_1 = x_2 < N,\ z_1 \leq z_2 & \text{(3b)} \\ \dbinom{\nu - y_1}{x_2} p^{\nu - y_1 - x_2} (1-p)^{x_2} & \\ \qquad \text{if } N = x_1 > x_2,\ z_2 = 0 & \text{(3c)} \\ \dbinom{\nu(x_1) - z_1}{-\Delta(z)} \dbinom{x_1}{\Delta(x)} p^{\Delta(x-z)} (1-p)^{\nu(x_1) - z_2 + x_2} & \\ \qquad \text{if } N > x_1 > x_2, z_1 \leq z_2,\ z_2 \leq (N-1)(N-x_1) & \text{(3d)} \\ 0 \qquad \text{ow.} & \text{(3e)} \end{cases}$$

In case (3a), clearly the number of non-diagonal 0-entries can only increase and is bounded by $N(N-2)$ (as mentioned above). Hence, to do the state transition, exactly $(y_2 - y_1)$ from the $(\nu - y_1)$ relevant 0-messages must arrive in order to replace $(y_2 - y_1)$ of the (-1)-entries in s_1.

If $x_1 = x_2 < N$ (case (3b)), the number of 1-entries cannot decrease. Since the number of 0-processes remains the same, none of the x (-1)-entries is allowed to be overwritten by a 0 or 1. This gives the factor $(1-p)^{x_1}$. Moreover, the number of 1s increases by $(z_2 - z_1)$, hence, for $x_1 > 1$, we can choose them among the $(\nu(x_1) - z_1)$ relevant channels. If $x_1 = 1$, we have to subtract the one (-1)-entry of the last process (otherwise we would do a global round switch).

In case (3c), where we have a transition from a state with N 0-processes to a state with $x_2 < N$ 0-processes, we must have $z_2 = 0$ since in s_1 there are only 0-processes and so no 1-messages can be sent. Due to the fact that having x_2 0-processes is equivalent to have x_2 (-1)-entries, all but x_2 of the $(\nu - y_1)$ (-1)-entries must be replaced by 0.

In case (3d), we do a transition with decreasing number of 0-processes again. To ensure that exactly $(x_2 - x_1)$ 0-processes become 1-processes, exactly $(x_2 - x_1)$ of the x_1 (-1)-entries in s_1 must be replaced with 0s. This leads to the term $\binom{x_1}{\Delta(x)} p^{\Delta(x)}(1-p)^{x_2}$. Clearly, the number of 1-entries cannot decrease. So we have to choose exactly $(z_2 - z_1)$ messages from the $(\nu(x_1) - z_1)$ relevant channels outgoing from 1-processes for a successful transmission.

The second part of the transition probabilities is given by

$$\hat{p}_{s_1 s_2} = \begin{cases} p^{\nu - y_1} & \\ \qquad \text{if } x_1 = x_2 = N, \ y_2 = z_2 = 0 & \text{(4a)} \\ \binom{\nu(x_1) - \delta_{1x_1} - z_1}{y_2 - z_1} p^{y_2 - z_1 + x_1}(1-p)^{\nu(x_1) - y_2 - \delta_{1x_1}} & \\ \qquad \text{if } x_1 < x_2 = N, z_1 \leq y_2, \ y_2 \leq (N-1)(N-x_1) - \delta_{1x_1} & \text{(4b)} \\ 0 \qquad\qquad \text{ow.} & \text{(4c)} \end{cases}$$

Clearly, if a global round switch occured we have only 0-processes in the matrix. Hence, a positive transition probability is only possible if $x_2 = N$ and $z_2 = 0$. In case (4a), we also have N 0-processes in state s_1. Thus, only 0-messages can be sent and so we have only (-1)s after the round switch in the non-diagonal entries of the matrix. This is why y_2 has to be 0. To do the round switch, all of the remaining $(\nu - y_1)$ (-1)-entries in s_1 have to be overwritten by 0. This immediately gives the transition probability in this case. [(1a)] Let us turn to case (4b): We start with $x_1 > 1$: If we have $(N - x_1)$ 1-processes and z_1 non-diagonal 1-entries in s_1, then the number y_2 of non-diagonal 0-entries in s_2 is at least z_1 (since the existing 1s convert to 0s in a global round switch) and at most $\nu(x_1) = (N-1)(N-x_1)$ (the maximum number of 1s generated by $(N - x_1)$ 1-processes). To make this state-transition, we have to take two things into account: Firstly, all x_1 (-1)-entries in s_1 must be overwritten – this gives the term p^{x_1}. Secondly, we have to produce exactly $(y_2 - z_1)$ new 1-entries (before doing the reduction due to the global round switch). These 1s overwrite existing 0s. Thus, we can choose them from the $(\nu(x_1) - z_1)$ relevant channels outgoing from 1-processes. Finally, note that in case $x_1 = 1$ the last (-1)-entry is overwritten by 0 although the correspoding messages was sent from a 1-process! This is why we have to add the correction term $-\delta_{1x_1}$.

3.3 Proof Upper Bound

Now we want to show that our previously defined process is indeed an upper bound. First of all, it is worthwile to mention that this can't be done execution-wise, since there exist schedules such that the α'-synchronizer is faster than the α-synchronizer.

Example 4 We give an example of a schedule such that the α'-synchronizer is faster than the α-synchronizer. Here $A'(t)$ and $A(t)$ denote the matrices as defined in Section 2.3.

$$A'(t)\text{:}\quad \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & -1 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \to \cdots$$

$$\mathcal{P}(t)\text{:}\quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \to \cdots$$

$$A(t)\text{:}\quad \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \to \cdots$$

To handle this problem we will construct a measure-preserving bijection f on the sets of schedules (i.e., a bijection that preserves the number of ones in the message patterns at each time step) such that the α-synchronizer under a schedule E is always faster than the α'-synchronizer given the schedule $f(E)$. The basic idea behind our construction is to map the relevant messages of the α-synchronizer on the relevant messages of the α'-synchronizer in such a way that the α-synchronizer is always in front.

Let

$$\mathcal{M} := \text{set of messagepatterns} \cong \{0,1\}^{N^2-N}$$

and

$$\mathcal{E} = \mathcal{M}^{\mathbb{N}} = \text{ set of schedules},$$

and

$$\mathcal{E}_n = \mathcal{M}^n$$

the set of sequences of messagepatters of length n (i.e., prefixes of schedules). For $E \in \mathcal{E}$ or $\mathcal{E}_m$ let E_n denote the nth element of E. Moreover, let $|E_n| = \sum_{i,j} E_{n,i,j}$ denote the number of ones in E_n. With $E_{\le n}$ we denote the prefix of length n of E, i.e., the tuple $(E_1, \dots, E_n)$.

Then we have the following theorem:

Theorem 1 *There exists a bijection $f : \mathcal{E} \to \mathcal{E}$ with*

- $|E_n| = |f(E)_n| \quad \forall\, E \in \mathcal{E},\ \forall n$ *and*
- $R(E,n) \ge R'(f(E),n) \quad \forall\, E \in \mathcal{E},\ \forall n.$

Here, $R(E,n)$ denotes the global round number of the α-synchronizer at time n given the schedule E and $R'(E',n)$ the global round number of the α'-synchronizer at time n given the schedule E'.

Proof Let $A(n) = A(E,n)$ and $A'(n) = A'(f(E),n)$ denote the matrices of the α- and the α'-synchronizer under the schedule E and $f(E)$, respectively. Moreover, define $m_n = \min_{i,j} A(n)_{i,j}$ $(i \ne j)$ as the minimum of $A(n)$, and $a_n(m_n)$ as the number of non-diagonal entries of $A(n)$ that equal the minimum m_n. With $b'_n(m_n)$ we denote the number of non-diagonal entries in $A'(n)$ less than or equal to m_n. Similarly, $R(n) = R(E,n)$ and $R'(n) = R'(f(E),n)$ are the global round numbers of $A(n)$ and $A'(n)$, respectively.

To simplify notation, we define for a matrix $A(n)$ the submatrix $A(n, u, v)$ as the submatrix consisting of the intersection of rows of u-processes and columns of v-processes. To illustrate this definition look at the following example.

Example 5 Let

$$A(n) = \begin{pmatrix} 5 & 5 & 4 & 4 & 4 & 4 \\ 4 & 5 & 4 & 4 & 4 & 4 \\ 5 & 4 & 5 & 4 & 4 & 4 \\ 4 & 5 & 4 & 4 & 3 & 3 \\ 4 & 4 & 3 & 3 & 4 & 4 \\ 5 & 3 & 4 & 4 & 3 & 4 \end{pmatrix}.$$

Then

$$A(n,5,5) = \begin{pmatrix} 5 & 5 & 4 \\ 4 & 5 & 4 \\ 5 & 4 & 5 \end{pmatrix}, \quad A(n,4,4) = \begin{pmatrix} 4 & 3 & 3 \\ 3 & 4 & 4 \\ 4 & 3 & 4 \end{pmatrix},$$
$$A(n,5,4) = \begin{pmatrix} 4 & 4 & 4 \\ 4 & 4 & 4 \\ 4 & 4 & 4 \end{pmatrix}, \quad A(n,4,5) = \begin{pmatrix} 4 & 5 & 4 \\ 4 & 4 & 3 \\ 5 & 3 & 4 \end{pmatrix}.$$

Now we will inductively construct functions $f_n : \mathcal{E}_n \to \mathcal{E}_n$ ($n \geq 1$), the limit of which gives f, with the following properties:

(i) f_n bijective on $\mathcal{E}_n$,
(ii) $|E_n| = |(f_n(E))_n| \quad \forall\, E \in \mathcal{E}_n$,
(iii) $R(E, n) \geq R'(f_n(E), n) \quad \forall\, E \in \mathcal{E}_n$,
(iv) $a_n(m_n) \leq b'_n(m_n)$ and
(v) if $R(n) = R'(n)$: $|\mathcal{A}_n| + |\mathcal{G}_n| \geq |\mathcal{A}'_n|$, where (for matrices $A(n)$ and $A'(n)$ with $r = R(n) = R'(n)$)

$$\begin{aligned} \mathcal{A}_n, \mathcal{A}'_n &= \{\text{positions of non-diagonal } (r+1)\text{-entries in } A(n) \text{ resp. } A'(n)\}, \\ \mathcal{G}_n, \mathcal{G}'_n &= \{\text{positions of } (r-1)\text{-entries in } A(n, r, r+1) \text{ resp. } A'(n, r, r+1)\}. \end{aligned}$$

(vi) The function f_n is an extension of f_{n-1}, i.e., if $f_n((x_1, \ldots, x_{n-1}, x_n)) = (y_1, \ldots, y_{n-1}, y_n)$ then $f_{n-1}((x_1, \ldots, x_{n-1})) = (y_1, \ldots, y_{n-1})$ for $n \geq 2$.

Then the function f defined by $f(E) = \lim_{n\to\infty} f_n(E_{\leq n})$ has the stated properties.

Remark 1 It is worthwile to mention that conditions (iv) and (v) imply $\sum_{i\neq j} A(E, n)_{i,j} + |\mathcal{G}_n| \geq \sum_{i\neq j} A'(f_n(E), n)_{i,j}$.

We start with $f_1 := \mathrm{id}_{\mathcal{M}}$. Then f_1 fulfils obviously $(i) - (v)$. Let f_n with $(i) - (vi)$ be already defined. We will first construct – in dependence on $E_{\leq n}$ (and consequently on $A(n)$) – the function $f_{E_{\leq n}}\colon \mathcal{M} \to \mathcal{M}$. Then, the function $f_{n+1} : \mathcal{E}_{n+1} \to \mathcal{E}_{n+1}$ is defined as follows:

$$f_{n+1}(E) = f_{n+1}(E_{\leq n}, E_{n+1}) = (f_n(E_{\leq n}), f_{E_{\leq n}}(E_{n+1})). \tag{5}$$

For our construction, we have to do a case distinction:

Case A $R(n) = R'(n) = r$**:**

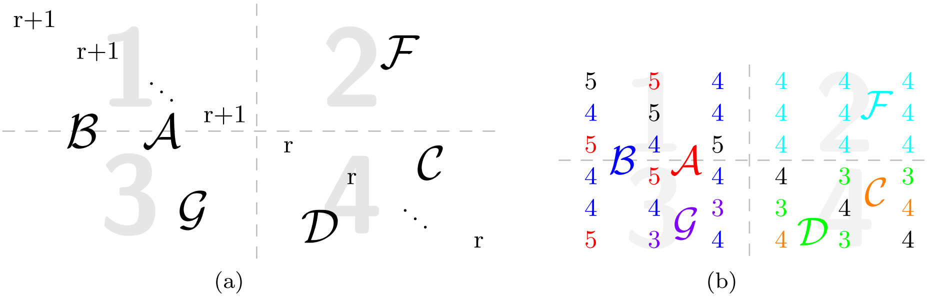


Fig. 2 The partioning of matrix A into the sets $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{F}, \mathcal{G}$

Given the matrices $A(n)$ and $A'(n)$, we also define the following sets:

$$\begin{aligned}
\mathcal{B}_n, \mathcal{B}'_n &= \{\text{positions of } r\text{-entries in } A(n, r+1, r+1) \text{ and } A(n, r, r+1) \\
&\qquad\qquad \text{resp. } A'(n, r+1, r+1) \text{ and } A'(n, r, r+1)\}, \\
\mathcal{F}_n, \mathcal{F}'_n &= \{\text{positions of } r\text{-entries in } A(n, r+1, r) \text{ resp. } A'(n, r+1, r)\}, \\
\mathcal{C}_n, \mathcal{C}'_n &= \{\text{positions of non-diagonal } r\text{-entries in } A(n, r, r) \text{ resp. } A'(n, r, r)\}, \\
\mathcal{D}_n, \mathcal{D}'_n &= \{\text{positions of } (r-1)\text{-entries in } A(n, r, r) \text{ resp. } A'(n, r, r)\}.
\end{aligned}$$

For a graphical interpretation of these sets see Fig. 2a: The sets $\mathcal{A}$ and $\mathcal{B}$ contain (the positions of) the $(r+1)$ resp. r-entries in regions 1 and 3, region 2 is exactly $\mathcal{F}$, the $(r-1)$-entries in region 3 correspond to $\mathcal{G}$, and the sets $\mathcal{C}$ and $\mathcal{D}$ contain the r resp. $(r-1)$-entries in region 4.

Example 6 Continuing Example 5 we have $r = 4$ and

$$\begin{aligned}
\mathcal{A}_n &= \{(1,2), (3,1), (4,2), (6,1)\}, \\
\mathcal{B}_n &= \{(1,3), (2,1), (2,3), (3,2), (4,1), (4,3), (5,1), (5,2), (6,3)\}, \\
\mathcal{F}_n &= \{(1,4), (1,5), (1,6), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6)\}, \\
\mathcal{C}_n &= \{(5,6), (6,4)\}, \\
\mathcal{D}_n &= \{(4,5), (4,6), (5,4), (6,5)\}, \\
\mathcal{G}_n &= \{(5,3), (6,2)\}.
\end{aligned}$$

A visualisation of these sets is given in Fig. 2b.

For simplicity, we will drop the index n in the following. Note that

$$|\mathcal{A}| + |\mathcal{B}| + |\mathcal{F}| + |\mathcal{C}| + |\mathcal{D}| + |\mathcal{G}| = N(N-1). \tag{6}$$

Moreover, $\mathcal{B}$, $\mathcal{D}$, and $\mathcal{G}$ correspond to relevant channels $M = \mathcal{B} \cup \mathcal{D} \cup \mathcal{G}$ in $A(n)$ (and analogous for $A'(n)$).

Our assumptions imply $|\mathcal{A}| + |\mathcal{G}| \geq |\mathcal{A}'|$ (due to (v)) and $|\mathcal{D}| + |\mathcal{G}| \leq |\mathcal{D}'| + |\mathcal{G}'|$ (see (iv): we have $m_n = r - 1$). The next property is a little bit more involved: Note that in Case A we have $m_n = r - 1$. Moreover, due to (iv) we have at least as many non-diagonal $(r-1)$-elements in $A'(n)$ than in $A(n)$; recall that because of computation step (1a) (where messages are inserted column by column) we generate with b'_n $(r-1)$-elements as many r-processes in $A'(n)$ as possible. Hence the number of r-processes in A_n (i.e., the number of diagonal elements equal r) is less than or equal to the number of r-processes in A'_n;

equivalently $|\mathcal{C}| + |\mathcal{D}| \leq |\mathcal{C}'| + |\mathcal{D}'|$ (the sizes of the submatrices $A(n, r, r)$ and $A'(n, r, r)$) or equivalently

$$|\mathcal{A}| + |\mathcal{B}| + |\mathcal{G}| \geq |\mathcal{A}'| + |\mathcal{B}'| + |\mathcal{G}'| \tag{7}$$

(the total number of entries in the columns with $(r + 1)$ in the diagonal).

Note that $\mathcal{G}'$ is either empty or contains just one element, namely $(N, N - 1)$. This is the special case in which we interpret the 1-message transmitted on this channel only as 0-message (i.e., applying step (3)).

Let h be any bijection from $[N]^2$ to $[N]^2$ keeping the diagonal fixed with

a) $h(\mathcal{D} \cup \mathcal{G}) \subseteq \mathcal{D}' \cup \mathcal{G}'$ and
b)

$$\begin{cases} \mathcal{B}' \subseteq h(\mathcal{B}) & \text{if } |\mathcal{B}'| \leq |\mathcal{B}| \\ h(\mathcal{B}) \subseteq \mathcal{B}' & \text{if } |\mathcal{B}'| > |\mathcal{B}|. \end{cases}$$

Now define $f_{E_{\leq n}}$ on $\mathcal{M}$ as the function induced by h. We have to check properties (i) – (v) for f_{n+1}.

Conditions (i) and (ii) are obviously fulfilled. Property a) implies immediately (iii).

To verify condition (iv) we distinguish two cases: In case $m_{n+1} = r - 1$ (i.e, no global round switch occured in $A(n)$) we have

$$b'_{n+1}(m_{n+1}) - a_{n+1}(m_{n+1}) \geq b'_n(m_n) - a_n(m_n) - (|\mathcal{D}'| + |\mathcal{G}'| - |\mathcal{D}| - |\mathcal{G}|) = 0,$$

since due to condition a) at most $(|\mathcal{D}'| + |\mathcal{G}'| - |\mathcal{D}| - |\mathcal{G}|)$ more $(r - 1)$-entries in $A'(n)$ can be overwritten by r- or $(r + 1)$-entries than in $A(n)$.

In case $m_{n+1} = r$ (i.e., in $A(n)$ a global round switch occured, this means all $(r - 1)$-entries in $A(n)$ were overwritten by r or $(r + 1)$) let $\mathcal{B}_-$ resp. $\mathcal{B}'_-$

denote the number of succesfully transmitted messages in the sets $\mathcal{B}$ resp. $\mathcal{B}'$. Hence,

$$\begin{aligned} b'_{n+1}(m_{n+1}) &= |\mathcal{F}'_n| + |\mathcal{C}'_n| + |\mathcal{D}'_n| + |\mathcal{B}'_n| - \mathcal{B}'_- + |\mathcal{G}'_n| \\ &\overset{(6)}{=} N(N - 1) - |\mathcal{A}'_n| - \mathcal{B}'_-, \\ a_{n+1}(m_{n+1}) &= |\mathcal{F}_n| + |\mathcal{C}_n| + |\mathcal{D}_n| + |\mathcal{B}_n| - \mathcal{B}_- \\ &\overset{(6)}{=} N(N - 1) - |\mathcal{A}_n| - |\mathcal{G}_n| - \mathcal{B}_- \end{aligned}$$

and consequently,

$$\begin{aligned} b'_{n+1}(m_{n+1}) - a_{n+1}(m_{n+1}) &= |\mathcal{A}_n| - |\mathcal{A}'_n| + |\mathcal{G}| + \mathcal{B}_- - \mathcal{B}'_- \\ &\geq |\mathcal{A}_n| - |\mathcal{A}'_n| + |\mathcal{G}| + \min(0, |\mathcal{B}| - |\mathcal{B}|') \geq 0. \end{aligned}$$

The last inequality holds in case of $\min(|\mathcal{B}|, |\mathcal{B}'|) = |\mathcal{B}'|$ due to condition (v), in case of $\min(|\mathcal{B}|, |\mathcal{B}'|) = |\mathcal{B}|$ due to Eq. 7. Thus also (iv) is valid.

If $R(n + 1) = R'(n + 1)$, we also have to verify condition (v). In case $R(n + 1) = r + 1$ there are no $(r + 2)$-processes and the condition is fulfilled trivially. On the other hand, i.e, in case $R(n + 1) = r$, firstly observe that the difference of $(r + 1)$-entries can decrease by up to $\max(0, |\mathcal{B}'| - |\mathcal{B}|)$ (due to b)). Secondly, use the following partioning of $\mathcal{G}_{n+1}$: We can write

$$\mathcal{G}_{n+1} = (\mathcal{G}_n \cup \mathcal{N}) \setminus \mathcal{G}_-$$

with $\mathcal{N} = \mathcal{G}_{n+1} \setminus \mathcal{G}_n$ the set of 'new' $(r-1)$-positions in $A(n+1, r, r+1)$ (i.e., those $(r-1)$-entries which are in columns of processes which switched from r- to $(r + 1)$-processes from $A(n)$ to $A(n + 1)$) and $\mathcal{G}_-$ the set of arriving (!) messages related to positions in $\mathcal{G}_n$.

Hence,

$$|\mathcal{G}_{n+1}| = |\mathcal{G}_n| + |\mathcal{N}| - |\mathcal{G}_-| \geq |\mathcal{G}_n| + 0 - |\mathcal{G}_-|.$$

Moreover, note that every message in $\mathcal{G}_-$ increases the number of $(r+1)$-entries. Hence,

$$\begin{aligned}|\mathcal{A}_{n+1}| &= |\mathcal{A}_n| + |\mathcal{B}_-| + |\mathcal{G}_-|,\\ |\mathcal{A}'_{n+1}| &= |\mathcal{A}'_n| + |\mathcal{B}'_-|.\end{aligned}$$

Note that we do not need the term $|\mathcal{G}'_-|$ in the second equation since in the only case $\mathcal{G}'$ is nonempty the transmitted message is treated as 0 due to computation step (3) of our bound. This yields

$$\begin{aligned}|\mathcal{A}_{n+1}| + |\mathcal{G}_{n+1}| - |\mathcal{A}'_{n+1}| &\geq |\mathcal{A}_n| + |\mathcal{G}_-| + |\mathcal{B}| + |\mathcal{G}_n| - |\mathcal{G}_-| - |\mathcal{A}'_n| - |\mathcal{B}'|\\ &\geq |\mathcal{A}_n| + |\mathcal{G}_n| - |\mathcal{A}'_n| - \max(0, |\mathcal{B}'| - |\mathcal{B}|) \geq 0.\end{aligned}$$

Note that the second inequality uses property b) of the bijection and the last inequality is again valid due to (v) or Eq. 7. So the proof is finished in Case A.

Case B $R(n) - 1 = R'(n) = r$**:** Given the matrices $A(n)$ and $A'(n)$ we define the following sets (in addition to the sets $\mathcal{A}', \mathcal{B}', \mathcal{F}', \mathcal{C}', \mathcal{D}', \mathcal{G}', \mathcal{A}, \mathcal{B}$ defined above):

$$\begin{aligned}\mathcal{H}_n &= \{\text{positions of non-diagonal } (r+2)\text{-entries in } A(n)\},\\ \mathcal{I}_n &= \{\text{positions of } (r+1)\text{-entries in } A(n, r+2, r+2) \text{ and } A(n, r+1, r+2)\},\\ \mathcal{K}_n &= \{\text{positions of } (r+1)\text{-entries in } A(n, r+2, r+1)\},\\ \mathcal{J}_n &= \{\text{positions of } r\text{-entries in } A(n, r+1, r+2)\}.\end{aligned}$$

A visualisation of the partioning of $A(n)$ and $A'(n)$ into these sets is given in Fig. 3.

Again, denote with $M = \mathcal{I} \cup \mathcal{B} \cup \mathcal{J}$ and $M' = \mathcal{B}' \cup \mathcal{D}' \cup \mathcal{G}'$ the sets of relevant channels. Choose any bijection h: $[N]^2 \to [N]^2$, that keeps the diagonal fixed, with

$$\begin{cases} M' \subseteq h(M) & \text{if } |M| \geq |M'| \\ h(M) \subseteq M' & \text{if } |M'| \geq |M| \end{cases} \tag{8}$$

and

$$\begin{cases} \mathcal{B}' \subseteq h(\mathcal{B} \cup \mathcal{J}) & \text{if } |\mathcal{B}'| \leq |\mathcal{B}| + |\mathcal{J}| \\ h(\mathcal{B} \cup \mathcal{J}) \subseteq \mathcal{B}' & \text{if } |\mathcal{B}'| > |\mathcal{B}| + |\mathcal{J}|. \end{cases} \tag{9}$$

Define $f_{E_{\leq n}}$ as the bijection on $\mathcal{M}$ induced by h. Then conditions (i) and (ii) for f_{n+1} defined in Eq. 5 are obviously fulfilled. The fact $R'(n+1) \leq R'(n)+1$ immediately implies (iii).

Let us we verify (iv). If $m_{n+1} = r+1$ (i.e., a global round switch in $A(n)$ occured), then $m_{n+1} \geq \max A'(n+1)$ and we are done. Otherwise, if $m_{n+1} = r$, observe that $m_n = r$, that $\mathcal{B}$ and $\mathcal{J}$ contain all relevant r-entries of $A(n)$ and $\mathcal{B}'$ contain all those relevant entries in $A'(n)$ which can be overwritten by $(r+1)$. Hence the change in a_n and b'_n can be bounded,

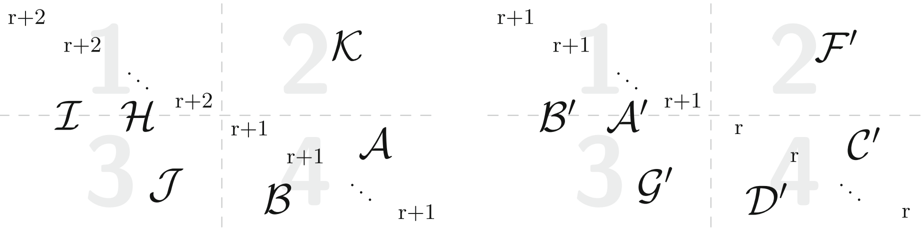


Fig. 3 The partioning of matrices $A(n)$ and $A'(n)$

due to property (9), by

$$\begin{aligned}
b'_{n+1}\ &(m_{n+1}) - a_{n+1}(m_{n+1}) \\
&\geq b'_n(m_n) - a_n(m_n) - \max(|\mathcal{B}'| - |\mathcal{B}| - |\mathcal{J}|, 0) \\
&= |\mathcal{B}'| + |\mathcal{C}'| + |\mathcal{D}'| + |\mathcal{F}'| + |\mathcal{G}'| - |\mathcal{B}| - |\mathcal{J}| - \max(|\mathcal{B}'| - |\mathcal{B}| - |\mathcal{J}|, 0) \\
&= \begin{cases} |\mathcal{B}'| + |\mathcal{C}'| + |\mathcal{D}'| + |\mathcal{F}'| + |\mathcal{G}'| - |\mathcal{B}| - |\mathcal{J}| \geq 0 & \text{if } |\mathcal{B}'| \leq |\mathcal{B}| + |\mathcal{J}| \\ |\mathcal{C}'| + |\mathcal{D}'| + |\mathcal{F}'| + |\mathcal{G}'| \geq 0 & \text{if } |\mathcal{B}'| > |\mathcal{B}| + |\mathcal{J}|, \end{cases}
\end{aligned}$$

and so (iv) is proven.

To finish the proof in Case B, it remains to check condition (v) in case of $R(n+1) = R'(n+1) = r+1$, i.e., a global round switch occured also in $A'(n)$, but then there are no $(r+2)$-processes in $A'(n+1)$ and so (v) is trivially fulfilled.

Case C $R(n) > R'(n) + 1$**:** Here we define $f_{E_{\leq n}} := \mathrm{id}_{\mathcal{M}}$. The conditions (i) and (ii) are clear, and we can ignore (v). From $R(n) > R'(n) + 1$ we obtain $R(n+1) > R(n+1)'$ and thus (iii). Moreover, we have $\min A(n) \geq \max A'(n)$ and hence $A(n+1)_{i,j} \geq A'(n+1)_{i,j}$ for all $1 \leq i, j \leq N$; this implies (iv). □

3.4 Asymptotics for $p \to 0$

In this section, we give the asymptotics of the behavior of the α'-synchronizer for $p \to 0$, which we argued in Section 1 to be the appropriate regime for p for wireless networks with destructive collisions (Moscibroda and Wattenhofer 2006).

For $M \geq 1$, let

$$\Lambda(M, p) = \mathbb{E} \max_{1 \leq i \leq M} G_i,$$

where the G_i are pairwise independent geometrically distributed random variables with parameter p. For $\Lambda(M, p)$, the asymptotics for $p \to 0$ is an immediate consequence of Füger et al. (2015, Proof of Theorem 3):

Lemma 1 *For* $p \to 0$,

$$\Lambda(M, p) \sim \frac{c_M}{p} \qquad \textit{with} \qquad c_M = \sum_{i=1}^{M} \binom{M}{i} \frac{(-1)^{i+1}}{i} = H_M,$$

where H_n *denotes the* n*-th harmonic number.*

The expected round duration λ' can be written in terms of these expectations: Let $\hat{a}(t)$ be the Markov chain obtained from $a(t)$ by adding to each state a an additional flag Step such that $\mathrm{Step}(\hat{a}(t)) = 1$ if there is a global round switch at time t, and 0 otherwise. Then we have

$$\lambda' = \frac{1}{\sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \hat{\pi}(\hat{a})} \sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \hat{\pi}(\hat{a}) \cdot \Lambda(\#_{-1}(\hat{a}), p),$$

where $\hat{\pi}$ denotes the steady-state distribution of the Markov chain $\hat{a}(t)$ and $\#_{-1}(\hat{a})$ is the number of (-1)-entries in $\hat{a}$ (see Függer et al. (2015) for details). This formula shows that λ' can be represented as a weighted sum (weighted by the steady-state probabilities) of the expected time to the next global round switch starting in state $\hat{a}$, where we only take those states which can occur after a global round switch.

Note that this representation is equivalent to the following one: Let $b(r)$ be the Markov chain only on the states which can occur after a global round switch, i.e., on the states $\hat{a}$ with $\mathrm{Step}(\hat{a}) = 1$ with steady state distribution ρ. Then

$$\lambda' = \sum_b \rho(b) \cdot \Lambda(\#_{-1}(b), p) = \sum_{i=N}^{N^2-N} \rho(i) \cdot \Lambda(i, p) \tag{10}$$

with $\rho(i) = \sum_{b:\ \#_{-1}(b)=i} \rho(b)$.

Moreover, a standard method to compute the stationary distribution of a Markov chain is the following: Let P denote the transition matrix of a homogeneous and irreducible Markov chain with finite state space. Then the steady state distribution π is the solution of the equation $B\pi = e$ with $e = (0, \ldots, 0, 1)^T$ and $B = ((P - I)^{(n \to 1)})^T$, where $M^{(k \to x)}$ denotes a matrix M with its k-th column set to x. For all states s of the Markov chain we have $\pi(s) > 0$, and by Cramer's rule $\pi(s) = \det(B_s)/\det(B)$ holds, where B_s is the matrix arising from B by replacing the column corresponding to s by e. In particular, all matrices B_s and B are regular.

Now we can establish the asymptotics for the α'-synchronizer:

Theorem 2 *For $p \to 0$,*

$$\lambda' \sim \frac{\theta'_N}{p} \qquad \textit{with} \qquad \theta'_N = \sum_{k=0}^{N(N-2)} \mathbb{P}(\#1 = k) \cdot c_{N^2-N-k},$$

where

$$\mathbb{P}(\#1 = k) = \sum_{\substack{(e_1,\ldots,e_{N-1}) \\ e_i \geq 0 \\ \sum e_i = k \\ \sum_{j=1}^{i} e_j \leq (N-1)i - \delta_{N-1,i}}} \prod_{i=1}^{N-1} \mathbb{P}\left(\#(\textit{new 1s in } \langle i, i+1\rangle) = e_i \,|\, \#(\textit{old 1s}) = \sum_{j=1}^{i-1} e_j\right) \tag{11}$$

for $0 \leq k \leq N(N-2)$ *and*

$$\mathbb{P}(\#(\textit{new 1s in } \langle d, d+1\rangle) = e \,|\, \#(\textit{old 1s}) = g) = \begin{cases} P_1(N, d, e, g) & \textit{if } 1 \leq d \leq N-2 \\ P_2(N, e, g) & \textit{ow.} \end{cases}$$

with

$$P_1(N, d, e, g) = \frac{N-d}{d(N-1) - g - e + N - d} \prod_{s=0}^{e-1} \frac{d(N-1) - g - s}{d(N-1) - g - s + N - d} \tag{12}$$

$$P_2(N, e, g) = \frac{1}{(N-1)^2 - g - e} \prod_{s=0}^{e-1} \frac{(N-1)^2 - g - s - 1}{(N-1)^2 - g - s} \tag{13}$$

Remark 2 Let us give a rough explanation of formula Eq. 11 (see Step 3 of the proof for details): With # we denote 'the number of', and let $\langle d, d+1\rangle$ be the time interval from switching from d to $(d+1)$ 1-processes. Then $\mathbb{P}(\#1 = k)$ stands for the probability that the number of (non-diagonal) 1s in the matrix a just before a global round switch is performed equals k and $\mathbb{P}(\#(\text{new 1s in } \langle d, d+1\rangle) = e | \#(\text{old 1s}) = g)$ is the probability that in $\langle d, d+1\rangle$ e new 1-entries in the matrix a are created given that we have already had g non-diagonal 1-entries before.

Proof

First step: We will observe that for the limit $p \to 0$ it is enough to consider a restricted model in which we allow only those transitions to be possible which arise from the successful arrival of at most one relevant message in one time step: Denote with P the transition matrix for the Markov chain $\hat{a}$ for the α'-synchronizer and consider the reduced matrix $\tilde{P} = P \pmod{p^2}$ (i.e., we delete all powers of order greater than or equal to 2 of p in P). Then $\tilde{P}$ is still a stochastic matrix (for p sufficiently small: the row sum is still one since we deleted all higher powers, but the remaining terms of the form $(1 - dp)$ could be negative) and corresponds to the model in which only at most one arriving message is allowed. Since every transition with $k > 1$ arriving messages can be simulated by k transitions with only 1 arriving message, the corresponding markov chain is still homogeneous, aperiodic and irreducible, and thus has a unique steady state distribution. In particular, $\tilde{B} = ((\tilde{P} - I)^{(n \to 1)})^T$ and $\tilde{B}_s$ are regular.

Since $\hat{\pi}(\hat{a}) = \det(B_{\hat{a}}) / \det(B)$, and all entries of B and $B_{\hat{a}}$ are polynomials in p, the involved determinants are polynomials in p, too, and therefore it is sufficient for the limit $p \to 0$ to know the term with the lowest power of p. Since for a matrix $M = (m_{i,j})$ we have

$$\det M = \sum_{\sigma \in S_n} \left(\operatorname{sgn}(\sigma) \prod_{i=1}^{n} m_{i,\sigma(i)} \right),$$

we just need to know the lowest power of p in each row (provided they do not cancel out in the sum). For B, note that the last row consists only of 1s, whereas in all the other rows the lowest power is p (the only entries where 1 could appear is in the diagonal when no message arrives and we stay in the same state, but we have substracted the indentity matrix!). Hence the term with the lowest power of p in $\det B$ is of the form $c \cdot p^{r(B)-1}$ (with $r(B)$ the number of rows of the matrix B), but this is nothing else than $\det(B \pmod{p^2})$. But $B \pmod{p^2}$ is the same as $((\tilde{P} - I)^{(n \to 1)})^T$, which is regular as mentioned above, and hence we can use the restricted model.

Second step: Let σ_s and σ_s' be two states that can occur after a global round switch. If we pass from σ_s to σ_s' then we have to go through the state

$$\hat{\sigma} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 0 \end{pmatrix}.$$

To see this, assume that we are in state σ_s and let k denote the number of (non-diagonal) zeros in σ_s. Due to computation step (3) of our upper bound we have $k \leq N(N-2)$, and due to (1a) each row contains at least one (-1)-entry. Again due to (1a) new zeros are filled in according to our construction 'column by column', but at most one zero in one time step. Hence, we have to pass $\hat{\sigma}$.

Third step: Now we want to compute the probability $\mathbb{P}(\#1 = k)$ of exactly k ones occuring at a global round switch given that we start from state $\hat{\sigma}$ (in fact, $\rho(N^2 - N - k) = \mathbb{P}(\#1 = k)$ with ρ from Eq. 10). When passing from $\hat{\sigma}$ to the next state σ_s after a global round switch, let t_d denote the time when we switch to a state with d one-processes. In a first step we will determine the probability $\mathbb{P}(\#(\text{new 1s in } [t_d, t_{d+1})) = e | \#1 = g)$, i.e., the probability of exactly e new 1-entries in the interval $[t_d, t_{d+1})$ (i.e., the time until the next 0-process becomes a 1-process) given already g 1-entries in the matrix $a(t_d)$. Assume $d \leq N-2$ first.

At every time there are three possibile events: no new message arrives ('n', i.e., the matrix remains unchanged), a new 1-message arrives ('1', i.e., a 0-entry is overwritten by 1) or a new 0-message arrives ('0', i.e., a (-1) is overwritten by 0 and the next 0-process becomes a 1-process). Thus all sequences of events in the interval $[t_d, t_{d+1})$ are of the form

$$n^{k_1} 1 n^{k_2} 1 \cdots n^{k_e} 1 n^{k_{e+1}} 0$$

with $k_i \geq 0$ $(0 \leq i \leq e+1)$.

Observe that the probability of the succesfully transmitted 0-message equals $(N-d)p$ (recall that we have $(N-d)$ 0-processes), the probability of the arrival of the s-th new 1-message equals $(d(N-1)-g-(s-1))p$ (we have $d(N-1)$ channels outgoing from 1-processes, and we have to subtract the number of 1s already arrived). Thus, the s-th sequence n^{k_s} appears with probability $(1-(d(N-1)-g-(s-1)+N-d)p)^{k_s}$. Hence, by summing over all possible lengths $k_s \geq 0$ and multiplying these three factors,

$$\begin{aligned}
&\mathbb{P}(\#(\text{new 1s in } [t_d, t_{d+1})) = e \mid \#1 = g) \\
&\quad = (N-d)p \cdot p^e \prod_{s=0}^{e-1} (d(N-1)-g-s) \times \\
&\qquad \times \prod_{s=0}^{e} \sum_{k \geq 0} (1-(d(N-1)-g-s+N-d)p)^k \\
&\quad = \frac{N-d}{d(N-1)-g-e+N-d} \prod_{s=0}^{e-1} \frac{d(N-1)-g-s}{d(N-1)-g-s+N-d}.
\end{aligned}$$

Thus Eq. 12 is shown.

In case $d = N-1$ the situation changes a little bit: Here we have only $(d(N-1)-1) = ((N-1)^2-1)$ channels outgoing from 1-processes not leading to the global round switch, hence the probability of the arrival of the s-th new 1-message equals $((N-1)^2-1-g-(s-1))p$. Analogous to above we obtain (13).

In a second step we compute $\mathbb{P}(\#1 = k)$: To obtain the probability of k ones at the global round switch we have to sum over all weak compositions of k with $(N-1)$ parts (a weak composition of k with m parts is an ordered collection of m non-negative integers whose sum is k). Moreover, the number of arrived 1s cannot exceed the number of possible 1-channels. Hence (11) is shown.

Fourth Step: Now we can determine the constant: We have (see Eq. 10)

$$\lambda' = \sum_{k=0}^{N(N-2)} \mathbb{P}(\#1 = k) \Lambda(N^2-N-k, p).$$

By Lemma 1, the stated asymptotics follows.

□

3.5 Relation to the Existing Upper Bound

In this section, we will prove that our new upper bound provided by α' is better than the one presented in Függer et al. (2015). The upper bound for the α-synchronizer given in Függer et al. (2015) was given by $\lambda^{\mathrm{III}}(N, p) = \Lambda(N(N-1), p)$, which originated from a synchronizer variant whose knowledge is reset after each global round switch. The following Corollary 1 shows that λ^{III} is also an upper bound for λ', which confirms our claim. Moreover, it establishes that our new upper bound has a strictly better asymptotic behavior for $p \to 0$.

Corollary 1 *Let θ denote the constant such that $\lambda^{\mathrm{III}}(N, p) \sim \theta / p$ for $p \to 0$. Then*

– $\lambda'(N, p) \leq \lambda^{\mathrm{III}}(N, p)$ *and*
– $\theta' < \theta = H_{N(N-1)}$.

Proof To show $\lambda'(N, p) \leq \lambda^{\mathrm{III}}(N, p)$, we define a synchronizer α'_f in the same way as α', except that we reset knowledge after every global round switch in α'_f , i.e., we introduce a new "forgetting" step (4):

(4) If $md^{(1)}(t) \neq md^{(2)}(t)$, then $A^{(4)}_{i,j}(t) = md^{(1)}(t)$ for all $i \neq j$.

Let A and A^f denote the matrices of the α'- resp. α'_f-synchronizer. Then the relation $A_{i,j}(t) \leq A^f_{i,j}(t)$ for every t, i, j can be seen directly from comparing A and A^f executionwise. Secondly, to see that α'_f is equivalent to λ^{III}, it is sufficient to note the following two properties:

(i) Step (3) is redundant for α'_f due to the forgetting step (4).
(ii) The reordering step (1a) cannot affect the expected time until the next global round switch, since we reset all knowledge and thus all $N(N-1)$ messages have to arrive every round.

Showing that our new bound has a better asymptotic behavior for $p \to 0$ is straightforward:

$$\theta'_N = \sum_{k=0}^{N(N-2)} \mathbb{P}(\#1 = k) \cdot c_{N^2-N-k} < c_{N^2-N} \sum_{k=0}^{N(N-2)} \mathbb{P}(\#1 = k) = c_{N^2-N-k}$$

By Lemma 1, $\theta = H_{N(N-1)}$, which concludes the proof. □

Remark 3 If one compares the the first few values of θ and θ' given below, it is apparent that our new upper bound is indeed strictly better than the old one.

	$N = 3$	4	5	6	7	8
θ	2,4500	3,1032	3,5977	3,9950	4,3267	4,6115
θ'	2,0965	2,5178	2,8378	3,0940	3,3069	3,4886

4 New Lower Bound

Similar to the upper bound, we can also construct a lower bound for the α-synchronizer. We will start by presenting the algorithm that generates our new lower bound, which will be called the α''-synchronizer. Subsequently, we will give the proof that it is indeed a lower bound. In contrast to the upper bound, we will now choose the channels on which messages arrive in a 'good' way, i.e., we try to generate 1-entries in the diagonal as fast as possible. The basic idea is to fill in incoming messages received by 0-processes now line by line. But to make our proof working, we have to speed up our algorithm even more: Roughly speaking, we will treat entries below the diagonal as 1-entries.

We adapt the steps (1a) and (3) of the α'-synchronizer in Section 3.1. Let $md := md^{(1)}(t)$ and $w := w^{(1)}(t)$, cf. Section 2.3.

(1a) *Partial reordering of the knowledge*: Let $(k_i)_{i=1}^{w(w-1)}$ be a decreasing sorting of the multiset $\{A^{(1)}_{i,j}(t) \ : \ i \neq j, \ A^{(1)}_{i,i}(t) = md\}$, i.e., a sorting only of the knowledge of the md-processes (note that the rows of $(md+1)$-processes remain unchanged!). Now fill in this sequence into the rows of 0-processes again according to the following rule: Fill them in line by line, but if a non-$(md-1)$-entry is to be placed into $A^{(1)}(t, md, md)$ replace it by a md, and if it is to be placed into $A^{(1)}(t, md, md+1)$ replace it by $(md+1)$.

This means that we not only reorder the knowledge, we also sometimes change the values of the knowledge depending on the round number of the new sender. Note the following:

(α) Firstly, this reordering step causes the diagonal of $a(n)$ to have all 1-processes before the 0-processes, i.e., the diagonal is of the form $(1, \dots, 1, 0, \dots, 0)$. If the 1s of the diagonal are on positions $1, \dots, j-1$, we call process j the *first 0-process* (of $a(n)$).

(β) Secondly, this reordering implies that below of 1-entries in the diagonal we either have 1- or (-1)-entries.

(3) a: If $md^{(2)}(t) = md^{(1)}(t)$ and if there are $(md^{(2)}+1)$-processes in $A^{(2)}$ then replace all $md^{(2)}(t)$-entries below the diagonal in $A^{(2)}(t)$ by $(md^{(2)}(t)+1)$.
b: If $md^{(2)}(t) > md^{(1)}(t)$, we do again a reordering of the knowledge: take all non-diagonal elements of $A^{(2)}$ and fill them in again line by line in decreasing order.

Step (3a) means that if the system does not perform a global round switch and if there are processes in front we increase some knowledge below the diagonal. This modification is necessary to make our proof work – it avoids some specific executions in which the α-synchronizer would overtake the lower bound: Roughly speaking, it can happen that in the lower bound several 0-processes become 1-processes at the same time step, whereas in the α-synchronizer they switch step by step. Then, in the lower bound, there are 0-entries between the 'new' 0-processes below the diagonal. But in the α-synchronizer there are still (-1)-entries at those positions which will be overwritten by 1s afterwards step by step. Hence the α-synchronizer could be in front.

Step (3b) ensures that also after a global round switch we immediately have knowledge filled in in decreasing order. Note that omitting this step would lead to a mathematical equivalent synchronizer, since reordering would be done in the next time step in step (1a), but with this additional step the synchronizer is easier to handle in the following.

Example 7 This example illustrates the α''-synchronizer and compares it to the α-synchronizer:

$$\begin{array}{ccccccccccc}
t: & 0 & & 1 & & 2 & & 3 & & 4 & \\
a(t): & \begin{pmatrix}0&-1&-1\\-1&0&-1\\-1&-1&0\end{pmatrix} & \to & \begin{pmatrix}0&-1&0\\-1&0&0\\0&0&1\end{pmatrix} & \to & \begin{pmatrix}0&-1&0\\0&1&0\\0&0&1\end{pmatrix} & \to & \begin{pmatrix}0&-1&0\\0&1&0\\0&0&1\end{pmatrix} & \to & \begin{pmatrix}0&0&0\\-1&0&0\\-1&0&0\end{pmatrix} & \to\cdots \\
\mathcal{P}(t): & & \begin{pmatrix}1&0&1\\0&1&1\\1&1&1\end{pmatrix} & \to & \begin{pmatrix}1&0&0\\1&1&0\\1&1&1\end{pmatrix} & \to & \begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix} & \to & \begin{pmatrix}1&1&1\\0&1&1\\1&1&1\end{pmatrix} & \to\cdots & \\
a''(t): & \begin{pmatrix}0&-1&-1\\-1&0&-1\\-1&-1&0\end{pmatrix} & \to & \begin{pmatrix}1&0&0\\1&1&0\\-1&-1&0\end{pmatrix} & \to & \begin{pmatrix}0&0&0\\0&0&-1\\-1&-1&0\end{pmatrix} & \to & \begin{pmatrix}1&0&0\\1&0&-1\\-1&-1&0\end{pmatrix} & \to & \begin{pmatrix}0&0&0\\-1&0&-1\\-1&-1&0\end{pmatrix} & \to\cdots
\end{array}$$

Here the α-synchronizer does only one global round switch (at time $t = 4$), but the α''-synchronizer performs two (at times $t = 2$ and $t = 4$). So the α''-synchronizer is indeed faster than the original α-synchronizer. We will see later on, however, that—when provided with the same sequence of matrices $\mathcal{P}(t)$—this speed-up does not always happen. Note that at times $t = 1$ and $t = 3$ we applied step (3a), whereas at $t = 2$ we applied step (3b).

Formally, the lower bound can be described as follows: We consider the number of 0-entries in the diagonal (i.e., 0-processes) x, the total number of 1-entries above the diagonal y, and the total number of 0- or 1-entries z in rows of 0-processes, where

$$1 \le x \le N, \qquad 0 \le y \le n(x)/2, \qquad 0 \le z \le \begin{cases} N-2 & \text{if } x < N \\ (N-1)^2 & \text{if } x = N, \end{cases} \tag{14}$$

with

$$n(x) = (N-x)(N-x-1).$$

Note, due to the reordering step (1a), that the non-(-1)-entries in rows of 0-processes are located in the row of the first 0-process if $x < N$. In case $x = N$ we can have up to $(N-1)^2$ 0-entries because after a global round switch the 1-entries become 0-entries, and we can have up to $(N-1)$ 1-processes and thus up to $(N-1)^2$ non-diagonal 1-entries before performing the round switch. So the number of states of the Markov chain equals $(N^4 - N^3 + 5N^2 - 11N + 12)/6$.

Moreover, let $\mu(y) = (N-1)y$ and $s_1 = (x_1, y_1, z_1)$ and $s_2 = (x_2, y_2, z_2)$ be two states of our Markov chain. Then, the transition probabilities $p_{s_1s_2}$ are given by $p_{s_1s_2} = \tilde{p}_{s_1s_2} + \hat{p}_{s_1s_2}$, where $\tilde{p}_{s_1s_2}$ is the probability of a transition from s_1 to s_2 without making a global round switch and $\hat{p}_{s_1s_2}$ is the probability of a transition performing a global round switch:

$$\tilde{p}_{s_1s_2} = \begin{cases} \binom{n(x_1)/2 - y_1}{-\Delta(y)}\binom{\mu(x_1) - z_1}{-\Delta(z)} \times \\ \quad \times\, p^{-\Delta(y+z)}(1-p)^{n(x_1)/2 - y_2 + \mu(x_1) - z_2} \\ \quad \text{if } x_1 = x_2,\, y_1 \le y_2,\, z_1 \le z_2,\, z_2 \le N-2 & \text{(15a)} \\ \binom{n(x_1)/2 - y_1}{-\Delta(y)}\binom{\mu(x_1) - z_1}{\mu(x_1 - x_2) - \Delta(z)} \times \\ \quad \times\, p^{\mu(x_1 - x_2) - \Delta(y+z)}(1-p)^{n(x_1)/2 - y_2 + \mu(x_2) - z_2} \\ \quad \text{if } x_1 > x_2,\, z_2 \le N-2,\, y_1 \le y_2 \le n(x_1)/2 & \text{(15b)} \\ 0 \qquad\qquad\qquad \text{ow.} & \text{(15c)} \end{cases}$$

In both cases, we have to bound z_2 by $(N-2)$ since states with $z_1 > (N-2)$ can only occur directly after a global round switch as all 0-processes with $(N-1)$ 0-entries would perform a local round switch and become 1-processes otherwise. Let us turn to the transition probabilities in detail:

Equation 15a: If the number of 0-processes remains constant, the number of 1-entries above the diagonal and the number of non-(-1)-entries of the first 0-process trivially cannot decrease. For the number of ones we have $(n(x_1)/2 - y_1)$ possible channels to choose $(y_2 - y_1)$ new entries. For the new entries of the first 0-process we have $(\mu(x_1) - z_1)$ possible channels for the new entries. This immediately gives the transitions probabilities in this case.

Equation 15b: If the number of 0-processes decreases, the number of 1-entries above the diagonal cannot decrease and is clearly bounded by $n(x_1)/2$. To generate $(x_1 - x_2)$ new 1-processes and z_2 non-(-1)-entries at the first 0-process $(\mu(x_1 - x_2) - z_1) + z_2$ messages overwriting (-1)-entries must arrive (the first summand equals the number of non-diagonal entries in $(x_1 - x_2)$ rows) minus the number of already existing non-(-1)-entries z_1 in rows of 0-processes). This gives the stated formula in this case.

For the transition probability of a transition performing a global round switch, we have

$$\hat{p}_{s_1 s_2} = \begin{cases} \binom{n(x_1)/2-y_1}{z_2-x_1(N-x_1)-y_1-n(x_1)/2} p^{z_2+x_1(x_1-1)-y_1-n(x_1)/2-z_1}(1-p)^{\nu(x_1)-z_2} \\ \qquad \text{if } x_1 \le x_2 = N,\ y_1 + x_1(N-x_1) + n(x_1)/2 \le z_2 \le \nu(x_1) \\ 0 \qquad \text{ow.} \end{cases} \tag{16}$$

After making a global round switch, the number of 0s in the diagonal equals N. For the number of zeros z_2 in s_2 recall that these 0-entries arise from 1-entries by applying the normalization step. Thus, the number of non-diagonal 0-entries z_2 must be at least the number of 1-entries above the diagonal before (i.e., y_1) plus all entries below the diagonal of 1-processes (for performing the global round switch all messages must arrive and below 1-processes we have only 1-entries – see (β)), this number equals $n(x_1)/2 + x_1(N - x_1)$. Since we had $(N - x_1)$ 0-processes in state s_1 and each process can send messages to at most $(N-1)$ other processes, we have $z_2 \le \nu(x_1)$.

Since all (-1)-entries must be overwritten and since due to (β) no 0-entries exists below 1s in the diagonal, the only choice we have for relevant messages leading to 1-entries (before normalizing) which may arrive or not is to choose them from the $(n(x_1)/2 - y_1)$ 0-entries above the diagonal; because of the argument above we have to choose $z2 - (y_1 + x_1(N - x_1) + n(x_1)/2)$ from them. This gives the binomal coefficient.

The total number of messages which must arrive for this state transition is the number of new 1-entries above the diagonal plus the number of (-1)-entries in s_1, i.e.,

$$\begin{aligned} &(z_2 - x_1(N-x_1) - y_1 - n(x_1)/2) \ + (x_1(N-1) - z_1) \\ &\qquad = \qquad z_2 + x_1(x_1-1) - y_1 - n(x_1)/2 - z_1; \end{aligned}$$

this is the exponent of p. Finally, computing the exponent of $(1-p)$, we have

$$\begin{aligned} &(n(x_1)/2 - y_1) \ - \ (z_2 - x_1(N-x_1) - y_1 - n(x_1)/2) \\ &\qquad = n(x_1) + x_1(N-x_1) - z_2 = \nu(x_1) - z_2 \end{aligned}$$

4.0.1 Proof Lower Bound

Now we want to show that our previously defined process is indeed a lower bound. As before this can not be done execution-wise, since there exist schedules such that the α-synchronizer is faster than the α''-synchronizer.

Example 8 We give an example of a schedule such that the α-synchronizer is faster than the α''-synchronizer. Here $A''(t)$ and $A(t)$ denote the matrices as defined in Section 2.3.

$$t\colon \quad 0 \qquad 1 \qquad 2 \qquad 3 \qquad 4$$

$$A(t)\colon \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \end{pmatrix} \to \cdots$$

$$\mathcal{P}(t)\colon \quad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \to \cdots$$

$$A''(t)\colon \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \to \cdots$$

As before we will construct a measure-preserving bijection g on the sets of schedules such that the α-synchronizer under a schedule E is always slower than the α''-synchronizer given the schedule $g(E)$. Again, the basic idea behind our construction is to map the relevant messages of the α-synchronizer on the relevant messages of the α''-synchronizer in such a way that the α''-synchronizer is always in front.

Then we have the following theorem:

Theorem 3 *There exists a bijection $g : \mathcal{E} \to \mathcal{E}$ with*

- $|E_n| = |g(E)_n| \quad \forall\, E \in \mathcal{E},\ \forall n$ *and*
- $R(E, n) \leq R''(g(E), n) \quad \forall\, E \in \mathcal{E},\ \forall n.$

Here, $R(E, n)$ denotes the global round number of the α-synchronizer at time n given the schedule E and $R''(E'', n)$ the global round number of the α''-synchronizer at time n given the schedule E''.

Proof Let $A(n) = A(E, n)$ and $A''(n) = A''(g(E), n)$ denote the matrices of the α- and the α''-synchronizer under the schedule E and $g(E)$, resp. Moreover, define $m''_n = \min_{i,j} A''(n)_{i,j}$ $(i \neq j)$ as the minimum of $A''(n)$, and $a''_n(m''_n)$ as the number of non-diagonal entries of $A''(n)$ that equal the minimum m''_n. With $b_n(m''_n)$ we denote the number of non-diagonal entries in $A(n)$ less than or equal to m''_n. Similarly, $R(n) = R(E, n)$ and $R''(n) = R''(g(E), n)$ are the global round numbers of $A(n)$ and $A''(n)$, respectively.

Now we will inductively construct functions $g_n : \mathcal{E}_n \to \mathcal{E}_n$ $(n \geq 1)$, the limit of which gives g, with the following properties:

(i) g_n bijective on $\mathcal{E}_n$,

(ii) $|E_n| = |(g_n(E))_n| \quad \forall\, E \in \mathcal{E}_n$,

(iii) $R(E, n) \leq R''(g_n(E), n) \quad \forall\, E \in \mathcal{E}_n$,

(iv) $a''_n(m''_n) \leq b_n(m''_n)$ and

(v) if $R(n) = R''(n)$: $|\mathcal{A}_n| + |\mathcal{G}_n| \leq |\mathcal{A}''_n| + |\mathcal{G}''_n|$ where (for matrices $A(n)$ and $A''(n)$ with $r = R(n) = R''(n)$)

$$\begin{aligned} \mathcal{A}_n, \mathcal{A}''_n &= \{\text{positions of non-diagonal } (r+1)\text{-entries in } A(n) \text{ resp. } A''(n)\}, \\ \mathcal{G}_n, \mathcal{G}''_n &= \{\text{positions of } (r-1)\text{-entries in } A(n, r, r+1) \text{ resp. } A''(n, r, r+1)\}. \end{aligned}$$

(vi) if $R(n) = R''(n)$: $|\tilde{\mathcal{A}}_n| \leq |\tilde{\mathcal{A}}''_n|$, where $\tilde{\mathcal{A}}_n$ resp. $\tilde{\mathcal{A}}''_n$ denote the set of positions of $(r+1)$-entries in $A(n, r+1, r+1)$ resp. $A''(n, r+1, r+1)$ above the diagonal.

(vii) The function g_n is an extension of g_{n-1}, i.e., if $g_n((x_1, \ldots, x_{n-1}, x_n)) = (y_1, \ldots, y_{n-1}, y_n)$ then $g_{n-1}((x_1, \ldots, x_{n-1})) = (y_1, \ldots, y_{n-1})$ for $n \geq 2$.

Then the function g defined by $g(E) = \lim_{n\to\infty} g_n(E_{\leq n})$ has the stated properties.

Remark 4 It is worthwile to mention that conditions (iv) and (v) imply $\sum_{i\neq j} A(E, n)_{i,j} + |\mathcal{G}_n| \leq \sum_{i\neq j} A''(g_n(E), n)_{i,j} + |\mathcal{G}''_n|$.

We start with $g_1 := \mathrm{id}_{\mathcal{M}}$. Then g_1 fulfils obviously $(i) - (vi)$. Let g_n with $(i) - (vi)$ be already defined. We will first construct – in dependence on $E_{\leq n}$ (and consequently on $A(n)$) – the functions $g_{E_{\leq n}}\colon \mathcal{M} \to \mathcal{M}$. Then, the function g_{n+1} is defined as follows:

$$g_{n+1}(E) = g_{n+1}(E_{\leq n}, E_{n+1}) = (g_n(E_{\leq n}), g_{E_{\leq n}}(E_{n+1})). \tag{17}$$

For our construction, we have to do a case distinction:

Case A $R(n) = R''(n) = r$**:**

Given the matrices $A(n)$ and $A''(n)$, we define the sets $\mathcal{A}, \mathcal{A}'', \ldots$ analogously to the proof of Theorem 1 for the upper bound. Moreover, let $\tilde{\mathcal{B}}_n$ resp. $\tilde{\mathcal{B}}''_n$ denote the set of positions of r-entries in $A(n, r+1, r+1)$ resp. $A''(n, r+1, r+1)$ above the diagonal.

Our assumptions imply $|\mathcal{A}|+|\mathcal{G}| \leq |\mathcal{A}''|+|\mathcal{G}''|$ (due to (v)) and $|\mathcal{D}|+|\mathcal{G}| \geq |\mathcal{D}''|+|\mathcal{G}''|$ (see (iv): we have $m''_n = r-1$). Similar to the upper bound we have due to (iv) and reordering step (1a) where messages are inserted line by line, that the number of r-processes in A_n is greater than or equal to the number of r-processes in A''_n; equivalently $|\mathcal{C}| + |\mathcal{D}| \geq |\mathcal{C}''| + |\mathcal{D}''|$ (the sizes of the submatrices $A(n, r, r)$ and $A''(n, r, r)$) or equivalently

$$|\mathcal{A}| + |\mathcal{B}| + |\mathcal{G}| \leq |\mathcal{A}''| + |\mathcal{B}''| + |\mathcal{G}''| \tag{18}$$

(i.e., the total number of entries in columns in $A(n)$ with $(r+1)$ in the diagonal is less than or equal to the number of entries in columns in $A''(n)$ with $(r+1)$ in the diagonal). In fact – by adding a term counting the difference of $(r+1)$-processes in $A(n)$ and $A''(n)$ – we even have

$$|\mathcal{A}| + |\mathcal{B}| + |\mathcal{G}| + (x_n - x''_n)(N-1) = |\mathcal{A}''| + |\mathcal{B}''| + |\mathcal{G}''| \tag{19}$$

with x_n resp. x''_n the number of r-processes in $A(n)$ resp. $A''(n)$. Furthermore, we have the relations

$$|\mathcal{B}''| = \frac{(N - x''_n)(N - x''_n - 1)}{2} - |\tilde{\mathcal{A}}''| \quad \text{and} \tag{20}$$

$$|\mathcal{B}| \geq |\tilde{\mathcal{B}}| = \frac{(N - x_n)(N - x_n - 1)}{2} - |\tilde{\mathcal{A}}|. \tag{21}$$

Let h be any bijection from $[N]^2$ to $[N]^2$ keeping the diagonal fixed with

a) $h(\mathcal{D} \cup \mathcal{G}) \subseteq \mathcal{D}'' \cup \mathcal{G}''$ and

b)

$$\begin{cases} \mathcal{B}'' \subseteq h(\mathcal{B}) & \text{if } |\mathcal{B}''| \leq |\mathcal{B}| \\ h(\mathcal{B}) \subseteq \mathcal{B}'' & \text{if } |\mathcal{B}''| > |\mathcal{B}|, \end{cases}$$

and

c)

$$\begin{cases} \tilde{\mathcal{B}}'' \subseteq h(\tilde{\mathcal{B}}) & \text{if } |\tilde{\mathcal{B}}''| \leq |\tilde{\mathcal{B}}| \\ h(\tilde{\mathcal{B}}) \subseteq \tilde{\mathcal{B}}'' & \text{if } |\tilde{\mathcal{B}}''| > |\tilde{\mathcal{B}}| \end{cases}.$$

Now define $g_{E_{\leq n}}$ on $\mathcal{M}$ as the function induced by h. We have to check properties (i) – (vi) for g_{n+1}.

Conditions (i) and (ii) are obviously fulfilled. Property a) implies immediately (iii).

To verify condition (iv) we distinguish two cases: In case $m''_{n+1} = r - 1$ (i.e, no global round switch occured) we have

$$b_{n+1}(m''_{n+1}) - a''_{n+1}(m''_{n+1}) \ge b_n(m''_n) - a''_n(m''_n) - (|\mathcal{D}| + |\mathcal{G}| - |\mathcal{D}''| - |\mathcal{G}''|) = 0,$$

since due to condition a) at most $(|\mathcal{D}| + |\mathcal{G}| - |\mathcal{D}''| - |\mathcal{G}''|)$ more $(r-1)$-entries in $A(n)$ can be overwritten by r- or $(r+1)$-entries than in $A''(n)$.

In case $m''_{n+1} = r$ (i.e., in $A''(n)$ a global round switch occured) let $\mathcal{B}_-$, $\mathcal{B}''_-$, and $\mathcal{G}_-$ denote the number of succesfully transmitted messages in the sets $\mathcal{B}$, $\mathcal{B}''$, and $\mathcal{G}$. Hence,

$$\begin{aligned} b_{n+1}(m''_{n+1}) &= |\mathcal{F}_n| + |\mathcal{C}_n| + |\mathcal{D}_n| + |\mathcal{B}_n| - \mathcal{B}_- + |\mathcal{G}_n| - \mathcal{G}_- \\ &\overset{(6)}{=} N(N-1) - |\mathcal{A}_n| - \mathcal{B}_- - \mathcal{G}_-, \\ a''_{n+1}(m''_{n+1}) &= |\mathcal{F}''_n| + |\mathcal{C}''_n| + |\mathcal{D}''_n| + |\mathcal{B}''_n| - \mathcal{B}''_- \\ &\overset{(6)}{=} N(N-1) - |\mathcal{A}''_n| - |\mathcal{G}''_n| - \mathcal{B}''_- \end{aligned}$$

and consequently,

$$\begin{aligned} b_{n+1}(m''_{n+1}) - a''_{n+1}(m''_{n+1}) &= |\mathcal{A}''_n| - |\mathcal{A}_n| + |\mathcal{G}''| + \mathcal{B}''_- - \mathcal{B}_- - \mathcal{G}_- \\ &\ge |\mathcal{A}''_n| - |\mathcal{A}_n| + |\mathcal{G}''| + \min(0, |\mathcal{B}| - |\mathcal{B}|') - |\mathcal{G}| \ge 0. \end{aligned}$$

The last inequality holds in case of $\min(|\mathcal{B}|, |\mathcal{B}'|) = |\mathcal{B}'|$ due to condition (v), in case of $\min(|\mathcal{B}|, |\mathcal{B}'|) = |\mathcal{B}|$ due to Eq. 18. Thus also (iv) is valid.

If $R(n+1) = R'(n+1)$, we also have to verify condition (v). In case $R(n+1) = r+1$ there are no $(r+2)$-processes and the condition is fulfilled trivially. On the other hand, i.e, in case $R(n+1) = r$, first note that we can write

$$\mathcal{G}_{n+1} = (\mathcal{G}_n \cup \mathcal{N}) \setminus \mathcal{G}_-$$

with $\mathcal{N} = \mathcal{G}_{n+1} \setminus \mathcal{G}_n$ the set of 'new' $(r-1)$-positions in $A(n+1, r, r+1)$ (i.e., those $(r-1)$-entries which are in columns of processes which switched from r- to $(r+1)$-processes from $A(n)$ to $A(n+1)$). Hence,

$$|\mathcal{G}_{n+1}| = |\mathcal{G}_n| + |\mathcal{N}| - |\mathcal{G}_-| \ge |\mathcal{G}_n| + 0 - |\mathcal{G}_-|.$$

Moreover, note that every arriving message related to $\mathcal{G}_-$ increases the number of $(r+1)$-entries.

Additionally, we can decompose the set $\mathcal{A}_{n+1}$ in the following way:

$$\mathcal{A}_{n+1} = \mathcal{A}_n \cup \mathcal{G}_- \cup \mathcal{B}_-.$$

But, due to our construction of the lower bound, the correspondig equation for $\mathcal{A}''_{n+1}$ is

$$\mathcal{A}''_{n+1} = \mathcal{A}''_n \cup \mathcal{G}''_- \cup \mathcal{B}''_- \cup \mathcal{L}'',$$

where $\mathcal{L}$ are the positions of the new $(r+1)$-entries (instead of r-entries) below the diagonal of new $(r+1)$-processes (see step (3)).

Hence,

$$\begin{aligned} \Delta &:= |\mathcal{A}''_{n+1}| + |\mathcal{G}''_{n+1}| - |\mathcal{A}_{n+1}| - |\mathcal{G}_{n+1}| \\ &= |\mathcal{A}''_n| + |\mathcal{G}''_-| + |\mathcal{B}''_-| + |\mathcal{L}''| + |\mathcal{G}''_n| + |\mathcal{N}''| - |\mathcal{G}''_-| \\ &\qquad -|\mathcal{A}_n| - |\mathcal{G}_-| - |\mathcal{B}_-| - |\mathcal{G}_n| - |\mathcal{N}| + |\mathcal{G}_-| \\ &= |\mathcal{A}''_n| + |\mathcal{B}''_-| + |\mathcal{L}''| + |\mathcal{G}''_n| + |\mathcal{N}''| - |\mathcal{A}_n| - |\mathcal{B}_-| - |\mathcal{G}_n| - |\mathcal{N}|; \end{aligned}$$

due to property b) we can bound the difference $|\mathcal{B}''_-| - |\mathcal{B}_-|$ by $\max(0, |\mathcal{B}''| - |\mathcal{B}|)$ from below, hence

$$\Delta \geq |\mathcal{A}''_n| + |\mathcal{G}''_n| - |\mathcal{A}_n| - |\mathcal{G}_n| + \max(0, |\mathcal{B}''| - |\mathcal{B}|) + |\mathcal{L}''| + |\mathcal{N}''| - |\mathcal{N}|.$$

Observe that the number $|\mathcal{L}''| + |\mathcal{N}''|$ of $(r+1)$- or $(r-1)$-entries below new $(r+1)$-processes equals $\frac{x''_n(x''_n-1)}{2} - \frac{x''_{n+1}(x''_{n+1}-1)}{2}$ (the first term equals the number of entries of r-processes in $A(n)$ below the diagonal, the second term equals the number of entries of r-processes in $A(n+1)$ below the diagonal). Furthermore, the number $|\mathcal{N}|$ can be bounded from above by $(x_n - x_{n+1})x_{n+1}$ (the size of the rectangle below the new $(r+1)$-processes). Moreover, using $x_{n+1} \geq x''_{n+1}$, we obtain

$$\begin{aligned}\mathcal{N} \leq (x_n - x_{n+1})x_{n+1} &\leq \frac{x_n(x_n-1)}{2} - \frac{x_{n+1}(x_{n+1}-1)}{2} \\ &\leq \frac{x_n(x_n-1)}{2} - \frac{x''_{n+1}(x''_{n+1}-1)}{2}.\end{aligned}$$

Consequently,

$$\begin{aligned}\Delta \geq |\mathcal{A}''_n| \;+\; &|\mathcal{G}''_n| - |\mathcal{A}_n| - |\mathcal{G}_n| + \max(0, |\mathcal{B}''| - |\mathcal{B}|) \\ &+ \frac{x''_n(x''_n-1)}{2} - \frac{x''_{n+1}(x''_{n+1}-1)}{2} - \frac{x_n(x_n-1)}{2} + \frac{x''_{n+1}(x''_{n+1}-1)}{2}.\end{aligned}$$

In case $|\mathcal{B}''| > |\mathcal{B}|$ we use Eq. 19 to get

$$\begin{aligned}\Delta &\geq (x_n - x''_n)(N-1) + \frac{x''_n(x''_n-1)}{2} - \frac{x_n(x_n-1)}{2} \\ &= \frac{1}{2}(x_n - x''_n)(2N - x_n - x''_n - 1) \geq 0\end{aligned}$$

(this is clear for $x_n \leq N-1$, and for $x_n = N$ condition (v) is trivially fulfilled). In case $|\mathcal{B}''| \leq |\mathcal{B}|$ we get by using Eq. 19 again

$$\Delta \geq |\mathcal{B}| - |\mathcal{B}''| + (x_n - x''_n)(N-1) + \tfrac{x''_n(x''_n-1)}{2} - \tfrac{x_n(x_n-1)}{2}$$

and by using Eqs. 20 and 21

$$\begin{aligned}\Delta &\geq \frac{(N-x_n)(N-x_n-1)}{2} - |\tilde{\mathcal{A}}| - \frac{(N-x''_n)(N-x''_n-1)}{2} + |\tilde{\mathcal{A}}''| \\ &\qquad +(x_n - x''_n)(N-1) + \frac{x''_n(x''_n-1)}{2} - \frac{x_n(x_n-1)}{2} \\ &= |\tilde{\mathcal{A}}''| - |\tilde{\mathcal{A}}| \geq 0.\end{aligned}$$

The last inequality holds due to (vi). So property (v) is proven.

It remains to show (vi). Property c) of the bijection ensures that in $A(n)$ at most $\max(|\tilde{\mathcal{B}}| - |\tilde{\mathcal{B}}''|, 0)$ more $(r+1)$-messages above the diagonal than in $A''(n)$ can arrive. This yields

$$\begin{aligned}&|\tilde{\mathcal{A}}''_{n+1}| - |\tilde{\mathcal{A}}_{n+1}| \geq |\tilde{\mathcal{A}}''_n| - |\tilde{\mathcal{A}}_n| - \max(|\tilde{\mathcal{B}}| - |\tilde{\mathcal{B}}''|, 0) \\ &= \begin{cases} |\tilde{\mathcal{A}}''_n| - |\tilde{\mathcal{A}}_n| \geq 0 & \text{if } |\tilde{\mathcal{B}}| \leq |\tilde{\mathcal{B}}''| \\ |\tilde{\mathcal{A}}''_n| + |\tilde{\mathcal{B}}''| - |\tilde{\mathcal{A}}_n| - |\tilde{\mathcal{B}}| = \frac{(N-x''_n)(N-x''_n-1)}{2} - \frac{(N-x_n)(N-x_n-1)}{2} \geq 0 & \text{ow.} \end{cases}\end{aligned}$$

So the proof is finished in Case A.

Case B $r := R(n) = R''(n) - 1$**:** We will use the sets $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{F}, \mathcal{G}$ (as defined in the proof of the upper bound) and define the sets $\mathcal{A}'', \mathcal{B}'', \mathcal{H}'', \mathcal{I}'', \mathcal{K}'', \mathcal{J}''$ analogously to those in the proof of the upper bound.

Again, denote with $M'' = \mathcal{I}'' \cup \mathcal{B}'' \cup \mathcal{J}''$ and $M = \mathcal{B} \cup \mathcal{D} \cup \mathcal{G}$ the sets of relevant channels. Choose any bijection $h\colon [N]^2 \to [N]^2$, that keeps the diagonal fixed, with

$$\begin{cases} M'' \subseteq h(M) & \text{if } |M| \geq |M''| \\ h(M) \subseteq M'' & \text{if } |M''| \geq |M| \end{cases} \tag{22}$$

and

$$\begin{cases} \mathcal{B} \cup \mathcal{G} \subseteq h(\mathcal{B}'' \cup \mathcal{J}'') & \text{if } |\mathcal{B}| + |\mathcal{G}| \leq |\mathcal{B}''| + |\mathcal{J}''| \\ h(\mathcal{B}'' \cup \mathcal{J}'') \subseteq \mathcal{B} \cup \mathcal{G} & \text{if } |\mathcal{B}| + |\mathcal{G}| > |\mathcal{B}''| + |\mathcal{J}''|. \end{cases} \tag{23}$$

Define $g_{E_{\leq n}}$ as the bijection on $\mathcal{M}$ induced by h. Then conditions (i) and (ii) for g_{n+1} defined in Eq. 17 are obviously fulfilled. The fact $R(n+1) \leq R(n)+1$ immediately implies (iii).

Let us we verify (iv). If $m''_{n+1} = r+1$ (i.e., a global round switch in $A''(n)$ occured), then $m_{n+1} \geq \max A(n+1)$ and we are done. Otherwise, if $m''_{n+1} = r$, observe that $m''_n = r$, that $\mathcal{B}''$ and $\mathcal{J}''$ contain all relevant r-entries of $A''(n)$ and $\mathcal{B}$ and $\mathcal{G}$ contain all those relevant entries in $A(n)$ which can be overwritten by $(r+1)$. Hence the change in a''_n and b_n can be bounded, due to property (23), by

$$\begin{aligned} & b_{n+1}\,(m''_{n+1}) - a''_{n+1}(m''_{n+1}) \\ & \quad \geq b_n(m''_n) - a''_n(m''_n) - \max(|\mathcal{B}| + |\mathcal{G}| - |\mathcal{B}''| - |\mathcal{J}''|, 0) \\ & \quad = |\mathcal{B}| + |\mathcal{C}| + |\mathcal{D}| + |\mathcal{F}| + |\mathcal{G}| - |\mathcal{B}''| - |\mathcal{J}''| - \max(|\mathcal{B}| + |\mathcal{G}| - |\mathcal{B}''| - |\mathcal{J}''|, 0) \\ & \quad = \begin{cases} |\mathcal{B}| + |\mathcal{C}| + |\mathcal{D}| + |\mathcal{F}| + |\mathcal{G}| - |\mathcal{B}''| - |\mathcal{J}''| \geq 0 & \text{if } |\mathcal{B}| + |\mathcal{G}| \leq |\mathcal{B}''| + |\mathcal{J}''| \\ |\mathcal{C}| + |\mathcal{D}| + |\mathcal{F}| \geq 0 & \text{if } |\mathcal{B}| + |\mathcal{G}| > |\mathcal{B}''| + |\mathcal{J}''|, \end{cases} \end{aligned}$$

and so (iv) is proven.

To finish the proof in Case B, it remains to check conditions (v) and (vi) in case of $R(n+1) = R''(n+1) = r+1$, i.e., a global round switch occured in $A(n)$, but then there are no $(r+2)$-processes in $A(n+1)$ and so (v) and (vi) are trivially fulfilled.

Case C $R(n) + 1 < R''(n) + 1$**:** Here we define $g_{E_{\leq n}} := \mathrm{id}_{\mathcal{M}}$. The conditions (i) and (ii) are clear, and we can ignore (v) and (vi). From $R(n) + 1 < R''(n)$ we obtain $R(n+1) < R(n+1)''$ and thus (iii). Moreover, we have $\max A(n) \leq \min A''(n)$ and hence $A(n+1)_{i,j} \leq A''(n+1)_{i,j}$ for all $1 \leq i, j \leq N$; this implies (iv). □

5 Discussion and Future Work

In the previous sections, we constructed an upper and a lower bound for the α-synchronizer and proved several properties of these approximations. In this section, we will complement our analytical findings by some simulation results.

Based on our constructions of the Markov chains in Sections 3 and 4, the upper and lower bound can be directly determined by solving the linear equation system for the steady state distribution π of the respective Markov chain and by computing the expected round round duration from π using Füger et al. (2015, Theorem 5). Algorithm 1 and 2 provide the details of these computations.

Algorithm 1 Upper bound for expected round duration.

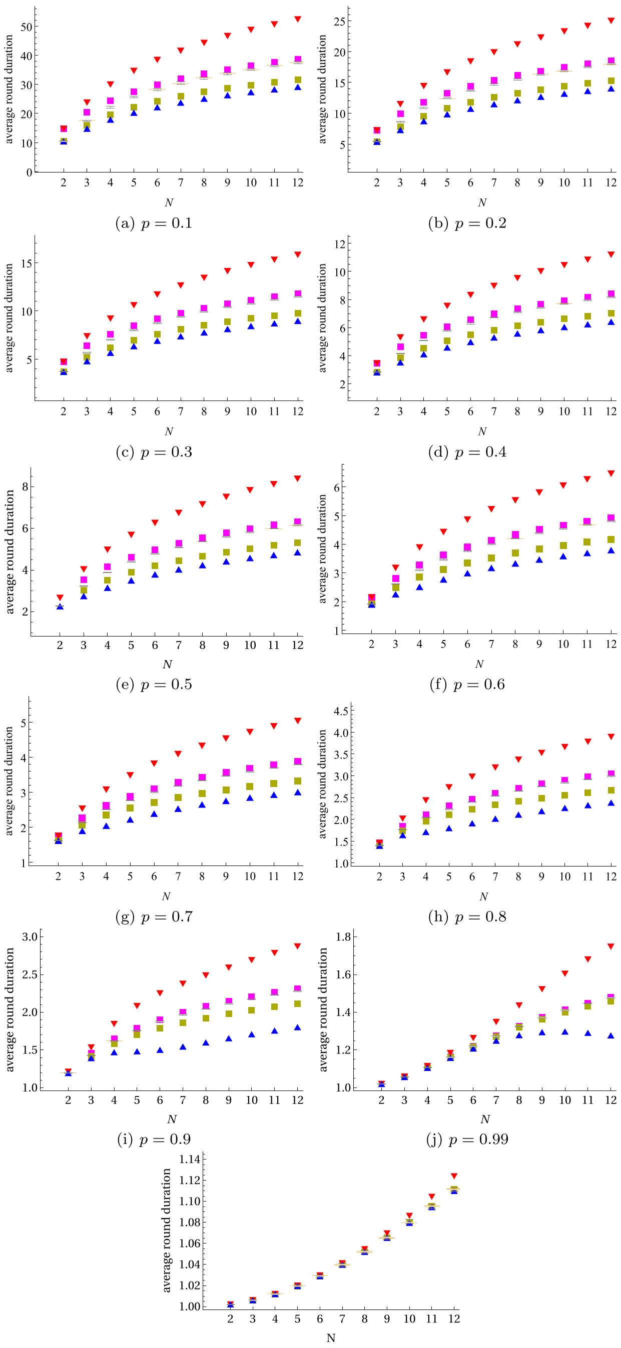
```
input : $N$ (number of processes), $p$ (probability of successful transmission)
output: upper bound for expected round duration
generate states $\sigma = (x, a)$ of the Markov chain according to Eq. 2
generate transmission probability matrix $P$ according to Eqs. 3a–3e and 4a–4c
substract from every diagonal entry in $P$ 1
replace every entry of the last column of $P$ by 1
solve linear equation system $P^T \pi = e$ with $e = (0, \dots, 0, 1)^T$ to obtain the steady state distribution $\pi$
compute $L = \sum_\sigma \pi(\sigma) \cdot p^{f(\sigma)}$ with $f(N, a) = N(N-1) - a$ and $f(x, a) = x$ otherwise
compute expected round duration $\lambda = 1/L$
```

Algorithm 2 Lower bound for expected round duration.

```
input : $N$ (number of processes), $p$ (probability of successful transmission)
output: lower bound for expected round duration
generate states $\sigma = (x, y, z)$ of the Markov chain according to Eq. 14
generate transmission probability matrix $P$ according to Eqs. 15a–15c and 16
substract from every diagonal entry in $P$ 1
replace every entry of the last column of $P$ by 1
solve linear equation system $P^T \pi = e$ with $e = (0, \dots, 0, 1)^T$ to obtain the steady state distribution $\pi$
compute $L = \sum_\sigma \pi(\sigma) \cdot p^{x(x-1)-z}$
compute expected round duration $\lambda = 1/L$
```

In Figs. 4 (for p fixed) and Fig. 5 (for N fixed), we compare simulations for the α-synchronizer against the exact values of the bounds given in Függer et al. (2015) and of our new bounds. For the simulations, we performed 30 runs with 100000 time steps each. The results for the α-synchronizer are represented as (very thin) box-whisker-charts in Fig 4, whereas in Fig. 5 we only show the average. Note that the bounds of Függer et al. (2015) are represented by triangles, whereas our bounds are represented by squares. Noticeable is the very good approximation by our new upper bound, which confirms the analytic dominance results obtained in Section 3.5.

Our bounds are indeed an improvement of the results of Függer et al. (2015), since we reduced the state-space of the Markov chains from exponential to polynomial size and since our bounds (especially the upper bound) are much better. Nevertheless, we don't have closed formulas and calculating and solving the Markov chains of our bounds is still expensive: We could solve it algebraically only up to $n = 4$ (whereas for the α-synchronizer we could do it only for $n = 3$ and were not even able to generate the state space for $n = 4$). Numercial solutions are computable also for higher n: the upper bound we could solve for $n = 12$ in about 12 seconds, for $n = 20$ it took us about five minutes. The lower bound is more time-consuming: For $n = 10$ it took us about half a minute, whereas for $n = 12$ the solution was computed in 2,2 minutes. To be more precise, the size of the state spaces of the Markov chain and time and memory consumption of our algorithms computing the bounds are summarized in the following table.



(a) $p = 0.1$

(b) $p = 0.2$

(c) $p = 0.3$

(d) $p = 0.4$

(e) $p = 0.5$

(f) $p = 0.6$

(g) $p = 0.7$

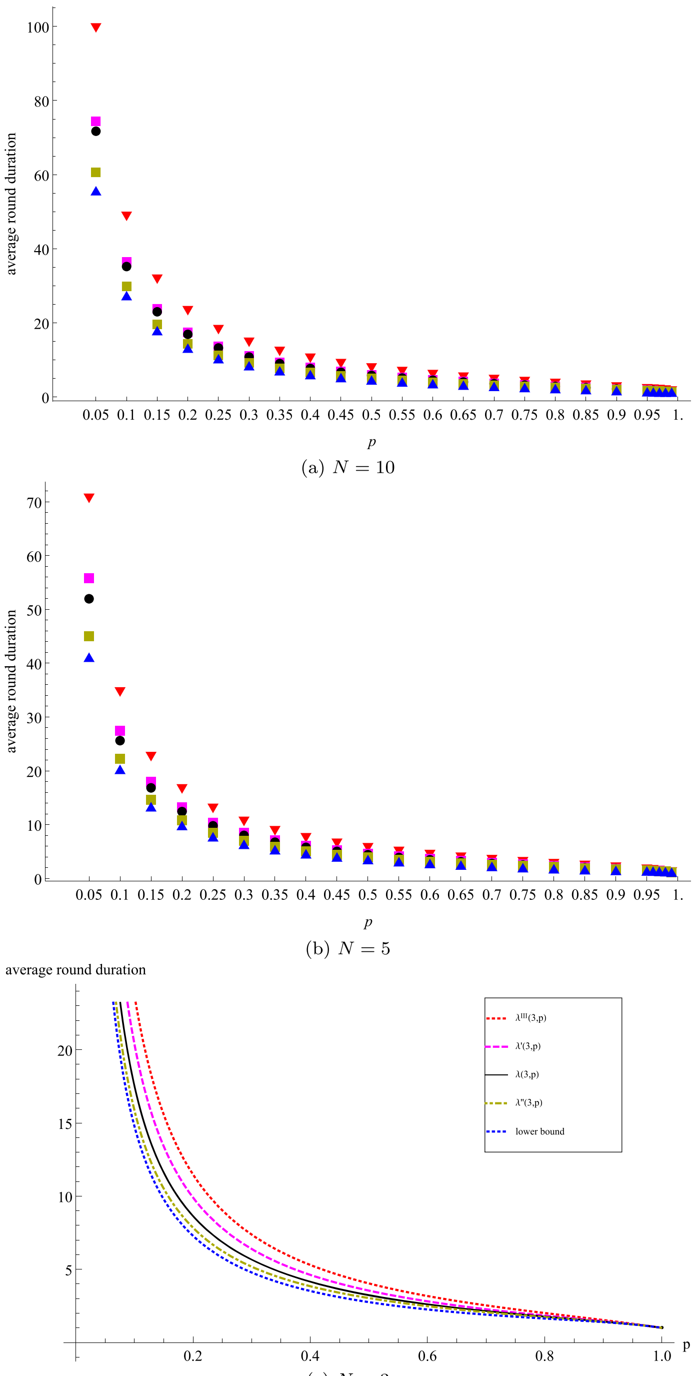
(h) $p = 0.8$

(i) $p = 0.9$

(j) $p = 0.99$

(k) $p = 0.999$

Fig. 4 Monte-Carlo simulation results for the α-synchronizer (represented as boxplots) compared against the lower (▲) and upper (▼) bound given in Függer et al. (2015), our new upper bound ■, and our new lower bound ■. As usual, the (orange) box of the boxplots represents the interval from the 25% to the 75% quantile, whereas the fences mark the minimum and the maximum



(a) $N = 10$

(b) $N = 5$

(c) $N = 3$

Fig. 5 Monte-Carlo simulation results (for $N = 5$ and $N = 10$ the means represented as ●) and exact round duration (for $N = 3$) for the α-synchronizer compared against the lower (▲) and upper (▼) bound given in Függer et al. (2015), our new upper bound ■, and our new lower bound ■

	Upper bound	Lower bound
Number of states	$N(N^2-1)/2$	$(N^4-N^3+5N^2-11N+12)/6$
Memory	$\Theta(N^6)$	$\Theta(N^8)$
Time	$\mathrm{O}(N^9)$	$\mathrm{O}(N^{12})$

For example, for $n = 12$ the Markov chain describing the upper bounds has 858 states, whereas the Markov chain for the lower bound has 3268 states.

So future challenges are to find bounds which are also easily computable for higher values of n and which are still of good quality. Furthermore, we plan to investigate the asymptotic behavior for large n. The present paper reveals that this is not a priori hopeless, in the sense that the improved understanding of the underlying Markov chain already allowed us to reduce the exponential state space to polynomial and to compute considerably improved upper and lower bounds. It may well be the case that the additional insights gained in the analysis of the modified synchronizer algorithms will eventually allow us to come up with computationally better tractable bounds.

Moreover, in Függer et al. (2015) the authors considered variants of the α-synchronizer by allowing processes to actively forget parts of their accumulated knowledge. It is also part of our future work to adapt the ideas of the upper and the lower bound of the α-synchronizer to construct bounds for their case II (when processes forget their knowledge at a local round switch).

6 Glossary of Notation

$R_i(t)$	local round number of process i	p. 4
$R(t)$	global round number	p. 4
$K_{i,j}(t)$	knowledge of process i from process j	p. 4
$\mathcal{P}$	probability matrix of successful transmission	p. 4
$\lambda = \lambda(N, p)$	expected round duration	p. 5
$A(t), A^{(i)}(t)$	matrix of the Markov Chain (after computation step (i))	p. 6
$a(t), a^{(i)}(t)$	matrix of the normalized Markov Chain	p. 6
$md^{(i)}(t)$	minimum of the diagonal of $A^{(i)}(t)$	p. 6
$w^{(i)}(t)$	number of $md^{(i)}(t)$-entries in the diagonal of $A^{(i)}(t)$	p. 6
$\Delta(x) =$	$x_1 - x_2$	p. 9
$\nu(x) =$	$(N-1)(N-x)$	p. 9
$\nu =$	$N(N-1)$	p. 9
$\mathcal{M}$	set of messagepatterns, $\cong \{0,1\}^{N^2-N}$	p. 12
$\mathcal{E}$	set of schedules, $= \mathcal{M}^{\mathbb{N}}$	p. 12
$\mathcal{A}_n =$	{positions of non-diagonal $(r+1)$-entries in $A(n)$}	p. 13
$\mathcal{G}_n =$	{positions of $(r-1)$-entries in $A(n, r, r+1)$}	p. 13
$\mathcal{B}_n =$	{positions of r-entries in $A(n, r+1, r+1)$ and $A(n, r, r+1)$}	p. 13
$\mathcal{F}_n =$	{positions of r-entries in $A(n, r+1, r)$}	p. 13
$\mathcal{C}_n =$	{positions of non-diagonal r-entries in $A(n, r, r)$}	p. 13
$\mathcal{D}_n =$	{positions of $(r-1)$-entries in $A(n, r, r)$}	p. 13

$\mathcal{H}_n =$	{positions of non-diagonal $(r+2)$-entries in $A(n)$}	p. 16
$\mathcal{I}_n =$	{positions of $(r+1)$-entries in $A(n, r+2, r+2)$ and $A(n, r+1, r+2)$}	p. 16
$\mathcal{K}_n =$	{positions of $(r+1)$-entries in $A(n, r+2, r+1)$}	p. 16
$\mathcal{J}_n =$	{positions of r-entries in $A(n, r+1, r+2)$}	p. 16
$n(x) =$	$(N-x)(N-x-1)$	p. 23
$\mu(x) =$	$(N-1)x$	p. 23

Funding Information Open access funding provided by Austrian Science Fund (FWF).

References

Awerbuch B (1985) Complexity of network synchronization. J ACM 32(4):804–823

Bertsekas DP, Tsitsiklis JN (1989) Parallel and distributed computation. Prentice Hall, Englewood Cliffs

Bettstetter C, Hartmann C (2005) Connectivity of wireless multihop networks in a shadow fading environment. Wirel Netw 11(5):571–579. https://doi.org/10.1007/s11276-005-3513-x

Cerpa A, Wong JL, Kuang L, Potkonjak M, Estrin D (2005a) Statistical model of lossy links in wireless sensor networks. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, IPSN '05. IEEE Press, Piscataway. http://dl.acm.org/citation.cfm?id=1147685.1147701

Cerpa A, Wong JL, Potkonjak M, Estrin D (2005b) Temporal properties of low power wireless links: Modeling and implications on multi-hop routing. In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '05, pp. 414–425. ACM, New York. https://doi.org/10.1145/1062689.1062741

Clementi AEF, Pasquale F, Monti A, Silvestri R (2007) Communication in dynamic radio networks. In: Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing, PODC '07, pp. 205–214. ACM, New York. https://doi.org/10.1145/1281100.1281131

Clementi AE, Macci C, Monti A, Pasquale F, Silvestri R (2008) Flooding time in edge-markovian dynamic graphs. In: Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing, PODC '08, pp. 213–222. ACM, New York. https://doi.org/10.1145/1400751.1400781, USA

Dousse O, Baccelli F, Thiran P (2005) Impact of interferences on connectivity in ad hoc networks. IEEE/ACM Trans Netw 13(2):425–436. https://doi.org/10.1109/TNET.2005.845546

Függer M, Kößler A, Nowak T, Schmid U, Zeiner M (2015) The effect of forgetting on the performance of a synchronizer. Perform Eval **93**, 1–16. https://doi.org/10.1016/j.peva.2015.08.002. http://www.sciencedirect.com/science/article/pii/S0166531615000826

Fussen M, Wattenhofer R, Zollinger A (2005) Interference arises at the receiver. In: 2005 international conference on Wireless networks, communications and mobile computing, vol. 1, pp. 427–432. https://doi.org/10.1109/WIRLES.2005.1549447

Gupta P, Kumar PR (2000) The capacity of wireless networks. IEEE Trans Inf Theory 46(2):388–404. https://doi.org/10.1109/18.825799

Moscibroda T, Wattenhofer R (2006) The complexity of connectivity in wireless networks. In: Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, pp. 1–13. https://doi.org/10.1109/INFOCOM.2006.23

Nowak T, Függer M, Kößler A (2013) On the performance of a retransmission-based synchronizer. Theor Comput Sci **509**, 25–39

Propp JG, Wilson DB (1996) Exact sampling with coupled markov chains and applications to statistical mechanics. Random Struct Algorithm 9(1-2):223–252

Rajsbaum S (1994a) Upper and lower bounds for stochastic marked graphs. Inf Process Lett 49(6):291–295

Rajsbaum S, Sidi M (1994b) On the performance of synchronized programs in distributed networks with random processing times and transmission delays. IEEE Trans Parallel Distrib Syst 5(9):939–950

Schilcher U, Bettstetter C, Brandner G (2012) Temporal correlation of interference in wireless networks with rayleigh block fading. IEEE Trans Mob Comput 11(12):2109–2120

Schilcher U, Toumpis S, Haenggi M, Crismani A, Brandner G, Bettstetter C (2016) Interference functionals in poisson networks. IEEE Trans Inf Theory 62(1):370–383. https://doi.org/10.1109/TIT.2015.2501799

Ware C, Judge J, Chicharo J, Dutkiewicz E (2000) Unfairness and capture behaviour in 802.11 adhoc networks. In: 2000 IEEE International conference on communications. ICC 2000. Global convergence through communications