Performance Evaluation 93 (2015) 1–16

Contents lists available at ScienceDirect

Performance Evaluation

journal homepage: www.elsevier.com/locate/peva

The effect of forgetting on the performance of a synchronizer[☆]

Matthias Függer[a], Alexander Kößler[a], Thomas Nowak[b], Ulrich Schmid[a], Martin Zeiner[a,*]

[a] TU Wien, Treitlstraße 3/II, 1040 Wien, Austria

[b] ENS Paris, 45 rue d'Ulm, 75230 Paris Cedex 05, France

ARTICLE INFO

Article history:
Received 11 November 2014
Received in revised form 24 June 2015
Accepted 6 August 2015
Available online 15 August 2015

Keywords:
Distributed systems
Synchronizer
Performance analysis
Probabilistic message loss

ABSTRACT

We study variants of the α-synchronizer by Awerbuch (1985) within a distributed message passing system with probabilistic message loss. The purpose of a synchronizer is to maintain a virtual (lock-step) round structure, which simplifies the design of higher-level distributed algorithms. The underlying idea of an 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.

In this work, we study the performance of several synchronizers in an environment with probabilistic message loss. In particular, we analyze how different strategies of forgetting affect the round durations. The synchronizer variants considered differ in the times when processes discard part of their accumulated knowledge during the execution. Possible applications can be found, e.g., in sensor fusion, where sensor data become outdated and thus invalid after a certain amount of time.

For all synchronizer variants considered, we develop corresponding Markov chain models and quantify the performance degradation using both analytic approaches and Monte-Carlo simulations. Our results allow to explicitly calculate the asymptotic behavior of the round durations: While in systems with very reliable communication the effect of forgetting is negligible, the effect is more profound in systems with less reliable communication. Our study thus provides computationally efficient bounds on the performance of the (non-forgetting) α-synchronizer and allows to quantitatively assess the effect accumulated knowledge has on the performance.

1. Introduction

It is well-known that reliable communication can be simulated in asynchronous distributed systems with unreliable links using retransmissions [1], even if up to a minority of the nodes may crash, provided links are fair-lossy, i.e., re-sending a message infinitely often causes it to eventually be received. Once reliable links are available, a *synchronizer* like the α-synchronizer, introduced by Awerbuch [2] as the first in a series of synchronizer algorithms for asynchronous failure-free message-passing systems, can be used to simulate a synchronous system atop of it. The computation of a synchronous system evolves in a sequence of *rounds*, conceptually executed in lock-step, where all nodes (called processes in the sequel)

☆ This research was partially supported by the Austrian Science Fund (FWF), grants NFN RiSE (S11405), PSRTS (P20529), and SIC (P26436).

* Corresponding author. Tel.: +43 15880118266.

E-mail addresses: fuegger@ecs.tuwien.ac.at (M. Függer), koe@ecs.tuwien.ac.at (A. Kößler), thomas.nowak@ens.fr (T. Nowak), s@ecs.tuwien.ac.at (U. Schmid), mzeiner@ecs.tuwien.ac.at (M. Zeiner).

http://dx.doi.org/10.1016/j.peva.2015.08.002

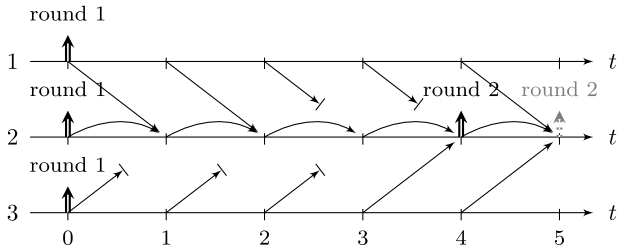


Fig. 1. Messages to process 2 and its resulting round switches without forgetting (black) and with forgetting (gray).

exchange messages that are processed at the end of the round. Obviously, the performance of the simulated synchronous distributed system is entirely determined by the achieved round durations.

The α-synchronizer's main idea is to let a process continuously broadcast its current round number together with the corresponding application data. The next round is started when a process has received the messages of its current round from all other processes; it also delivers the data received in the current round to the application layer on that occasion. Note that the original α-synchronizer by Awerbuch uses additional acknowledgment messages, which we omit. Rather, a message with round number R is treated as an implicit acknowledge for messages with round numbers less than R in our setting.

A crucial characteristics of a synchronizer algorithm is its *precision*, i.e., the maximum number of rounds any two processes can be apart at the same time during any of its executions. The α-synchronizer, and the variants considered in this paper, guarantee a precision equal to 1, provided every process can communicate with every other process. Note that fault-tolerant synchronizers, like the ones described in [3] (which can tolerate even Byzantine, i.e., arbitrarily, faulty processes), usually guarantee a precision >1 only. A synchronizer with precision 1 constructs rounds, which have a duration equal to the (maximum) end-to-end delay of the messages exchanged over the (simulated) reliable link during the round.

Unfortunately, just knowing that re-sending a message infinitely often causes it to be received eventually over a fair-lossy link does not guarantee a bounded end-to-end delay of a simulated reliable link. Stronger models for the underlying unreliable links are hence required in order to assess the round durations and thus the performance of the simulated synchronous system.

In this paper,[1] we consider a system of N non-faulty processes that are pairwise connected by unreliable links that may lose messages probabilistically. Time elapses in discrete steps, simultaneously at all processes in the system. At every time step, every process sends a message to every other process. With some fixed probability p, such a message is successfully delivered before the next step; with probability $1-p$, it is lost.[2]

Fig. 1 shows the beginning of an execution of the α-synchronizer executed in a system with three processes in our setting. For better clarity, only messages to process 2 are shown. Initially, at time 0, all processes start round 1. By time 4, process 2 has received round 1 messages from all processes and thus proceeds to round 2.

It is apparent, though, that the age of the round 1 data handed over to the application layer when switching to round 2 differs significantly per process: while its own data and the data from process 3 is of age 1 (discrete time units), data from process 1 is of age 3. If this data is time-variant, e.g., the position of a moving object, this may cause problems: Different processes may receive different data from the same sensor in the same round. For example, time-dependent sensor data are often represented by an interval (i.e., a value $\pm$ some accuracy) that deteriorates with time [5]. A proper deterioration accounts for the maximum change of the position since the actual sampling of the data. When merging intervals representing the same data, from different sources, e.g. using (fault-tolerant) interval-intersection functions like [6,7], relying on old data obviously yields less accurate results.

A strategy to counteract this problem is to let the synchronizer actively "forget" too old data, by just discarding it. As an extreme, consider a variant of the synchronizer that discards data at each (discrete) time step, resulting in all the data to be of age 1 at each round switch. Clearly, however, this results in a performance loss, i.e., longer round durations: The resulting execution is depicted in gray in Fig. 1, with the difference that process 2 then switches to round 2 only at time 5. Possible applications of active forgetting could be found in sensor fusion [5], where time-dependent sensor data that is sampled periodically may also be (detectably) erroneous, e.g., due to measurement or communication errors: If the probability $1-p$ of getting erroneous data is large, it may be beneficial in terms of the accuracy of the fused data to discard data received from some other sensor long ago.

Detailed contributions. In this paper, we consider four variants of the α-synchronizer that differ in the conditions of when to forget memory content, that is, reset the variables representing the knowledge to their initial values. While three of the variants, namely the variant I that never forgets ($=$ the original α-synchronizer), the variant II in which a process forgets

[1] A preliminary version of this paper appeared in [4].

[2] We thus assume the existence of an underlying mechanism that prevents the processes' discrete time from diverging, i.e., a synchronous system underneath.

when it makes a round switch, and the variant IV that forgets at each time step, can be implemented in a distributed manner, the variant III which forgets when the last process makes a round switch serves as a theoretical bound of low computational complexity for variants I and II only.

Thanks to the independence of the message loss at every link at every time step, the behavior of our synchronizers is governed by relatively simple stochastic processes, namely, finite Markov chains. Owing to the considerable amount of state information gathered during a round in some variants (in particular, in variant I), however, the state space may be huge. For a system of N processes and probability p of successful transmission, we define the *expected round duration* of process i as $\lambda_i(N,p) = \mathbb{E}\lim_{t\to\infty} t/R_i(t)$, where $R_i(t)$ is the round number of process i at time t. Since the synchronization algorithm guarantees precision 1 regardless of the forgetting strategy, it immediately 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 $\lambda(N,p)$. To distinguish the four proposed conditions on forgetting, I–IV, we write λ^{I}, λ^{II}, λ^{III}, and λ^{IV}, respectively. We will see that indeed $\lambda^{\mathrm{I}} \leqslant \lambda^{\mathrm{II}} \leqslant \lambda^{\mathrm{III}} \leqslant \lambda^{\mathrm{IV}}$.

By giving explicit formulas and simulation results for the performance as well as simulation results for the average age of data when a process makes a round switch, our results can be used to quantify the tradeoff between the different strategies.

For cases III and IV, we obtain two computationally feasible formulas where λ^{III} and λ^{IV} can be computed with $O(N^2)$ and $O(N)$ arithmetic operations, respectively:

Theorem 1. *The expected round duration when forgetting at every global round switch reads*

$$\lambda^{\mathrm{III}}(N,p) = \sum_{i=1}^{N(N-1)} \binom{N(N-1)}{i} \frac{(-1)^i}{(1-p)^i - 1},$$

and when always forgetting,

$$\lambda^{\mathrm{IV}}(N,p) = \sum_{i=1}^{N} \binom{N}{i} (-1)^i \frac{1}{(1-p^{N-1})^i - 1}.$$

For the remaining cases I and II, we state a finite Markov chain that captures the synchronizer dynamics. Unfortunately, explicit calculation of λ^{I} and λ^{II} from the chain's steady state is prohibitively costly in terms of time and space complexity.

However, the Markovian representation allows us to derive two asymptotic results for synchronizer systems with near to no message loss ($p \to 1$) and very high loss ($p \to 0$):

Theorem 2. *For all four conditions on forgetting,* $\frac{d}{dp}\lambda(N,p)\big|_{p=1} = -N(N-1)$.

Theorem 3. *For* $p \to 0$,

1. $\lambda^{\mathrm{I}}(N,p)$, $\lambda^{\mathrm{II}}(N,p)$ *and* $\lambda^{\mathrm{III}}(N,p)$ *are in* $\Theta\left(p^{-1}\right)$, *and*
2. $\lambda^{\mathrm{IV}}(N,p)$ *is in* $\Theta\left(p^{-(N-1)}\right)$.

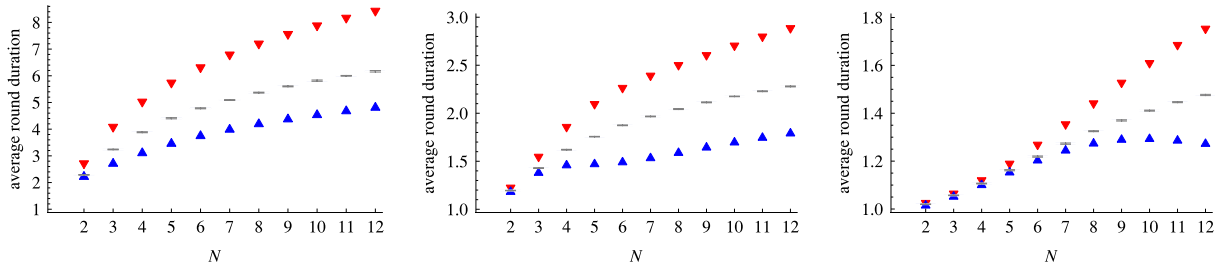
Fig. 6 shows, with probability p varying in the unit interval [0, 1], the calculated exact value of the expected round duration for conditions on forgetting I–IV in a system with $N = 3$ processes. The figure shows the gap between the cases I, II, and III, having an asymptotic growth in $\Theta(1/p)$ when p approaches 0, and the case IV, which has an asymptotic growth in $\Theta(1/p^{N-1})$. Furthermore, all the plots have the same slope in the point $p = 1$: the efficiently computable cases III and IV thus provide good approximations for the hard to calculate cases I and II in a system with reliable communication.

In settings with unreliable communication, for which the approximation result on the derivative of λ at $p = 1$ is not discriminating, cases I and II can be approximated by analytic lower bounds (cf. Section 5.4), and bounded from above by the λ for case III (Theorem 1). A comparison between the lower bounds and the actual results is also contained in Fig. 6.

As the calculations of the exact values for the expected round duration using the Markov chain model are computationally very expensive, we used Monte-Carlo simulations to compare them with our bounds. To this end, we simulated systems with $2 \leqslant N \leqslant 12$ processes for 100 000 steps and averaged over 30 runs. The simulations were done using three different values for p. Figs. 2 and 3 show box-and-whisker-plots of the simulated average round durations with the calculated lower bound and with case III as upper bound. The average round durations for case I (where processes never forget) is shown in Fig. 2(a)–(c) and the case II (where processes forget after a local round switch) is shown in Fig. 3(a)–(c). Fig. 4(a)–(c) depicts the calculated expected round duration for case IV, i.e., the synchronizer variant that forgets at each time step. Note that it is significantly higher than all the other variants when message loss is considerable.

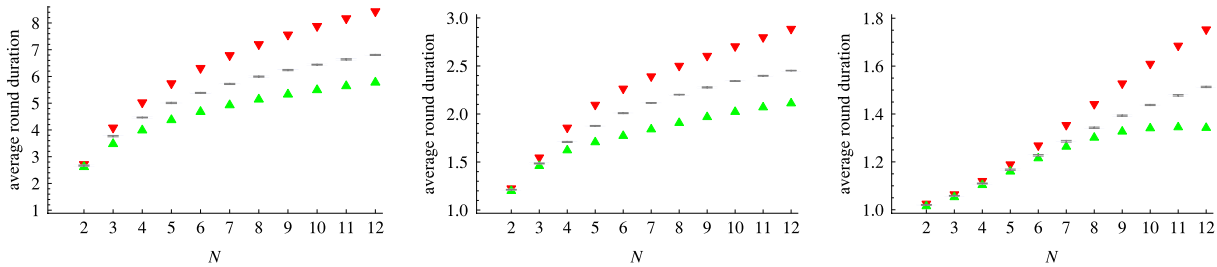
Fig. 5 shows box-and-whisker-plots of the Monte-Carlo simulation results of the average age of data when a process performs a round switch, for cases I and II, both of which can be implemented in a distributed manner. Case IV, for which the same holds, by definition has an average age of data of 1. One immediately observes that while the average age of both cases I and II is significantly higher than in case IV, forgetting at each processes' round switch only has a marginal effect on the average age compared to not forgetting at all.

Related work. In distributed computing, notions of a process' knowledge have been explicitly introduced before, see e.g., Halpern [8] and Fagin et al. [9]: A process knows a fact if in all locally indistinguishable scenarios the fact also holds.



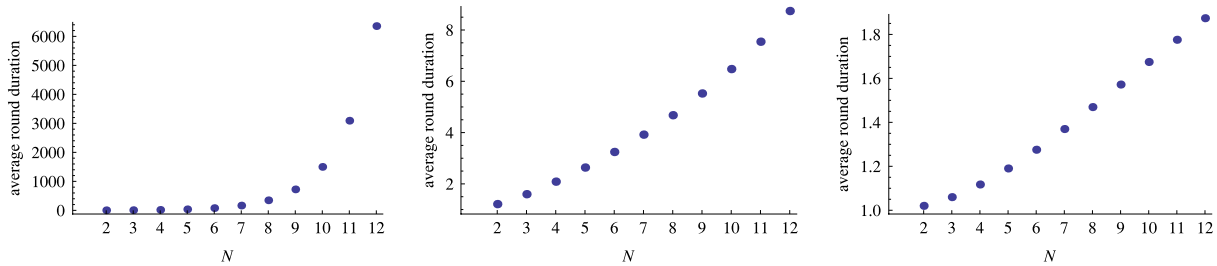
(a) $p = 0.5$. (b) $p = 0.9$. (c) $p = 0.99$.

Fig. 2. Monte-Carlo simulation results for case I (represented as boxplots) compared against the calculated lower bound (▲) and the calculated expected round duration of case III serving as an upper bound (▼).



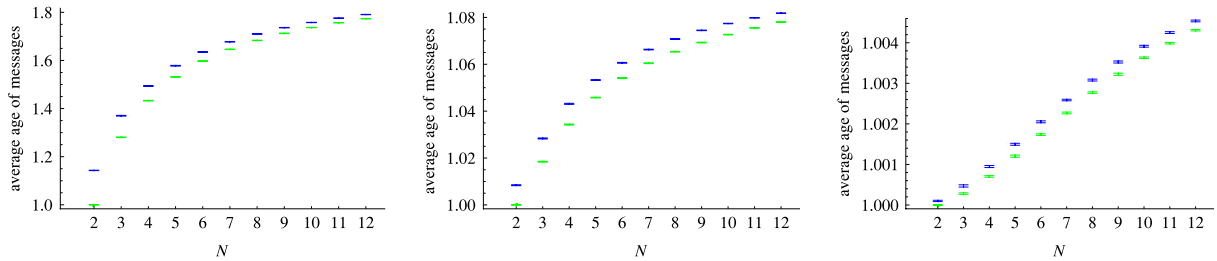
(a) $p = 0.5$. (b) $p = 0.9$. (c) $p = 0.99$.

Fig. 3. Monte-Carlo simulation results for case II (represented as boxplots) compared against the calculated lower bound (▲) and the calculated expected round duration of case III serving as an upper bound (▼).



(a) $p = 0.5$. (b) $p = 0.9$. (c) $p = 0.99$.

Fig. 4. Calculated expected round duration for case IV.



(a) $p = 0.5$. (b) $p = 0.9$. (c) $p = 0.99$.

Fig. 5. Boxplots of the average age Monte-Carlo simulation results for case I (blue, upper) and II (green, lower).

Studying the evolution of knowledge has been a promising tool in analyzing agreement protocols, whose purpose is to establish common knowledge on an agreed upon value. The approach was further generalized by Fagin and Halpern [10] to include probabilistic facts by assigning probabilities to scenarios. Our notion of knowledge, however, is different from these notions of knowledge, particularly since it is tailored to the synchronizer problem statement we study in this paper, and we allow processes to actively forget.

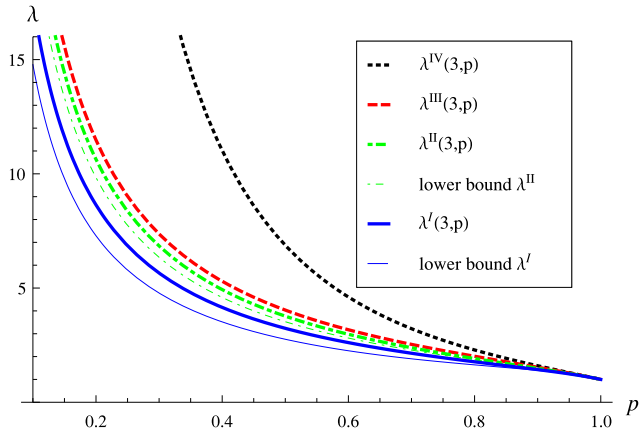


Fig. 6. Expected round durations for $N = 3$ and lower bounds for cases I and II.

Mahesh and Varghese [11] have used forgetting in a destructive way to model crashed processes without permanent storage after rejoining the network. By contrast, we allow processes to explicitly trigger forgetting themselves, using it in a constructive algorithmic manner. The closest relation to our active forgetting strategies bears the work on self-stabilizing distributed systems, see e.g. [12]. Since self-stabilizing protocols are required to recover from arbitrary initial states, processes typically discard status flags and message in-buffers after certain time-outs to prevent deadlocks. While these protocols also use forgetting in a constructive way, purpose and policies differ: Self-stabilizing protocols clean up memory with periods depending on system parameters such as message delay bounds to prevent deadlocks. By contrast, we assume no external corruption of a process i's state, including its round counter R_i. Processes may thus rely on their local states, and forgetting strategies only influence timing performance rather than correctness.

Synchronizer performance has been studied under different environmental assumptions in the past. Bertsekas and Tsitsiklis [13] proved performance bounds for the case of constant processing times and exponentially distributed message delays. In their work, the authors assumed reliable links without message loss. Rajsbaum [14] presented bounds on the synchronizer performance for the case of exponentially distributed processing times and transmission delays, still assuming reliable messages. Rajsbaum and Sidi [15] calculated the exact 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 already been considered by Nowak et al. [16]. The authors calculated the expected round duration of a retransmission-based synchronizer when a single transmission arrives with constant probability p, subject to the constraint that a message that was retransmitted at least M times is guaranteed to arrive. [16] did not investigate the impact of forgetting on the synchronizer's performance, and assumed M to be finite, which we do not.

For the computationally difficult forgetting variants I and II, we present finite Markov chains that allow to determine average round durations λ^{I} and λ^{II} from the steady states. Hereby, the dominant computational complexity 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 [17] 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.

2. System model and algorithm

We consider a fully-connected message passing system with processes $1, 2, \ldots, N$ each of which runs a synchronizer algorithm. Processes take steps simultaneously at all integral times $t \geqslant 0$, but messages may be lost. 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 variants have 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 continuously broadcast their local round number. 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, after it has gained knowledge that all other processes have already started the current round. The round increment rule assures a precision of 1, i.e., $|R_i(t) - R_j(t)| \leqslant 1$ for all t. We write $R_G(t) = \min_i R_i(t)$ and call it the *global round number* at time t.

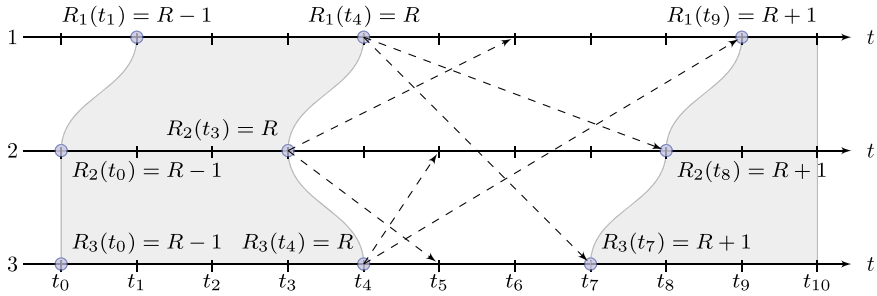


Fig. 7. An execution of the synchronizer.

After updating its local round number, a process may *forget*, i.e., lose its knowledge about other processes' local round numbers. We are considering four different conditions COND, describing the times when process i forgets:

I. Never, i.e., COND := *false*.
II. At every *local round switch*, i.e., COND := $\left[R_i(t) = R_i(t-1) + 1\right]$.
III. At every *global round switch*, i.e., COND := $\left[R_G(t) = R_G(t-1) + 1\right]$.
IV. Always, i.e., COND := *true*.

Formally, we write $\mathcal{M}_{i,j}(t) = 0$ if process j's message to process i sent at time t was lost, and $\mathcal{M}_{i,j}(t) = 1$ if it arrives (at time $t + 1$). Process i's computation in its step at time t consists of the following:

1. *Update knowledge according to received messages:* $K_{i,j}(t) \leftarrow R_j(t-1)$ if $\mathcal{M}_{i,j}(t-1) = 1$, 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 $K_{i,j}(t) \geqslant R_i(t-1)$ for all j, and $R_i(t) \leftarrow R_i(t-1)$ otherwise.
3. *Conditional forget:* $K_{i,j}(t) \leftarrow 0$ if COND is true.

Initially, $K_{i,j}(0) = 0$, and no messages are received at time 0. In particular, $R_i(0) = 1$. In the remainder of this paper, when we refer to $K_{i,j}(t)$, we mean its value after step 3.

We assume that the $\mathcal{M}_{i,j}(t)$ are pairwise independent random variables with

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

Call the parameter p the *probability of successful transmission*.

Fig. 7 shows part of an execution for condition I on forgetting. 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 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.

3. Performance measure

Clearly, the extent of forgetting influences the expected round duration λ. For case IV, where processes always forget, and for case III, where processes forget on global round switches, λ can be calculated efficiently with explicit formulas stated in Theorem 1. For the remaining cases, I and II, we could compute $\lambda(N, p)$ by means of a steady state analysis of a finite Markov chain, with time complexity exponential in N. We show how to do this in Section 5.2. The Markov chain model is also useful to study the behavior of λ, for all four conditions on forgetting, when $p \to 1$ and $p \to 0$. We do this in Sections 5.6 and 5.7, respectively. We derive explicit lower bounds on λ^{I} and λ^{II} in Section 5.4.

We will repeatedly use the dual of $R_i(t)$, namely $T_i(r)$, the time process i switches to round r. Further set $T_G(r) = \max_i T_i(r)$. The next proposition allows to calculate λ dually by:

Proposition 1. *For all four conditions on forgetting,* $\lambda = \mathbb{E} \lim_{t\to\infty} t/R_i(t) = \mathbb{E} \lim_{r\to\infty} T_i(r)/r$.

Proof. From the equality $T_i(r) = \inf\{t \mid R_i(t) = r\}$ we obtain $R_i(T_i(r)) = r$. It follows that $(T_i(r)/r)_{r\geqslant 1} = (T_i(r)/R_i(T_i(r)))_{r\geqslant 1}$. Since the latter is a subsequence of $(t/R_i(t))_{t\geqslant 0}$, both converge to the same value, which is equal to λ by definition. □

It is not hard to show, by comparing $T_i(r)$ for every fixed choice of the sequence $\mathcal{M} = (\mathcal{M}(t))_{t\geqslant 0}$, that

$$\lambda^{\mathrm{I}} \leqslant \lambda^{\mathrm{II}} \leqslant \lambda^{\mathrm{III}} \leqslant \lambda^{\mathrm{IV}}. \tag{2}$$

4. Explicit formulas for λ^{III} and λ^{IV}

From the expected maximum of geometrically distributed random variables (Proposition 2), we derive explicit formulas for λ^{III} and λ^{IV} in Theorem 1. For that purpose, define

$$\Lambda(M,p) = \mathbb{E} \max_{1\leqslant i\leqslant M} G_i, \tag{3}$$

where the G_i are pairwise independent geometrically distributed random variables with parameter p. We will make use of the following well-known proposition [18,19], whose proof we state for sake of completeness.

Proposition 2. $\Lambda(M,p) = \sum_{i=1}^{M} \binom{M}{i}(-1)^i \frac{1}{(1-p)^i-1}$.

Proof. Let $p_k = \mathbb{P}\left(\max_{1\leqslant i\leqslant M} G_i \leqslant k\right)$. Then $p_k = (1-(1-p)^k)^M$. Thus, with $q = 1-p$ and the binomial formula,

$$\Lambda(M,p) = \sum_{k\geqslant 1} k(p_k - p_{k-1}) = \sum_{k\geqslant 1} k \sum_{i=1}^{M} \binom{M}{i}(-1)^i q^{i(k-1)}\left(q^i-1\right),$$

changing the order of summation and shifting the summation index yields

$$= \sum_{i=1}^{M} \binom{M}{i}(-1)^i\left(q^i-1\right)\sum_{k\geqslant 0}(k+1)q^{ik}$$
$$= \sum_{i=1}^{M} \binom{M}{i}(-1)^i \frac{1}{\left(q^i-1\right)}. \quad \square$$

Consider case III, i.e., processes forget on global round switches. Initially, all processes i are in round $R_i(0) = 1$, and their knowledge is $K_{i,j}(0) = 0$. Observe that processes switch to round 2 as messages are received. At time t at which the last process switches to round 2, it holds that (i) all processes i have $R_i(t) = 2$, (ii) all processes have knowledge $K_{i,j}(t) \geqslant 1$ for all j before forgetting, and (iii) all processes forget, since a global round switch occurred, ultimately resulting in $K_{i,j}(t) = 0$. The only difference between the initial state and the state at time t is the constant round number offset $R_i(t) = R_i(0)+1$. By repeated application of the above arguments we obtain that the system is reset to the initial state modulo a constant offset in round numbers R_i, each time a global round switch occurs. This allows to determine the expected average round duration by analyzing the expected time until the first round switch.

We will now prove the explicit formulas for the expected round duration in cases III and IV. We will use these formulas in particular in Section 5.7 when studying the behavior of λ for $p \to 0$.

Proof of Theorem 1 $\lambda^{\mathrm{III}}(N,p)$**.** Recall that the events that i receives a message from j at time t are pairwise independent for all i, j and times t. Thus the smallest time t, at which i receives a message from j is geometrically distributed with parameter p. Noting that the first global round switch occurs at time $T_G(2) = \max_i(T_i(2))$, we obtain

$$\lambda(N,p) = \mathbb{E} \lim_{r\to\infty} T_G(r)/r = \mathbb{E}T_G(2) = \mathbb{E} \max_{1\leqslant i\leqslant N(N-1)} G_i$$

where the G_i are geometrically distributed with parameter p. The theorem now follows from Proposition 2. $\square$

Proof of Theorem 1 $\lambda^{\mathrm{IV}}(N,p)$**.** Observe that the first global round switch occurs at the minimum time t by which each of the processes has received messages from all processes simultaneously; and that $R_i(t) = 2$ as well as $K_{i,j}(t) = 0$ holds at this time. Again the state at time t is identical to the initial state with all round numbers incremented by 1. Repeated application of the above arguments allows to calculate the expected round duration by $\lambda(N,p) = \mathbb{E}T_G(2)$. The first time i receives a message from all processes simultaneously is geometrically distributed with parameter p^{N-1}. Since we have N nodes, we take the maximum over N such geometrically distributed random variables. The theorem now follows from Proposition 2. $\square$

5. Markovian analysis

Determining λ^{I} and λ^{II}, the expected round duration in the cases that processes never forget or forget at local round switches, is more involved. In the following, we will calculate λ by modeling the system as a finite Markov chain and analyzing its steady state distribution. Additionally, we derive the asymptotic behaviors for $p \to 1$ and for $p \to 0$ from the Markov chain model. As the computation of the chain's steady state distribution is computationally very expensive, we will give analytical lower bounds in Section 5.4.

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 $A(t) - \min_k A_{k,k}(t)$ cropping negative entries to -1, i.e., $a_{i,j}(t) = \max\{A_{i,j}(t) - \min_k A_{k,k}(t), \ -1\}$. Normalized states belong to the finite set $\{-1, 0, 1\}^{N\times N}$.

The sequence of normalized states $a(t)$ is a Markov chain: The probability that $A(t+1) = Y$, given that $A(t) = X$, is equal to the probability that $A(t+1) = Y + c$, given that $A(t) = X + c$ for any constant c. We may thus restrict ourselves without loss of generality to considering the system being in state $X - \min_i(X_{i,i})$ at time t. Further, by the algorithm and the fact that the precision is 1, cropping the entries of $X - \min_i(X_{i,i})$ at -1 does not lead to different transition probabilities: the probability that $A(t+1) = Y$ given that $A(t) = X - \min_i(X_{i,i})$ is equal to the probability that $A(t+1) = Y$ given that $A(t)$ is $X - \min_i(X_{i,i})$ cropped at -1. It follows that $a(t)$ is a finite Markov chain, for the algorithm with any of the four conditions on forgetting.

We will repeatedly need to distinguish whether there is a global round switch at time t or not. Let $\hat{a}(t)$ be the Markov chain obtained from $a(t)$ by adding to each state a an additional flag Step such that $\text{Step}(\hat{a}(t)) = 1$ if there is a global round switch at time t, and 0 otherwise.

5.1. Markov chains for conditions I and II

In the following we characterize the normalized states of the Markov chain $a(t)$, and their transition probability for the case that processes never forgets (I) and the case that processes forget at local round switches (II).

For any two states a and a' with nonzero transition probability from a to a', we introduce the notation of *relevant messages*: Given that $a(t) = a$, the message from i to j is *positively relevant* if it is required to be received for $a(t+1) = a'$ to hold. It is *negatively relevant* if it is required not to be received for $a(t+1) = a'$ to hold. Denote by $\text{prel}(a' \mid a)$ resp. $\text{nrel}(a' \mid a)$ the number of positively resp. negatively relevant messages for the transition from a to a'.

5.1.1. Condition I

In case the algorithm never forgets, the Markov chain $a(t)$ can attain exactly those matrices $a \in \{-1, 0, 1\}^{N\times N}$ that fulfill either properties M1–(M4) or (N1)–(N3):

(M1) $\min_j(a_{j,j}) = 0$.
(M2) For all i, if $a_{i,i} = 1$, then $\min_j(a_{i,j}) \geqslant 0$.
(M3) For all i, if $a_{i,i} = 0$, then $\min_j(a_{i,j}) = -1$.
(M4) For all i, j, $a_{i,j} \leqslant a_{j,j}$.

Hereby, (M1) is due to normalization. Letting r be the smallest local round number of a process at time t, (M2) requires a process i that is ahead, i.e. has $R_i(t) = r + 1$, having received round r messages from all processes by time t. Property (M3) requires a process i not being ahead, i.e. $R_i(t) = r$, not having received round r messages from all processes by time t. (M4) ensures that no process i has received a message from a process j with round number larger than j's local round number.

(N1) For all j, $a_{j,j} = 0$.
(N2) There is exactly one i with $a_{i,j} = 0$ for all j. For this i, $a_{j,i} = -1$ for all $j \neq i$.
(N3) is equivalent to (M4).

Properties (N1)–(N3) describe states that occur only at times at which a global round switch was performed: Assume that, at time t, all processes $j \neq i$ are ahead, and process i receives all messages from all the other processes at time $t+1$. Then i performs a local round switch at time $t+1$, and thus a global round switch occurs at time $t+1$. The resulting state $a(t+1)$ fulfills (N1)–(N3) and is unstable as i can perform a local round switch at time $t+2$ without receiving any messages from other processes.

Define $\text{Tran}^{\text{I}}(a' \mid a)$ to be 1 if a transition from state a to state a' is possible in one time step and 0 otherwise, i.e., $\text{Tran}^{\text{I}}(a' \mid a)$ equals 1 if and only if, for all i, $a'_{i,i} \geqslant a_{i,i}$ and for all $i \neq j$, $a'_{i,j} = a_{i,j} \vee a'_{i,j} = a_{j,j}$. For the number of positively and negatively relevant messages it holds that,

$$\text{prel}(a' \mid a) = |\{(i,j), i \neq j : a'_{i,j} > a_{i,j}\}| \quad \text{and}$$
$$\text{nrel}(a' \mid a) = |\{(i,j), i \neq j : a'_{i,j} = a_{i,j} \wedge a_{i,j} < a_{j,j}\}|.$$

Further let $P'(a' \mid a) = p^{\text{prel}(a'|a)} \cdot (1-p)^{\text{nrel}(a'|a)}$ be the probability that every positively relevant message arrives and every negatively relevant message is dropped. A transition from state a to a' occurs with probability $P'(a' \mid a)$, specified as follows:

- If (M1)–(M4) holds for a' and $\max_{i,j}(a'_{i,j}) \leqslant 0$, then

  $$P(a' \mid a) = \text{Tran}^{\text{I}}(a' \mid a) \cdot P'(a' \mid a) + \text{Tran}^{\text{I}}(a'+1 \mid a) \cdot P'(a'+1 \mid a);$$

  note that the first term is the probability of a transition from a to a' without a global round switch, the second term is the probability of a transition from a to a' with a global round switch: in the latter case we have to increment every entry of a' by 1 and then to apply our transition rules.
- If (M1)–(M4) holds for a' and $\max_{i,j}(a'_{i,j}) = 1$, then $P(a' \mid a) = \text{Tran}^{\text{I}}(a' \mid a) \cdot P'(a' \mid a)$.
- If (N1)–(N3) holds for a', then $P(a' \mid a) = \text{Tran}^{\text{I}}(a'+1 \mid a) \cdot P'(a'+1 \mid a)$.

5.1.2. Condition II

To analyze the case of the algorithm that forgets every local round switch we introduce a variant of the algorithm with forgetting condition II: we merge the conditional forgetting in step 3 of the algorithm with the update of the knowledge in step 1, into a step 1′. Process i's computation in its step at time t now consists of the following steps:

1′. *Bounded knowledge update according to received messages:* $K_{i,j}(t) \leftarrow \min(R_i(t-1), R_j(t-1))$ if $\mathcal{M}_{i,j}(t-1) = 1$, 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 $K_{i,j}(t) \geqslant R_i(t-1)$ for all j, and $R_i(t) \leftarrow R_i(t-1)$ otherwise.

Hereby, a process interprets any round number it receives as at most its own round number, i.e., forgets whether it was strictly larger. For example, a process i with $R_i = 0$ only interprets all received messages as 0 messages, whereas a process with $R_i = 1$ updates its knowledge according to the content of the message. Observe that both, the original algorithm and its variant, are equivalent in the sense that they perform local round switches at the same times, given that the received messages are the same: For both algorithms it holds that, after a process performed a local round switch, the elements of its knowledge vector are strictly less than its round number. To perform a local round switch a process has to receive the same set of messages in both algorithms.

Considering the variant of the algorithm, the resulting Markov chain $a(t)$ can attain exactly those states $a \in \{-1, 0, 1\}^{N\times N}$ with (M1)–(M3) and (M4′), where

(M4′) $a_{i,j} \leqslant \min(a_{i,i}, a_{j,j})$.

Define $\mathrm{Tran}^{\mathrm{II}}(a' \mid a)$ to be 1 if, for all i, $a'_{i,i} \geqslant a_{i,i}$ and for all $i \neq j$, $a'_{i,j} = a_{i,j} \vee a'_{i,j} = \min(a_{i,i}, a_{j,j})$; and 0 otherwise. The number of positively relevant messages is the same as in the case where processes never forget, and for the negatively relevant messages we obtain

$$\mathrm{nrel}(a' \mid a) = |\{(i,j),\ i \neq j : a'_{i,j} = a_{i,j} \wedge a_{i,j} < \min(a_{i,i}, a_{j,j})\}|.$$

Letting P' be as in Section 5.1.1, a transition from state a to a' occurs with the following probability $P(a' \mid a)$:

- If $\max_{i,j}(a'_{i,j}) \leqslant 0$, then
$$P(a' \mid a) = \mathrm{Tran}^{\mathrm{II}}(a' \mid a) \cdot P'(a' \mid a) + \mathrm{Tran}^{\mathrm{II}}(a'+1 \mid a) \cdot P'(a'+1 \mid a);$$
- If $\max_{i,j}(a'_{i,j}) = 1$, then $P(a' \mid a) = \mathrm{Tran}^{\mathrm{II}}(a' \mid a) \cdot P'(a' \mid a)$.

5.2. Using the steady state to calculate λ

Call a Markov chain *good* if it is aperiodic, irreducible, Harris recurrent, and has a unique steady state distribution. It is not difficult to see that $\hat{a}(t)$ is good for all four conditions on forgetting.

A standard method, given the chain's transition matrix P, to compute the steady state distribution π is by matrix inversion:

$$\pi = e \cdot \left(P^{(n\to 1)} - I^{(n\to 0)}\right)^{-1} \tag{4}$$

where $M^{(k\to x)}$ denotes matrix M with its kth column set to x, I is the identity matrix, and $e = (1, 1, \ldots, 1)$.

We will next show how to compute λ from the steady state distribution π.

Theorem 4. *Let $X(r)$ be good Markov chain with state space $\mathcal{X}$ and steady state distribution π. Further, let $g : \mathcal{X} \to \mathbb{R}$ be a function such that $\sum_{X\in\mathcal{X}} |g(X)| \cdot \pi(X) < \infty$. Then, $\lim_{r\to\infty} \frac{1}{r} \sum_{k=1}^{r} g\big(X(k)\big) = \sum_{X\in\mathcal{X}} g(X) \cdot \pi(X)$ with probability 1 for every initial distribution.*

Proof (*[20, Theorem 17.0.1(i)]*). □

We call a processes i a 1-*process* in state $\hat{a}$ if $\hat{a}_{i,i} = 1$. Likewise, we call i a 0-*process* in $\hat{a}$ if $\hat{a}_{i,i} = 0$. Denote by $\#_{-1}(\hat{a})$ the number of -1 entries in rows of matrix $\hat{a}$ that correspond to 0-processes in $\hat{a}$ (note that a row for a 1-process cannot contain -1).

Theorem 5. *For all conditions of forgetting, $R_i(t)/t \to 1/\lambda$ with probability 1 as $t \to \infty$. Furthermore,*

$$\lambda = 1/\left(\sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \cdot \pi(\hat{a})\right) = 1/\left(\sum_{a} p^{\#_{-1}(a)} \cdot \pi(a)\right). \tag{5}$$

Proof. It holds that $R_G(t) = \sum_{k=1}^{t} \mathrm{Step}\big(\hat{a}(k)\big)$. By Theorem 4, with probability 1 it holds that:

$$\lim_{t\to\infty} R_i(t)/t = \lim_{t\to\infty} R_G(t)/t = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \mathrm{Step}(\hat{a}(k)) = \sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \pi(\hat{a}).$$

Since $\hat{a}(t)$ is a finite Markov chain, the last sum is finite. It follows that $R_i(t)/t$ converges to a constant, say c, with probability 1. Thus $t/R_i(t)$ converges to $1/c$ with probability 1. By definition of λ, it follows that $\lambda = 1/c$. This shows the first part of the theorem.

The second part of the theorem is proved by the following calculation:

$$\begin{aligned}
1/\lambda &= \mathbb{E} \lim_{t\to\infty} R_i(t)/t = \mathbb{E} \lim_{t\to\infty} R_G(t)/t = \mathbb{E} \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \mathrm{Step}\big(\hat{a}(k)\big) \\
&= \sum_{\hat{a}} \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \mathbb{P}\big(\hat{a}(k-1) = \hat{a}\big) \cdot \mathbb{E}\big(\mathrm{Step}(\hat{a}(k)) \mid \hat{a}(k-1) = \hat{a}\big) \\
&= \sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \mathbb{P}\big(\hat{a}(k-1) = \hat{a}\big) = \sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \cdot \pi(\hat{a}).
\end{aligned}$$

From $\pi(\hat{a}) = \pi((a,0)) + \pi((a,1)) = \pi(a)$ the theorem follows. □

5.3. Computational complexity of the Markov approach

Space and time complexity of calculating λ for cases I and II depend on the number of states $a(t)$ of the respective finite Markov chain. A simple strategy to reduce the number of states is to consider only those with sorted diagonal, i.e., processes 1 to some k are 1-processes, and processes $k+1$ to n are 0-processes. We, however, show that asymptotically this reduction does not lead to improved computational complexities.

Proposition 3. *For the variant of the case II synchronizer introduced in Section 5.1.2, the number of states $a(t)$ is*

$$A := \sum_{i=0}^{N-1} \binom{N}{i} 2^{i(i-1)} \left(2^{N-1} - 1\right)^{N-i}.$$

Using the equivalence of states with the same sorted diagonal, the number of states $a(t)$ reduces to

$$B := \sum_{i=0}^{N-1} 2^{i(i-1)} \left(2^{N-1} - 1\right)^{N-i},$$

where

$$A \sim B \sim 2^{N(N-1)}.$$

Calculating λ^{II} thus is in $O(2^{3N(N-1)})$.

Proof. We first show that the number of states is A. For a fixed number i of 1s in the diagonal we have $\binom{N}{i}$ possibilities to place them there. The knowledge one process has of another one, i.e., $i(i-1)$ entries, can be 1 or 0. For each 0-process each entry can be 0 and -1 but not all channels can be set to 0.

For proving the asymptotics note that, by only taking the summand for $i = 0$,

$$A \geqslant \left(2^{N-1} - 1\right)^{N} \sim 2^{N(N-1)}.$$

Moreover,

$$A \leqslant \sum_{i=0}^{N-1} \binom{N}{i} 2^{i(i-1)} 2^{(N-1)N} = 2^{N(N-1)} \left(1 + \sum_{i=1}^{N-1} \binom{N}{i} 2^{-i(N-i)}\right).$$

The sum on the right-hand side converges to 0, which completes the proof for A.

If we use states with sorted diagonal, clearly we obtain B from A be just omitting the binomial coefficient. The asymptotics can be shown analogous to A. □

Proposition 4. *For the non-forgetting case I synchronizer, the number of states $a(t)$ is*

$$C := \sum_{i=0}^{N-1} \binom{N}{i} 2^{i(i-1)} \left(3^{i} \cdot 2^{N-1-i} - 2^{i}\right)^{N-i} + N 2^{(N-2)(N-1)}.$$

Using the equivalence of states with the same sorted diagonal, the number of states $a(t)$ reduces to

$$D := \sum_{i=0}^{N-1} 2^{i(i-1)} \left(3^{i} \cdot 2^{N-1-i} - 2^{i}\right)^{N-i} + 2^{(N-2)(N-1)},$$

where

$$C \sim D \sim 2^{N(N-1)}.$$

Calculating λ^{I} *thus is in* $O(2^{3N(N-1)})$.

Proof. We start with determining C. The sum counts those matrices that fulfill properties (M1)–(M4) (see Section 5.1.1): For placing i ones in the diagonal we have $\binom{N}{i}$ possibilities. The entries corresponding to the $i(i-1)$ channels between 1-processes can be 0 or 1. For a 0-process there are 3 possible values: 0, 1, and -1 for the channels incoming from a 1-process, and 2 possible values, 0 and -1, for the channels incoming from a 0-process. Since there must be at least one entry equal to -1 we have to subtract all combinations without a -1, i.e., 2^i many.

On the other hand, the remaining term $N2^{(N-2)(N-1)}$ counts those matrices that fulfill properties (N1)–(N3): We have N possibilities to choose the process i from condition (N2). The entries in the corresponding row and column are fixed, and all the other entries equals -1 or 0.

The term for $i = 0$ equals $\left(2^{N-1} - 1\right)^N$ thus

$$C \geqslant \left(2^{N-1} - 1\right)^N \sim 2^{N(N-1)}.$$

Moreover,

$$\begin{aligned} C &\leqslant \sum_{i=0}^{N-1} \binom{N}{i} 2^{i(i-1)} \left(3^i \cdot 2^{N-1-i}\right)^{N-i} + N2^{(N-2)(N-1)} \\ &= 2^{N(N-1)} \left(\sum_{i=0}^{N-1} \binom{N}{i} 2^{-2i(N-i)} 3^{i(N-i)} + N2^{-2(N-1)} \right) \\ &= 2^{N(N-1)} \left(1 + \sum_{i=1}^{N-1} \binom{N}{i} \left(\frac{3}{4}\right)^{i(N-i)} + N2^{-2(N-1)} \right). \end{aligned}$$

The sum on the right-hand side converges to 0, which completes the proof for C. If we use states with sorted diagonal, clearly we obtain D from C be just omitting the binomial coefficient in the sum and the factor N in the remaining term. The asymptotics can be shown analogous to C. □

5.4. Lower Bounds on λ^{I} and λ^{II}

As has been shown, determining the expected round duration for cases I and II by means of the Markov chain $a(t)$ is computationally expensive, even for small system sizes N. We can, however, efficiently compute lower and upper bounds on $\lambda(N, p)$: For both, cases I and II, $\lambda^{\mathrm{III}}(N, p)$ is an upper bound. We will next derive computationally feasible lower bounds for $\lambda^{\mathrm{I}}(N, p)$ and $\lambda^{\mathrm{II}}(N, p)$. Bear in mind that in case II we will always refer to the simplified variant of our algorithm introduced in Section 5.1.2.

From Proposition 1 and Theorem 4 follows, by considering the conditional expectation of T_G:

$$\lambda = \frac{1}{\sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \pi(\hat{a})} \sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \pi(\hat{a}) \cdot \mathbb{E}(T_G(2) \mid \hat{a}(0) = \hat{a}),$$

where $\mathbb{E}(T_G(2) \mid \hat{a}(0) = \hat{a})$ is the expected time until the first global round switch, given that the system initially is in state $\hat{a}$. It holds that $\mathbb{E}(T_G(2) \mid \hat{a}(0) = \hat{a}) = \Lambda(\#_{-1}(\hat{a}), p)$, where Λ is the expectation of the maximum of geometrically distributed random variables as defined in (3).

Denote by $\#_0(\hat{a})$ the number of non-diagonal entries in $\hat{a}$ equal to 0. Let $[n]$ denote the set of states $\hat{a}$ with $\#_{-1}(\hat{a}) = n$ and $\mathrm{Step}(\hat{a}) = 1$, and denote by $\bigcup[n]$ the union of all $[n]$ for $0 \leqslant n \leqslant N(N-1)$. Further let $\alpha := \sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \pi(\hat{a})$ be the steady-state-probability that the system is in one of the states in which a global round switch was performed, and denote with $\hat{\pi}(n) = \alpha^{-1} \sum_{\hat{a} \in [n]} \pi(\hat{a})$ the conditioned steady-state-probability that we are in one of the states with exactly n entries equal to -1.

Our first observation is that $\hat{\pi}(n) = 0$ for $n < 2N - 2$ in case II and $\hat{\pi}(n) = 0$ for $n < N - 1$ in case I: If a global round switch occurs at time t, then there exists one process – say j – which does a local round switch at time t (in fact, all 0-processes after time $(t-1)$ do so). This process must have sent 0-messages to all the other processes. In terms of the matrices this means that $\hat{a}_{i,j} = 0$ for all $i \neq j$. Due to the global round switch, the entries of the matrix are reduced by -1, resulting in $\hat{a}_{i,j} = -1$ in $\hat{a}$ for all $i \neq j$. Consequently, after a global round switch we have at least $(N-1)$ entries equal to -1 and $\hat{\pi}(n) = 0$ for $0 \leqslant n < N - 1$. Moreover, in case II, condition (M4′) implies that also the messages received by process j are treated as 0-messages, i.e., $a_{j,i} = 0$ for all $i \neq j$ when processes j makes a step. After reducing the matrix by 1 we obtain $a_{j,i} = -1$ for all $i \neq j$ and so there are at least $2(N-1)$ entries equal to -1.

The basic idea of the bounds on λ is to bound $\hat{\pi}(n)$. Let $\mathbb{P}(\hat{a} \rightsquigarrow [n])$ be the probability that, given the system is in state $\hat{a}$ at some time t, for the minimum time $t' > t$ at which a global round switch occurs, $\hat{a}(t') \in [n]$. Starting from the steady-state

equation for $\hat{\pi}(n)$, we obtain:

$$\begin{aligned}\hat{\pi}(n) &= \frac{1}{\alpha}\sum_{\hat{a}} \mathrm{Step}(\hat{a})\cdot\pi(\hat{a})\cdot\mathbb{P}(\hat{a}\rightsquigarrow[n]) = \frac{1}{\alpha}\sum_{\hat{a}\in\bigcup[n]}\pi(\hat{a})\cdot\mathbb{P}(\hat{a}\rightsquigarrow[n])\\ &= \frac{1}{\alpha}\sum_{\hat{a}\in[n]}\pi(\hat{a})\cdot\mathbb{P}(\hat{a}\rightsquigarrow[n]) + \frac{1}{\alpha}\sum_{\hat{a}\in\bigcup[n]\setminus[n]}\pi(\hat{a})\cdot\mathbb{P}(\hat{a}\rightsquigarrow[n])\\ &\geqslant \hat{\pi}(n)\min_{\hat{a}\in[n]}\mathbb{P}(\hat{a}\rightsquigarrow[n]) + (1-\hat{\pi}(n))\min_{\hat{a}\in\bigcup[n]\setminus[n]}\mathbb{P}(\hat{a}\rightsquigarrow[n])\\ &\geqslant \hat{\pi}(n)c_n + (1-\hat{\pi}(n))d_n\end{aligned}$$

for c_n, d_n suitably chosen. Hence,

$$\hat{\pi}(n) \geqslant \frac{d_n}{1+d_n-c_n} =: \pi_n.$$

So, our goal is to find c_n and d_n, and by that obtain the lower bound π_n on $\hat{\pi}(n)$. Since we will choose different c_n and d_n for different conditions on forgetting, we will write d_n^{I} and c_n^{I} in case processes never forget, and d_n^{II} and c_n^{II} if we consider the case where processes forget on local round switches. The resulting bounds are denoted by π_n^{I} and π_n^{II} respectively. Since Λ is nondecreasing in its first argument, we can bound $\lambda(N,p)$ by

$$\left(1-\sum_{n=N}^{N(N-1)}\pi_n^{\mathrm{I}}\right)\Lambda(N-1,p) + \sum_{n=N}^{N(N-1)}\pi_n^{\mathrm{I}}\Lambda(n,p) \leqslant \lambda^{\mathrm{I}}(N,p) \tag{6}$$

in case I. For case II we obtain

$$\left(1-\sum_{n=2N-1}^{N(N-1)}\pi_n^{\mathrm{II}}\right)\Lambda(2N-2,p) + \sum_{n=2N-1}^{N(N-1)}\pi_n^{\mathrm{II}}\Lambda(n,p) \leqslant \lambda^{\mathrm{II}}(N,p). \tag{7}$$

5.4.1. Lower bound on λ^{II}

We start our analysis for case II with determining $\pi_{N(N-1)}$. Since $\mathbb{P}(\hat{a}\rightsquigarrow[N(N-1)])$ is greater than the probability that $\hat{a}(t+1)\in[N(N-1)]$, given that $\hat{a}(t)=\hat{a}$, for arbitrary t, we have $\mathbb{P}(\hat{a}\rightsquigarrow[N(N-1)]) \geqslant p^{\#_{-1}(\hat{a})}$. Thus we may choose $c_{N(N-1)}^{\mathrm{II}} = p^{N(N-1)}$, $d_{N(N-1)}^{\mathrm{II}} = p^{N(N-1)-1}$ and obtain

$$\pi_{N(N-1)} = \frac{p^{N(N-1)-1}}{1+p^{N(N-1)-1}(1-p)}.$$

Next we turn to the analysis of $\pi_{N(N-1)-1}$. Note that it is not possible to make a direct transition from a state $\hat{a}\in\bigcup[n]$ to a state in $[N(N-1)-1]$, since we need a 0-entry in the matrix after the global round switch. So we need at least two time steps: We have to generate at least two 1-processes first, and afterwards exactly one 1-message between 1-processes must arrive (note that due to (M4′) messages from 1-processes to 0-processes are treated as 0-messages). Our lower bounds are based on the probability that the system is in a state within $[N(N-1)-1]$ at time $t+2$, given that $\hat{a}(t)=\hat{a}$. In fact, we restrict ourselves not only to the two-step-probability, but consider only specific transitions for our lower bound.

So fix in $\hat{a}$ one column j whose all non-diagonal entries equal -1; clearly such a column exists, since $\mathrm{Step}(\hat{a})=1$. For our bound, we consider the following specific scenario: Given that $\hat{a}(t)=\hat{a}$, assume that at time $t+1$, all messages from processes $i\neq j$ to all processes i' with $K_{i',i}(t)=-1$, and exactly one message from process j to some fixed $j'\neq j$, are received. Moreover, we allow k more (up to $(N-3)$) of the remaining $(N-2)$ messages sent by j to be received. That is, $N(N-2)+2-\#_0(\hat{a})+k$ messages are received. Moreover, we have that $k+2$ of the processes are 1-processes at time $t+1$ (i.e., processes j, j', and those k processes receiving the additional messages from j). For $\hat{a}(t+1)\in[N(N-1)-1]$ to hold, it is sufficient that all 0-processes i with $\hat{a}_{i,j}(t+1)=-1$ receive a message from j at time $t+2$, and exactly one of the messages from a 1-process to a 1-process is received. Note that we need not rule out the messages from 1-processes to 0-processes here, as (M4′) implies that they result in -1-entries (after the global round switch). Since at time $t+1$ there are $(k+2)(k+1)$ messages from 1-processes to 1-processes, we obtain: For all $\hat{a}\in\bigcup[n]$,

$$\begin{aligned}\mathbb{P}(\hat{a}\rightsquigarrow[N(N-1)-1]) &\geqslant \sum_{k=0}^{N-3}\binom{N-2}{k}p^{N(N-2)+2-\#_0(\hat{a})+k}(1-p)^{N-2-k}\\ &\quad\cdot p^{N-2-k}\cdot p\cdot(1-p)^{(k+2)(k+1)-1}\cdot((k+2)(k+1))\\ &= p^{N(N-1)-\#_0(\hat{a})+1}\cdot\sum_{k=0}^{N-3}\binom{N-2}{k}((k+2)(k+1))(1-p)^{N+k^2+2k-1}\\ &=: \beta(\#_0(\hat{a})).\end{aligned}$$

So we choose $c_{N(N-1)-1}^{\mathrm{II}} = \beta(1)$ and $d_{N(N-1)-1}^{\mathrm{II}} = \beta(0)$.

Finally, we turn to the analysis of π_n for $n = 2(N-1)+x$, where $0 \leqslant x \leqslant (N-2)(N-1)-2$. Again we bound $\mathbb{P}(\hat{a} \rightsquigarrow [2(N-1)+x])$, for $\hat{a} \in \bigcup[n]$, by determining the probability that $\hat{a}(t+2) \in [2(N-1)+x]$, given that $\hat{a}(t) = \hat{a}$, under specific transitions. For our purpose, fix a row j of $\hat{a}$ with T non-diagonal entries equal to 0. Assume that at time $t+1$, all messages to processes $i \neq j$ from all processes i' with $K_{i,i'}(t) \neq 0$ are received. Additionally, k more (up to $N-T-2$) of the remaining $N-T-1$ messages to j are allowed to be received. That is, $(N-1)(N-1) - \#_0(\hat{a}) + T + k$ messages are received. Hence, all processes except process j are 1-processes at time $t+1$. Afterwards, at time $(t+2)$, all the remaining messages to process j must arrive. Moreover, from the $(N-2)(N-1)$ messages sent by 1-processes to 1-processes exactly x are not allowed to be received for $\hat{a}(t+2) \in [2(N-1)+x]$ to hold. Thus, for fixed row j and $\hat{a} \in \bigcup[n]$,

$$\begin{aligned}
&\mathbb{P}(\hat{a} \rightsquigarrow [2(N-1)+x] \mid \text{ row } j \text{ has } T \text{ 0-entries}) \\
&\geqslant \sum_{k=0}^{N-2-T} \binom{N-T-1}{k} p^{k+(N-1)^2-\#_0(\hat{a})+T} \\
&\quad \cdot (1-p)^{N-1-k-T} p^{N-1-k-T} p^{(N-2)(N-1)-x} (1-p)^x \cdot \binom{(N-1)(N-2)}{x} \\
&= \binom{(N-1)(N-2)}{x} (1-p)^x p^{N(N-1)-\#_0(\hat{a})+(N-2)(N-1)-x} \cdot \left((2-p)^{N-1-T} - 1\right) \\
&=: \gamma(\#_0(\hat{a}), T, x).
\end{aligned}$$

Note that γ is nonincreasing in its second argument. Every state $\hat{a}$ has at least one row with $T = 0$ non-diagonal entries equal to 0. All other rows must have $T \leqslant N-2$ non-diagonal entries equal to 0, since a row must have at least one entry equal to -1. Thus, we have

$$\mathbb{P}(\hat{a} \rightsquigarrow [2(N-1)+x]) \geqslant \gamma(\#_0(\hat{a}), 0, x) + (N-1) \cdot \gamma(\#_0(\hat{a}), N-2, x) =: \tilde{\gamma}(\#_0(\hat{a}), x).$$

We may thus choose $c^{\mathrm{II}}_{2(N-1)+x} = \tilde{\gamma}((N-1)(N-2)-x, x)$ and $d^{\mathrm{II}}_{2(N-1)+x} = \tilde{\gamma}(0, x)$.

The lower bound on λ^{II} follows from (7).

5.4.2. Lower bound on λ^{I}

For $\pi_{N(N-1)}$ we may choose $d^{\mathrm{I}}_{N(N-1)} = d^{\mathrm{II}}_{N(N-1)}$ and $c^{\mathrm{I}}_{N(N-1)} = c^{\mathrm{II}}_{N(N-1)}$, by the same arguments as in Section 5.4.1.

To determine $\pi_{N(N-1)-1}$, we use the same construction as in Section 5.4.1, with the modification that at time $t+2$, exactly one of the $(k+2)(N-1)$ messages sent by 1-processes is received. We thus obtain for all $\hat{a} \in \bigcup[n]$,

$$\begin{aligned}
\mathbb{P}(\hat{a} \rightsquigarrow [N(N-1)-1]) &\geqslant \sum_{k=0}^{N-3} \binom{N-2}{k} p^{N(N-2)+2-\#_0(\hat{a})+k} (1-p)^{N-2-k} p^{N-2-k} p (1-p)^{(k+2)(N-1)-1} (k+2)(N-1) \\
&= p^{N(N-1)-\#_0(\hat{a})+1} \sum_{k=0}^{N-3} \binom{N-2}{k} (k+2)(N-1) \cdot (1-p)^{N-2-k+(k+2)(N-1)-1} \\
&=: \beta'(\#_0(\hat{a})).
\end{aligned}$$

So we choose $c^{\mathrm{I}}_{N(N-1)-1} = \beta'(1)$ and $d^{\mathrm{I}}_{N(N-1)-1} = \beta'(0)$.

Next consider π_n with $n = (N-1)+x$, and $0 \leqslant x \leqslant (N-2)(N-1)+1$. Choose an arbitrary $\hat{a} \in [n]$. It holds that $\hat{a}$ has at least $N-1$ non-diagonal entries equal to 0. Now fix a row j with $T \leqslant N-2$ non-diagonal entries equal to 0. We use the same construction as in case II, but note that at time $t+2$, the messages received by j lead to 0-entries. Thus,

$$\begin{aligned}
&\mathbb{P}(\hat{a} \rightsquigarrow (N-1)+x \mid \text{ row } j \text{ contains } T \text{ 0-entries}) \\
&\geqslant \sum_{k=0}^{N-2-T} \binom{N-T-1}{k} p^{k+(N-1)^2-\#_0(\hat{a})+T} \cdot (1-p)^{N-1-k-T} p^{N-1-k-T} \\
&\quad \cdot p^{N(N-1)-(N-1)-x-(N-1-k-T)} \cdot (1-p)^{(N-1)^2-(N(N-1)-(N-1)-x)} \cdot \binom{(N-1)^2-(N-1-k-T)}{x} \\
&\geqslant p^{2N^2-4N+T-x+2-\#_0(\hat{a})} (1-p)^x \binom{(N-1)(N-2)}{x} \cdot \sum_{k=0}^{N-2-T} \binom{N-T-1}{k} p^k (1-p)^{N-1-k-T} \\
&= p^{2N^2-4N+T-x+2-\#_0(\hat{a})} (1-p)^x \binom{(N-1)(N-2)}{x} \cdot (1-p^{N-1-T}) \\
&=: \gamma(\#_0(\hat{a}), T, x).
\end{aligned}$$

Note that γ is nonincreasing in its second argument. Every state $\hat{a} \in [n]$ has at least one row with $\lceil n/N \rceil$ non-diagonal entries equal to -1. Such a row must have $T \leqslant N-1-\lceil n/N \rceil$ non-diagonal entries equal to 0. Thus, we have

$$\mathbb{P}(\hat{a} \rightsquigarrow [(N-1)+x]) \geqslant \gamma(\#_0(\hat{a}), N-1-\lceil n/N \rceil, x).$$

We may thus choose $c^{\mathrm{I}}_n = \gamma((N-1)^2-x, N-1-\lceil n/N \rceil, x)$ and $d^{\mathrm{I}}_n = \gamma(0, N-1-\lceil n/N \rceil, x)$.

Note that this construction fails in case π_n for $N(N-1)-N+2 \leqslant n \leqslant N(N-1)-2$: Since we do not restrict the number of successfully transmitted message to process j at time $(t+1)$ it may happen that we generate up to $(N-1)$ 1-entries at time $(t+2)$ (and thus up to $(N-1)$ 0-entries after the global round switch) and this could violate the condition $N(N-1)-N+2 \leqslant n \leqslant N(N-1)-2$. To complete our lower bound we just define $\pi_n = 0$ in this case. Now the lower bound follows from (6).

5.5. Computational complexity of the lower bounds

The following proposition shows that the computational complexity of the lower bounds is polynomial.

Proposition 5. *Both the lower bound for λ^{II} and for λ^{I} can be computed by $O(N^4)$ multiplications/additions.*

Proof. We start our analysis by determining the computational complexity for π_x^{II}, with $2N-1 \leqslant x \leqslant N(N-1)$. Since computing powers p^n is in $O(\log n)$, $\pi_{N(N-1)}^{\mathrm{II}}$ and $\pi_{N(N-1)-1}^{\mathrm{II}}$ can be computed in $O(\log N)$ and $O(N \log N)$, respectively. To compute $\pi_{2(N-1)+x}^{\mathrm{II}}$ for $0 \leqslant x \leqslant (N-2)(N-1)-2$ we need to compute $\gamma((N-1)(N-2)-x, 0, x)$, $\gamma((N-1)(N-2)-x, N-2, x)$, $\gamma(0, 0, x)$, and $\gamma(0, N-2, x)$. For $x = 0$ each of these four evaluations of γ can be done in $O(\log N)$. Having stored the values for x we can obtain the values for $(x+1)$ in constant time. Hence, computing all $((N-2)(N-1)-2)$ values π_x^{II} is in $O(N^2)$.

Moreover, due to (7), we need N^2 evaluations of Λ (given in Proposition 2), each of which can be done in $O(N^2)$: Note that we have at most $N(N-1)$ summands, each of which can be computed in constant time from the previous one. Thus we can compute the lower bound for case II in $O(N^4)$. Analogous arguments hold for the lower bound for case I. □

5.6. Behavior of λ for $p \to 1$

Theorem 2 provides means to approximate the expected round duration for all conditions on forgetting when messages are successfully received with high probability. Since this is typically the case for real-world systems, it allows to characterize their expected round duration very efficiently.

Proof of Theorem 2. Let $p \in (0, 1)$. Let $\pi_{N,p}(\hat{a})$ be the steady state probability of state $\hat{a}$ of Markov chain $\hat{a}(t)$. From Theorem 5, $1/\lambda(N, p) = \sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \cdot \pi_{N,p}(\hat{a})$. Then

$$\frac{d}{dp} 1/\lambda(N, p) = \sum_{\hat{a}} \#_{-1}(\hat{a}) \cdot p^{\#_{-1}(\hat{a})-1} \cdot \pi_{N,p}(\hat{a}) + \sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \cdot \frac{d}{dp} \pi_{N,p}(\hat{a}).$$

Evaluation of the derivative at $p = 1$ leads to

$$\frac{d}{dp} 1/\lambda(N, p)\Big|_{p=1} = \sum_{\hat{a}} \#_{-1}(\hat{a}) \cdot \pi_{N,1}(\hat{a}) + \sum_{\hat{a}} \frac{d}{dp} \pi_{N,p}(\hat{a})\Big|_{p=1}.$$

Observe that as p goes to 1, $\pi_{N,p}(\hat{a})$ goes to 0 for all states $\hat{a}$, except for $\hat{a}_0$, the state with 0 in the diagonal, -1 everywhere else, and $\mathrm{Step}(\hat{a}) = 1$. It is $\#_{-1}(\hat{a}_0) = N(N-1)$. Moreover, as p goes to 1, $\pi_{N,p}(\hat{a}_0)$ approaches 1. Hence,

$$= N(N-1) + \frac{d}{dp}\left(\sum_{\hat{a}} \pi_{N,p}(\hat{a})\right)\Bigg|_{p=1} = N(N-1) + 0,$$

as the sum of the steady state probabilities over all states a equals 1. The theorem follows from $\frac{d}{dp}\lambda(N, p)\big|_{p=1} = -\frac{d}{dp} 1/\lambda(N, p)\big|_{p=1} \cdot \lambda^2(N, 1)$ and $\lambda(N, 1) = 1$. □

5.7. Behavior of λ for $p \to 0$

In systems with unreliable communication, in which Theorem 2 is not applicable, Theorem 3 characterizes the asymptotic behavior of the expected round duration for all our conditions on forgetting. It turns out that λ^{I}, λ^{II}, and λ^{III} have the same order of growth for $p \to 0$, namely p^{-1}, while λ^{IV} has a higher order of growth.

Proof of Theorem 3. It is $(1-p)^i - 1 = \sum_{j=1}^{i} \binom{i}{j}(-p)^j = \Omega(p)$ for $p \to 0$. Hence by Theorem 1, $\lambda^{\mathrm{III}}(N, p) = O(p^{-1})$ for $p \to 0$.

For all conditions on forgetting, all transition probabilities of the Markov chain $\hat{a}(t)$ are polynomials in p. Hence by Eq. (4), all steady state probabilities $\pi(\hat{a})$ are rational functions in p. Theorem 5 then in particular implies that $\lambda^{\mathrm{I}}(N, p)$ is also rational in p. Clearly, $\lambda^{\mathrm{I}}(N, p) \to \infty$ as $p \to 0$. Hence $\lambda^{\mathrm{I}}(N, p)$ has a pole at $p = 0$ of order at least 1. This implies $\lambda^{\mathrm{I}}(N, p) = \Omega(p^{-1})$. From the inequalities $\lambda^{\mathrm{I}} \leqslant \lambda^{\mathrm{II}} \leqslant \lambda^{\mathrm{III}}$, the first part of the theorem follows.

To show the second part of the theorem, we prove the more general statement

$$\Lambda(M, p) \sim p^{-1} H_M \quad \text{for } p \to 0, \tag{8}$$

where H_M denotes the Mth harmonic number.

It is $(1-p)^i - 1 = -p \sum_{j=1}^{i} \binom{i}{j} (-p)^{j-1} \sim -p \cdot i$ for $p \to 0$. Thus, Proposition 2 shows that

$$p \cdot \Lambda(M, p) \sim \sum_{i=1}^{M} \binom{M}{i} (-1)^{i+1} \frac{1}{i}.$$

Denoting by S_M this last sum, we show $S_M = H_M$. This identity is well-known; a proof can be found in the textbook by Graham et al. [21, (6.72) and (6.73)]. We give a proof, by induction on M, for the sake of completeness. The case $M = 1$ is trivial. If $M \geqslant 2$, then

$$\begin{aligned} S_M &= \sum_{i=1}^{M} \binom{M-1}{i} (-1)^{i+1} \frac{1}{i} + \sum_{i=1}^{M} \binom{M-1}{i-1} (-1)^{i+1} \frac{1}{i} \\ &= H_{M-1} - \frac{1}{M} \sum_{i=1}^{M} \binom{M}{i} (-1)^i = H_{M-1} + \frac{1}{M} = H_M \end{aligned}$$

by the induction hypothesis. This shows (8) and concludes the proof. □

6. Conclusion

We studied the effect of actively discarding memory content on variants of the α-synchronizer with active forgetting. We obtained asymptotic formulas for the behavior of the expected round duration $\lambda(N, p)$ as the probability of successful transmission $p \to 0$ and $p \to 1$, as well as means to calculate $\lambda(N, p)$ for arbitrary N and p, which allow to assess whether the resulting loss of performance is acceptable for a specific application. In a nutshell (see Section 1 for details), there is not much difference between variants I (the original α-synchronizer) and II (forgetting at local round switches), neither for the expected round duration nor for the average age of the messages. By contrast, variant IV (forget at every time step) results in always fresh messages, albeit at the price of a considerably larger average round duration (in particular, for small values of p). More aggressive forms of forgetting may hence be a viable alternative for increasing the accuracy in certain sensor fusion applications.

Our results for variants III and IV also provide computationally feasible and accurate bounds on the performance of the original non-forgetting α-synchronizer. In fact, due to its huge state space, the explicit calculation of λ for variant I is in $O(2^{3N(N-1)})$, which is prohibitively expensive even for small system sizes. Thanks to our formulas, accurate predictions of its performance can be obtained also for large N.

References

[1] A. Basu, B. Charron-Bost, S. Toueg, Crash failures vs. crash + link failures, in: Proc. PODC, ACM, 1996, p. 246. http://dx.doi.org/10.1145/248052.248102.
[2] B. Awerbuch, Complexity of network synchronization, J. ACM 32 (4) (1985) 804–823.
[3] J. Widder, U. Schmid, The Theta-Model: Achieving synchrony without clocks, Distrib. Comput. 22 (1) (2009) 29–47. http://dx.doi.org/10.1007/s00446-009-0080-x. URL: http://www.vmars.tuwien.ac.at/documents/extern/1724/paper.pdf.
[4] M. Függer, A. Kößler, T. Nowak, U. Schmid, M. Zeiner, The effect of forgetting on the performance of a synchronizer, in: P. Flocchini, J. Gao, E. Kranakis, F. Meyer auf der Heide (Eds.), Algorithms for Sensor Systems, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, ISBN: 978-3-642-45345-8, 2014, pp. 185–200.
[5] E.F. Nakamura, A.A.F. Loureiro, A.C. Frery, Information fusion for wireless sensor networks: Methods, models, and classifications, ACM Comput. Surv. 39 (3) (2007).
[6] K. Marzullo, Tolerating failures of continuous-valued sensors, ACM Trans. Comput. Syst. 8 (4) (1990) 284–304.
[7] U. Schmid, K. Schossmaier, How to reconcile fault-tolerant interval intersection with the Lipschitz condition, Distrib. Comput. 14 (2) (2001) 101–111.
[8] J.Y. Halpern, Using reasoning about knowledge to analyze distributed systems, Annu. Rev. Comput. Sci. 2 (1) (1987) 37–68.
[9] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, Reasoning About Knowledge, MIT Press, Cambridge, MA, 1995.
[10] R. Fagin, J.Y. Halpern, Reasoning about knowledge and probability, J. ACM (ISSN: 0004-5411) 41 (2) (1994) 340–367. http://dx.doi.org/10.1145/174652.174658.
[11] M. Jayaram, G. Varghese, Crash failures can drive protocols to arbitrary states, in: Proc. PODC, ACM, 1996, pp. 247–256.
[12] S. Dolev, Self-Stabilization, MIT press, 2000.
[13] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation, Prentice Hall, Englewood Cliffs, 1989.
[14] S. Rajsbaum, Upper and lower bounds for stochastic marked graphs, Inform. Process. Lett. 49 (6) (1994) 291–295.
[15] S. Rajsbaum, M. Sidi, On the performance of synchronized programs in distributed networks with random processing times and transmission delays, IEEE Trans. Parallel Distrib. Syst. 5 (9) (1994) 939–950.
[16] T. Nowak, M. Függer, A. Kößler, On the performance of a retransmission-based synchronizer, Theoret. Comput. Sci. 509 (2013) 25–39.
[17] J.G. Propp, D.B. Wilson, Exact sampling with coupled Markov chains and applications to statistical mechanics, Random Struct. Algorithms 9 (1–2) (1996) 223–252.
[18] P. Kirschenhofer, H. Prodinger, A result in order statistics related to probabilistic counting, Computing 51 (1) (1993) 15–27.
[19] W. Szpankowski, V. Rego, Yet another application of a binomial recurrence. Order statistics, Computing 43 (4) (1990) 401–410.

[20] S. Meyn, R.L. Tweedie, Markov Chains and Stochastic Stability, second ed., Cambridge University Press, Cambridge, 2009, with a prologue by Peter W. Glynn.
[21] R.L. Graham, D.E. Knuth, O. Patashnik, Concrete Mathematics—A Foundation for Computer Science, Addison-Wesley, Reading, MA, 1989.

Matthias Függer received his M.Sc. (2006) and Ph.D. (2010) in computer engineering from TU Wien. He worked as an assistant professor at TU Wien, as a post-doctoral researcher at LIX, Ecole polytechnique, and currently as a post-doctoral researcher at the Max-Planck-Institut für Informatik. His main research interest is the formal study of the fundamentals of computationally highly restricted distributed devices; such as (fault-tolerant) distributed algorithms in hardware and biology.

Alexander Kößler received his M.Sc. (2009) and Ph.D. (2014) in computer engineering from TU Wien. He worked as an assistant professor and as a research assistant at TU Wien at the Institut für Technische Informatik in the Embedded Computing Systems Group. His main research interest is the performance analysis of distributed algorithms executed in real-time systems.

Thomas Nowak received an M.Sc. degree in computer engineering from Vienna UT in 2010 and a Ph.D. degree in computer science from École polytechnique in 2014. He is currently a post-doctoral teaching and research assistant at École normale supérieure in Paris, France. His research focuses on discrete event systems and distributed algorithms in dynamic fault-tolerant networks.

Ulrich Schmid is full professor and head of the Embedded Computing Systems Group at the Institut für Technische Informatik at TU Vienna. He studied computer science and mathematics and also spent several years in industrial electronics and embedded systems design. He authored and co-authored numerous papers in the field of theoretical and technical computer science and received several awards and prices, like the Austrian START-price 1996. His current research interests focus on the mathematical analysis of fault-tolerant distributed algorithms and real-time systems, with special emphasis on their application in systems-on-chips and networked embedded systems.

Martin Zeiner received his M.Sc. degree in mathematics from TU Wien in 2007 and his Ph.D. degree in mathematics from TU Graz in 2010. He is currently a post-doctoral researcher at TU Wien. His main research interests are the mathematical analysis of distributed algorithms, q-analogues of classical distribution functions, and combinatorics.