HAL
open science

The Effect of Forgetting on the Performance of a Synchronizer

Matthias Függer, Alexander Kössler, Thomas Nowak, Ulrich Schmid, Martin Zeiner

To cite this version:

Matthias Függer, Alexander Kössler, Thomas Nowak, Ulrich Schmid, Martin Zeiner. The Effect of Forgetting on the Performance of a Synchronizer. ALGOSENSORS 2013 - 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, Sep 2013, Sophia Antipolis, France. pp.185-200, 10.1007/978-3-642-45346-5_14 . hal-00993885

HAL Id: hal-00993885

https://hal.science/hal-00993885v1

Submitted on 20 May 2014

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

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

The Effect of Forgetting on the Performance of a Synchronizer*

Matthias Függer[1], Alexander Kößler[1], Thomas Nowak[2], Ulrich Schmid[1], and Martin Zeiner[1]

[1] ECS Group, TU Wien, Vienna, Austria
{fuegger,koe,s,mzeiner}@ecs.tuwien.ac.at
[2] Laboratoire d'Informatique, École polytechnique, Palaiseau, France
nowak@lix.polytechnique.fr

Abstract. We study variants of the α-synchronizer by Awerbuch (J. ACM, 1985) within a distributed message passing system with probabilistic message loss. The purpose of synchronizers is to maintain a virtual (discrete) round structure. Their idea essentially 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 its own current round.
In this work, we study how four different, naturally chosen, strategies of forgetting affect the performance of these synchronizers. The variants differ in the times when processes discard part of their accumulated knowledge during execution. Such actively forgetting synchronizers have applications, e.g., in sensor fusion where sensor data becomes outdated and thus invalid after a certain amount of time.
We give analytical formulas to quantify the degradation of the synchronizers' performance in an environment with probabilistic message loss. In particular, the formulas allow to explicitly calculate the performance's asymptotic behavior. Interestingly, all considered synchronizer variants behave similarly in systems with low message loss, while one variant shows fundamentally different behavior from the remaining three in systems with high message loss. The theoretical results are backed up by Monte-Carlo simulations.

1 Introduction

A set of sensor nodes collecting in-field data and exchanging it over an ad-hoc wireless network is a common setup for sensor fusion applications [9]. Message loss is typically a non negligible issue within such systems. A common strategy to deal with message loss is to run a synchronizer algorithm, whose purpose is to generate a virtual (discrete) round structure at the application layer such that, at each round step, a process receives all messages from all processes sent in the

* This work has been partially supported by the Austrian Science Fund (FWF), grant NFN RiSE (S11405).

current round. In this work we study a retransmission-based variant of the α-synchronizer, introduced by Awerbuch [1] as the first in a series of synchronizer algorithms for asynchronous message-passing systems. Its main idea is that each process continuously broadcasts its current round number together with the corresponding application data. In systems with high dynamics, this application data may vary between broadcasts, even within one round. A process starts the next round when it has received the messages of its current round from all other processes. Additionally it delivers the most actual data it has received in its current round to the application layer. The synchronizer then guarantees a synchronization precision equal to the diameter of the network graph. The original α-synchronizer by Awerbuch used 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.

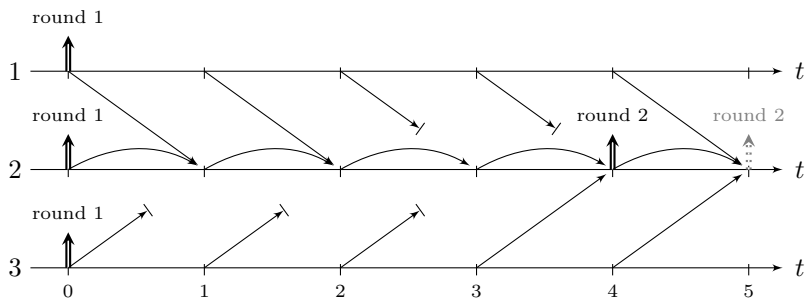


Fig. 1. Messages to process 2 and its resulting round switches without forgetting (black) and with forgetting (gray).

Figure 1 shows the beginning of an execution of the synchronizer executed in a system with three processes. For clarity, only messages to process 2 are shown. Time is assumed to elapse in discrete steps at all processes. We assume the existence of an underlying mechanism preventing the processes' discrete time from diverging, i.e., a synchronous system. At each point in time a process broadcasts its application data, e.g., the current sensor reading. 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. Note, however, that the age of the round 1 data it hands 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, it is typically represented by an interval (i.e., a value $\pm$ some accuracy) that detoriates with time [9]. 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 [7, 13], relying on old data obviously yields imprecise results.

A strategy to counteract this problem is to let the synchronizer actively "forget" old data it has received by 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 times between round switches. The resulting execution is depicted in gray in Figure 1 with the difference that process 2 then switches to round 2 only at time 5.

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 that never forgets, the variant in which a process forgets when it makes a round switch, and the variant that forgets at each time step, can be implemented in a distributed manner, the variant which forgets when the last process makes a round switch serves as a theoretical bound only.

We study the impact of forgetting on the performance of the synchronizer variants in an environment where every message transmission succeeds with a certain probability. 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.

Detailed Contribution. We make the following contributions in this paper: (1) We formally introduce the notion of forgetting in the context of a (specific) synchronizer. We consider four different conditions on when processes forget and study the respective degrading effects on the synchronizer's performance in a probabilistic environment. (2) We state explicit formulas for the expected round duration for two of these conditions and give efficient bounds for the other two conditions. These bounds are shown to approximate the true value well if the probability p of successful message transmission is high. (3) We show that for all four conditions, the expected round durations collapse when $p \to 1$: All four expected round durations, as well as its first derivative as a function of p, are equal in $p = 1$. (4) We prove that for $p \to 0$, the expected round duration for three of the conditions has the same order of growth, which we calculate explicitly for all four conditions. (5) We present simulation results of the expected round duration, comparing them to our calculations, and simulation results for the average age of data when a process makes a round switch.

Related Work. Our notion of knowledge is different from that of Fagin et al. [3], who studied the evolution of knowledge in distributed systems with powerful agents; in particular, their agents do not forget. While Mahesh and Varghese [5] use crashing processes and the forgetting during reboot in a destructive way, we use forgetting in a constructive manner. Nowak et al. [10] calculated the expected round duration of a retransmission-based synchronizer when a single transmission arrives with constant probability p, but a message that was retransmitted at least M times is guaranteed to arrive. They did not investigate the impact of forgetting on the synchronizer's performance, and assumed M to be finite,

which we do not. Bertsekas and Tsitsiklis [2] proved bounds for the case of constant processing times and exponentially distributed message delays. They did not derive exact performance measures. Rajsbaum [11] presented bounds on the synchronizer rate for the case of exponentially distributed processing times and transmission delays. Rajsbaum and Sidi [12] calculated the rate's exact value in the case of exponentially distributed processing times and negligible transmission delays.

Organization of the Paper. The rest of the paper is organized as follows. Section 2 introduces our system model, the studied synchronizer algorithm, and the four conditions on forgetting we investigate, i.e., when the processes forget the gained knowledge. In Section 3 we give the performance measure and we derive explicit formulas for two of the four conditions in Section 4. Section 5 uses a Markov chain model to compute the expected round duration of the remaining two conditions on forgetting and states results on the asymptotic behavior of the expected round duration. It also presents analytical lower bounds that facilitate estimations of the expected round duration. In Section 6 we compare the performance as well as the average age of data achieved by the different conditions on forgetting against each other.

2 System Model and Algorithm

In this paper we study the performance of variants of the α-synchronizer [1] running in a fully-connected message passing system with processes $1, 2, \ldots, N$. 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.

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 := $\big[R_i(t) = R_i(t-1) + 1\big]$.
III. At every *global round switch*, i.e., COND := $\big[R_G(t) = R_G(t-1) + 1\big]$.

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

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

We call the parameter p the *probability of successful transmission*.

Figure 2 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

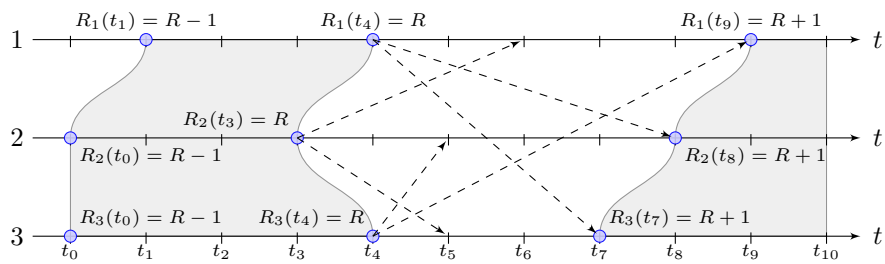


Fig. 2. An execution of the synchronizer

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

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. To distinguish the four proposed conditions on forgetting, I to IV, we will write λ^{I}, λ^{II}, λ^{III}, and λ^{IV}, respectively.

In the rest of the paper, we study the expected round duration λ for the four conditions on forgetting. Note that the condition in case III cannot be detected locally and thus does not allow for a distributed implementation. We rather use λ^{III} as a bound (cf. Equation (2)). 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, which we give in Section 4 in Theorems 1 and 2. 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 5.1. 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 5.2 and 5.3, 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,* $\lambda = \mathbb{E}\lim\limits_{t\to\infty} t/R_i(t) = \mathbb{E}\lim\limits_{r\to\infty} T_i(r)/r$.

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

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

4 Explicit Formulas for λ^{III} and λ^{IV}

In this section, by elementary probability theory and calculations, we derive explicit formulas for λ^{III} and λ^{IV} in Theorems 1 and 2, respectively. Both use a formula for the expected maximum of geometrically distributed random variables (Proposition 2). For that purpose define for pairwise independent with parameter p geometrically distributed random variables G_i

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

We will make use of the following well-known proposition [6, 14].

Proposition 2. $\Lambda(M,p) = \sum\limits_{i=1}^{M} \binom{M}{i} (-1)^i \dfrac{1}{(1-p)^i - 1}$

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 the 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 state explicit formulas for the expected round duration in cases III and IV. We will use these formulas in particular in Section 5.3 when studying the behavior of λ for $p \to 0$.

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

Proof. 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. □

Theorem 2. $\lambda^{\mathrm{IV}}(N,p) = \Lambda\big(N,p^{N-1}\big) = \displaystyle\sum_{i=1}^{N} \binom{N}{i} (-1)^i \frac{1}{(1-p^{N-1})^i - 1}$

Proof. 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. □

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., $a_{i,j}(t) = \max\big\{A_{i,j}(t) - \min_k A_{k,k}(t)\ ,\ -1\big\}$. 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 $A(t) = X + c$. 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 $\mathrm{Step}(\hat{a}(t)) = 1$ if there is a global round switch at time t, and 0 otherwise.

5.1 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.

Theorem 3. *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. [8, Theorem 17.0.1(i)]

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

$$\pi = e \cdot \big(P^{(n \to 1)} - I^{(n \to 0)}\big)^{-1} \tag{3}$$

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 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 $\hat{a}$.

Proposition 3. *For all conditions of forgetting, $R_i(t)/t \to 1/\lambda$ with probability 1 as $t \to \infty$. Furthermore, $\lambda = 1/\left(\sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \cdot \pi(\hat{a})\right)$.*

Proof. It holds that $R_G(t) = \sum_{k=1}^{t} \mathrm{Step}\left(\hat{a}(k)\right)$. By Theorem 3, 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 proposition.

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

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

5.2 Behavior of λ for $p \to 1$

The next theorem 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.

Theorem 4. *For all four conditions on forgetting, $\frac{d}{dp}\lambda(N,p)\big|_{p=1} = -N(N-1)$.*

Proof. 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 Proposition 3, $1/\lambda(N,p) = \sum_{\hat{a}} p^{\#_{-1}(\hat{a})} \cdot \pi_{N,p}(\hat{a})$. 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 $\lambda(N,1) = 1$. □

5.3 Behavior of λ for $p \to 0$

In systems with unreliable communication, in which Theorem 4 is not valuable, the following theorem on the asymptotic behavior of the expected round duration for all our conditions on forgetting, is useful. 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.

Theorem 5. *For $p \to 0$, $\lambda^{\mathrm{I}}(N,p)$, $\lambda^{\mathrm{II}}(N,p)$ and $\lambda^{\mathrm{III}}(N,p)$ are in $\Theta\big(p^{-1}\big)$, and $\lambda^{\mathrm{IV}}(N,p)$ is in $\Theta\big(p^{-(N-1)}\big)$.*

Proof. We first show the statement for λ^{III}. 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 Equation (3), all steady state probabilities $\pi(\hat{a})$ are rational functions in p. Proposition 3 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 asymptotic behavior of $\lambda^{\mathrm{IV}}(N,p)$, observe that by $(1-p)^i - 1 = -p\sum_{j=1}^{i}\binom{i}{j}(-p)^{j-1} \sim -p \cdot i$ for $p \to 0$ and by Proposition 2, we have

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

As shown in the textbook by Graham et al. [4, (6.72) and (6.73)] this sum equals H_M, denoting the Mth harmonic number. This concludes the proof. □

5.4 Lower Bounds on λ^{I} and λ^{II}

Determining the expected round duration for cases I and II by means of the Markov chain $a(t)$ is computationally intensive, even for small system sizes N. We can, however, compute efficient lower and upper bounds on $\lambda(N,p)$: For both, case 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 $\lambda^{\mathrm{II}}(N,p)$.

From Proposition 1 and Theorem 3 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 $\mathbb{E}(T_G(2) \mid \hat{a}(0) = \hat{a}) = \Lambda(\#_{-1}(\hat{a}), p)$.

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 $\hat{\pi}(n) = \sum_{\hat{a} \in [n]} \pi(\hat{a}) / (\sum_{\hat{a}} \mathrm{Step}(\hat{a}) \cdot \pi(\hat{a}))$. It follows that $\hat{\pi}(n) = 0$ for $n < 2N-2$ in case II and $\hat{\pi}(n) = 0$ for $n < N-1$ in case I.

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, $\hat{a}(t') \in [n]$. We obtain for $\hat{\pi}(n)$:

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

for c_n, d_n suitably chosen. One can derive valid choices for both parameters for cases I and II by excessive case inspection of transition probabilities for all state equivalence classes $[k]$, $k \geqslant 0$. We provide only a proof for case II in Section 5.5, as the proof for case I is by analogous arguments.

Partitioning the above sum into a one term from states in $[n]$ to states in $[n]$, and one remaining term, allows us to finally state inequality

$$\hat{\pi}(n) \geqslant \frac{d_n}{1 + d_n - c_n} =: \pi_n \ . \tag{4}$$

The resulting lower bounds on $\hat{\pi}(n)$, denoted by π_n^{I} and π_n^{II} for cases I and II respectively, finally yield lower bounds on λ. Since Λ is nondecreasing in its first argument, we can bound $\lambda(N, p)$ by

$$\left(1 - \sum_{n=N}^{N(N-1)} \pi_n^{\mathrm{I}}\right) \Lambda(N-1, p) + \sum_{n=N}^{N(N-1)} \pi_n^{\mathrm{I}} \Lambda(n, p) \leqslant \lambda^{\mathrm{I}}(N, p) \tag{5}$$

in case I. For case II we obtain

$$\left(1 - \sum_{n=2N-1}^{N(N-1)} \pi_n^{\mathrm{II}}\right) \Lambda(2N-2, p) + \sum_{n=2N-1}^{N(N-1)} \pi_n^{\mathrm{II}} \Lambda(n, p) \leqslant \lambda^{\mathrm{II}}(N, p) \ . \tag{6}$$

5.5 Lower Bound on Parameters for λ^{II}

We next show how to derive bounds on parameters c_n and d_n, in the following denoted by d_n^{II} and c_n^{II}. From these we obtain bounds on $\pi_{N(N-1)}$ from (4).

We start our analysis 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 $c^{\mathrm{II}}_{N(N-1)} = p^{N(N-1)}$, $d^{\mathrm{II}}_{N(N-1)} = p^{N(N-1)-1}$ 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 $\pi_{N(N-1)-1}$. Since it is not possible to make a direct transition from a state $\hat{a} \in \bigcup[n]$ to a state in $[N(N-1)-1]$, we consider bounds on the probability that the system is in a state within $[N(N-1)-1]$ at time $t+2$, given that $\hat{a}(t) = \hat{a}$. Fix in $\hat{a}$ one column j whose all non-diagonal entries equal -1. Clearly such a column must exist, since $\mathrm{Step}(\hat{a}) = 1$. 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 one message from process j to some fixed $j' \neq j$, are received. That is, $N(N-2) + 2 - \#_0(\hat{a})$ messages are received. Moreover, at time $t+1$, k (up to $N-3$) of the remaining $N-2$ message sent by j are received. By construction, $k+2$ of the processes are 1-processes at time $t+1$. For $\hat{a}(t+1) \in [N(N-1)-1]$ to hold, it is sufficient that: For all 0-processes i with $\hat{a}_{i,j}(t+1) = -1$, process i must receive a message from j at time $t+2$; exactly one of the messages from a 1-process to a 1-process is received. Since at time $t+1$ there are $(k+2)(k+1)$ messages from 1-processes to 1-processes, we obtain: For all $\hat{a} \in \bigcup[n]$,

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

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

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 analyzing the probability that $\hat{a}(t+2) \in [2(N-1)+x]$, given that $\hat{a}(t) = \hat{a}$. Fix a row j of $\hat{a}$ with T non-diagonal entries equal to 0. Given that $\hat{a}(t) = \hat{a}$, 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. That is, $(N-1)(N-1) - \#_0(\hat{a}) + T$ messages are

received. Moreover, at time $t+1$, k (up to $N-T-2$) of the remaining $N-T-1$ messages to j are received. Hence, all processes different from j are 1-processes at time $t+1$. At time $t+2$ all remaining messages to process j are received. 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 $\hat{a} \in \bigcup[n]$,

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

Note that γ is nonincreasing in its second and third 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

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

We 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 (4) and (6).

6 Discussion of Results

In this section we present the results obtained by calculating the expected round duration λ for the four conditions on forgetting that we consider. Additionally, we used Monte-Carlo simulations to estimate λ and the average age of data when a process performs a round switch.

Fig. 3 shows, with varying probability p, the exact value of the expected round duration for conditions on forgetting I–IV in a system with $N = 3$ processes. As stated in Section 5.3, 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, as depicted in Section 5.2, all the plots have the same slope in the point $p = 1$ resulting in a good approximation for the hard to calculate cases I and II in a system with reliable communication.

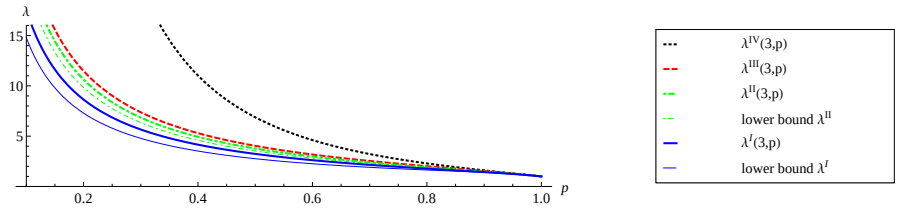


Fig. 3. Expected round durations for $N = 3$ and lower bounds for cases I and II.

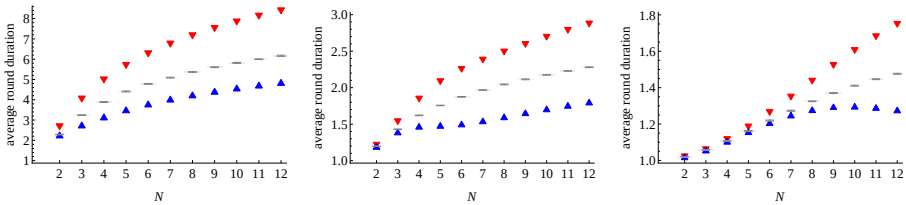
In settings with unreliable communication, for which the approximation result on the derivative of λ at $p = 1$ is not valuable, cases I and II can be approximated by their analytical lower bounds (Section 5.4), and bounded from above by the λ for case III (Theorem 1). A comparison between the lower bounds and the actual systems is illustrated in Fig. 3.

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 calculations. 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. Fig. 4 and 5 show the obtained 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. 4(a) to 4(c) and the case II (where processes forget after a local round switch) is shown in Fig. 5(a) to 5(c). Fig. 6(a) to 6(c) depict 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. 7 shows 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.

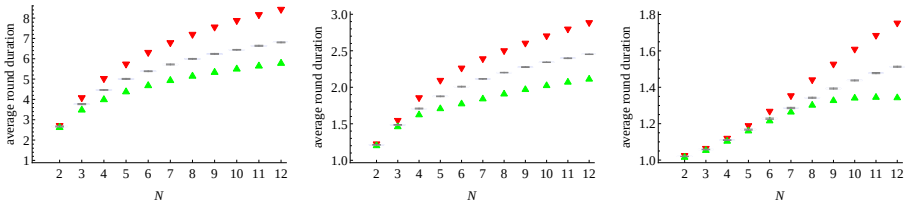
7 Conclusion

We studied the effect of actively discarding memory content on a variant of the α-synchronizer. For practically relevant applications, e.g., in sensor fusion, forgetting turns out to be a simple strategy to decrease the average age of data handed over to the application layer when a process makes a round switch. In case the accuracy of data degrades over time, e.g., data samples taken from a timed process, a decreased average age of the samples results in an increased accuracy of the merged data. To assess the inevitable drawback of forgetting strategies, namely degraded performance, we analyzed four naturally chosen strategies of



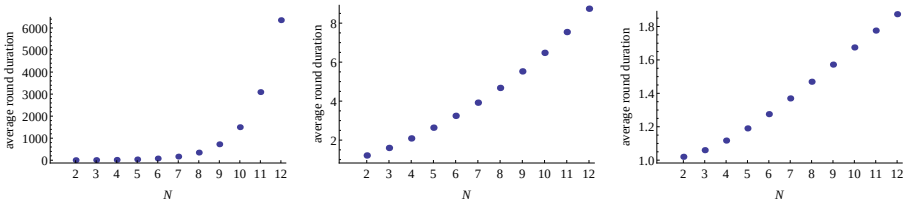
(a) $p = 0.5$ (b) $p = 0.9$ (c) $p = 0.99$

Fig. 4. Monte-Carlo simulation results for case I compared against the calculated lower bound and the calculated expected round duration of case III serving as an upper bound.



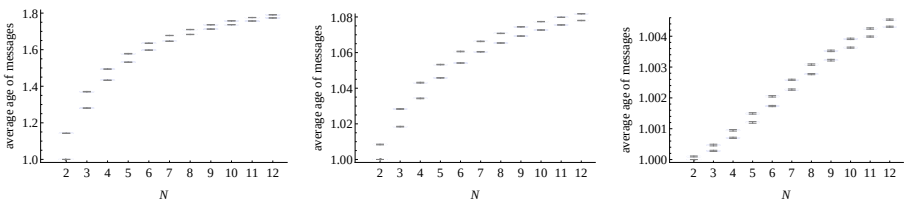
(a) $p = 0.5$ (b) $p = 0.9$ (c) $p = 0.99$

Fig. 5. Monte-Carlo simulation results for case II 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. 6. Calculated expected round duration of case IV.



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

Fig. 7. Average age Monte-Carlo simulation results for case I (blue, upper) and II (green, lower).

forgetting. We obtained analytic 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, allowing to assess whether the resulting loss of performance is acceptable for a specific application. Interestingly, it turned out that the behavior of all four variants is similar for $p \to 1$. For $p \to 0$ only two asymptotic behaviors of the expected round duration were observed: $\Theta(1/p^{N-1})$ for the significantly slower variant that forgets at each time step, and $\Theta(1/p)$ for the other variants.

References

1. Awerbuch, B.: Complexity of Network Synchronization. J. ACM 32, 804–823 (1985)
2. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice Hall, Englewood Cliffs (1989)
3. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIT Press, Cambridge, MA (1995)
4. Graham, R.L., Knuth, D.E., Patashnik, O.: Concrete Mathematics. Addison-Wesley, Reading, MA (1989)
5. Jayaram, M., Varghese, G.: Crash Failures Can Drive Protocols to Arbitrary States. In: 15th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 247–256. ACM, New York (1996)
6. Kirschenhofer, P., Prodinger, H.: A result in order statistics related to probabilistic counting. Computing 51 (1), 15–27 (1993)
7. Marzullo, K.A.: Tolerating Failures of Continuous-Valued Sensors. ACM Trans. on Computer Systems 8 (4), 284–304 (1990)
8. Meyn, S., Tweedie, R.L.: Markov Chains and Stochastic Stability. Springer, Heidelberg (1993)
9. Nakamura, E.F., Loureiro, A.A.F., Frery, A.C.: Information fusion for wireless sensor networks: Methods, models, and classifications. ACM Comput. Surv. 39 (3). ACM, New York (2007). doi:10.1145/1267070.1267073
10. Nowak, T., Függer, M., Kößler, A.: On the Performance of a Retransmission-Based Synchronizer. Theor. Comput. Sci. (2012). doi:10.1016/j.tcs.2012.04.035
11. Rajsbaum, S.: Upper and Lower Bounds for Stochastic Marked Graphs. Inform. Process. Lett. 49, 291–295 (1994)
12. Rajsbaum, S., Sidi, M.: On the Performance of Synchronized Programs in Distributed Networks with Random Processing Times and Transmission Delays. IEEE T. Parall. Distr. 5, 939–950 (1994)
13. Schmid, U., Schossmaier, K.: How to Reconcile Fault-Tolerant Interval Intersection with the Lipschitz Condition. Dis. Comp. 14 (2), 101–111 (2001)
14. Szpankowski, W., Rego, V.: Yet another application of a binomial recurrence. Order statistics. Computing 43 (4), 401–410 (1990)