On the Performance of a Retransmission-Based Synchronizer

Thomas Nowak, Matthias Függer, Alexander Kößler

On the Performance of a Retransmission-based Synchronizer*

Thomas Nowak[1], Matthias Függer[2], and Alexander Kößler[2]

[1] LIX, Ecole polytechnique, Palaiseau, France
nowak@lix.polytechnique.fr
[2] ECS Group, TU Wien, Vienna, Austria
{fuegger,koe}@ecs.tuwien.ac.at

Abstract. Designing algorithms for distributed systems that provide a round abstraction is often simpler than designing for those that do not provide such an abstraction. However, distributed systems need to tolerate various kinds of failures. The concept of a synchronizer deals with both: It constructs rounds and allows masking of transmission failures. One simple way of dealing with transmission failures is to retransmit a message until it is known that the message was successfully received. We calculate the *exact value* of the average rate of a retransmission-based synchronizer in an environment with probabilistic message loss, within which the synchronizer shows nontrivial timing behavior. The theoretic results, based on Markov theory, are backed up with Monte Carlo simulations.

1 Introduction

Analyzing the time-complexity of an algorithm is at the core of computer science. Classically this is carried out by counting the number of steps executed by a Turing machine. In distributed computing [12, 1], local computations are typically viewed as being completed in zero time, focusing on communication delays only. This view is useful for algorithms that communicate heavily, with only a few local operations of negligible duration between two communications.

In this work we are focusing on the implementation of an important subset of distributed algorithms where communication and computation are highly structured, namely *round based algorithms* [2, 4, 8, 17]: Each process performs its computations in consecutive rounds. Thereby a single *round* consists of (1) the processes exchanging data with each other and (2) each process executing local computations. Call the number of rounds it takes to complete a task the round-complexity.

We consider repeated instances of a problem, i.e., a problem is repeatedly solved during an infinite execution. Such problems arise when the distributed system under consideration provides a continuous service to the top-level application, e.g., repeatedly solves distributed consensus [11] in the context of

*This research was partially supported by grants P21694 and P20529 of the Austrian Science Fund (FWF).

state-machine replication. A natural performance measure for these systems is the average number of problem instances solved per round during an execution. In case a single problem instance has a round-complexity of a constant number $R \geqslant 1$ of rounds, we readily obtain a rate of $1/R$.

If we are interested in time-complexity in terms of Newtonian real-time, we can scale the round-complexity with the duration (bounds) of a round, yielding a real-time rate of $1/RT$, if T is the duration of a single round. Note that the attainable accuracy of the calculated real-time rate thus heavily relies on the ability to obtain a good measurement of T. In case the data exchange within a single round comprises each process broadcasting a message and receiving messages from all other processes, T can be related to message latency and local computation upper and lower bounds, typically yielding precise bounds for the round duration T. However, there are interesting distributed systems where T cannot be easily related to message delays: consider, for example, a distributed system that faces the problem of message loss, and where it might happen that processes have to resend messages several times before they are correctly received, and the next round can be started. It is exactly these nontrivial systems the determination of whose round duration T is the scope of this paper.

We claim to make the following contributions in this paper: (1) We give an algorithmic way to determine the expected round duration of a general retransmission scheme, thereby generalizing results concerning stochastic max-plus systems by Resing *et al.* [18]. (2) We present simulation results providing (a) deeper insights in the convergence behavior of round duration times and indicating that (b) the error we make when restricting ourselves to having a maximum number of retransmissions is small. (3) We present nontrivial theoretical bounds on the convergence speed of round durations to the expected round duration.

Section 2 introduces the retransmission scheme in question and the probabilistic environment in which the round duration is investigated, and reduces the calculation of the expected round duration to the study of a certain random process. Section 3 provides a way to compute the asymptotically expected round duration λ, and also presents theoretical bounds on the convergence speed of round durations to λ. Section 4 contains simulation results. We give an overview on related work in Section 5.

An extended version of this paper, containing detailed proofs, appeared as a technical report [15].

2 Retransmitting under Probabilistic Message Loss

Simulations that provide stronger communication directives on top of a system satisfying weaker communication directives are commonly used in distributed computing [9, 8]. In this section we present one such simulation—a retransmission scheme. The proposed retransmission scheme is a modified version of the α synchronizer [2].

We assume a fully-connected network of processes $1, 2, \ldots, N$. Given an algorithm B designed to work in a failure-free round model, we construct algorithm

$A(B)$, simulating B on top of a model with transient message loss. The idea of the simulation is simple: Algorithm $A(B)$ retransmits B's messages until it is known that they have been successfully received by all processes.

Explicitly, each process periodically, in each of its steps, broadcasts (1) its current (simulated) round number Rnd, (2) algorithm B's message for the current round (Rnd), and (3) algorithm B's message for the previous round ($Rnd - 1$). A process remains in simulated round Rnd until it has received all other processes' round Rnd messages. When it has, it advances to simulated round $Rnd+1$.

In an execution of algorithm $A(B)$, see Figure 1, we define the *start of simulated round* r at process i, denoted by $T_i(r)$, to be the number of the step in which process i advances to simulated round r. We assume $T_i(1) = 1$. Furthermore, define $L(r)$ to be the number of the step in which the last process advances to simulated round r, i.e., $L(r) = \max_i T_i(r)$. The *duration of simulated round* r at process i is $T_i(r+1) - T_i(r)$, that is, we measure the round duration in the number of steps taken by a process.

Define the *effective transmission delay* $\delta_{j,i}(r)$ to be the number of tries until process j's simulated round r message is successfully received for the first time by process i.[3] We obtain the following equation relating the starts of the simulated rounds:

$$T_i(r+1) = \max_{1 \leqslant j \leqslant N} \big(T_j(r) + \delta_{j,i}(r)\big) \tag{1}$$

Figure 1 depicts part of an execution of $A(B)$. Messages from process i to itself are not depicted as they can be assumed to be received in the next step.

To allow for a quantitative assessment of the durations of the simulated rounds, we extend the environment using a probability space. Let **ProbLoss**(p) be the following probability distribution: The random variables $\delta_{j,i}(r)$ are pairwise independent, and for any two processes $i \neq j$, the probability that $\delta_{j,i}(r) = z$ is equal to $(1-p)^{z-1} \cdot p$, i.e., using p as the probability of a successful message

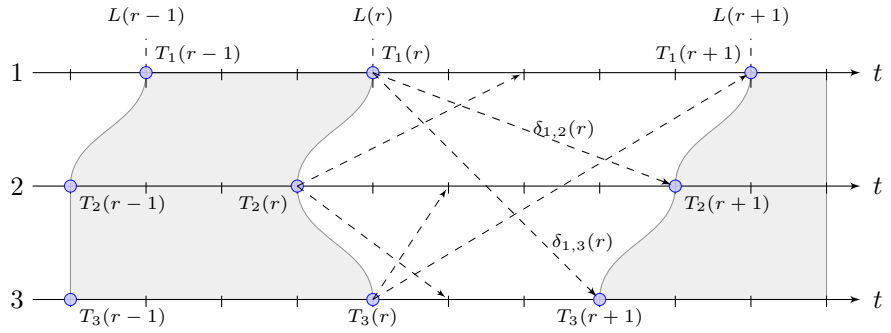


Fig. 1. An execution of $A(B)$

[3] Formally, for any two processes i and j, let $\delta_{j,i}(r) - 1$ be the smallest number $\ell \geqslant 0$ such that (1) process j sends a message m in its $(T_j(r) + \ell)^{\text{th}}$ step and (2) process i receives m from j in its $(T_j(r) + \ell + 1)^{\text{th}}$ step.

transmission, the first $z-1$ tries of sending j's round r message to i failed and the z^{th} try was successful. Note that we can assume $\delta_{i,i}(r) = 1$.

For computational purposes we further introduce the probability distribution $\mathbf{ProbLoss}(p, M)$, where $M \in \mathbb{N} \cup \{\infty\}$, which is a variant of $\mathbf{ProbLoss}(p)$ where the number of tries per simulated round message until it is successfully received is bounded by M. Call M the *maximum number of tries per round*. Variable $\delta_{j,i}(r)$ can take values in the set $\{z \in \mathbb{N} \mid 1 \leqslant z \leqslant M\}$. For any two processes $i \neq j$, and for integers z with $1 \leqslant z < M$, the probability that $\delta_{j,i}(r) = z$ is $(1-p)^{z-1} \cdot p$. In the remaining cases, i.e., with probability $(1-p)^{M-1}$, $\delta_{j,i}(r) = M$. If $M = \infty$, this case vanishes. In particular, $\mathbf{ProbLoss}(p, \infty) = \mathbf{ProbLoss}(p)$.

We will see in Sections 3.3 and 4, that the error we make when calculating the expected duration of the simulated rounds in $\mathbf{ProbLoss}(p, M)$ with finite M instead of $\mathbf{ProbLoss}(p)$ is small, even for small values of M.

3 Calculating the Expected Round Duration

The expected round duration of the presented retransmission scheme, in the case of $\mathbf{ProbLoss}(p, M)$, is determined by introducing an appropriate Markov chain, and analyzing its steady state. To this end, we define a Markov chain $\Lambda(r)$, for an arbitrary round $r \geqslant 1$, that (1) captures enough of the dynamics of round construction to determine the round durations and (2) is simple enough to allow efficient computation of each of the process i's *expected round duration* λ_i, defined by $\lambda_i = \mathbb{E} \lim_r T_i(r)/r$.

Since for each process i and $r \geqslant 2$, it holds that $L(r-1) \leqslant T_i(r) \leqslant L(r)$, we obtain the following equivalence:

Proposition 1. *If $L(r)/r$ converges, then* $\lim_{r\to\infty} T_i(r)/r = \lim_{r\to\infty} L(r)/r$.

We can thus reduce the study of the processes' average round durations to the study of the sequence $L(r)/r$ as $r \to \infty$. In particular, for any two processes i, j it holds that $\lambda_i = \lambda_j = \lambda$, where $\lambda = \mathbb{E} \lim_r L(r)/r$.

3.1 Round Durations as a Markov Chain

A *Markov chain* is a discrete-time stochastic process $X(r)$ in which the probability distribution for $X(r+1)$ only depends on the value of $X(r)$. We denote the transition probability from state Y to state X by $P_{X,Y}$.

A Markov chain that, by definition, fully captures the dynamics of the round durations is $T(r)$, where $T(r)$ is defined to be the collection of local round finishing times $T_i(r)$ from Equation (1). However, directly using Markov chain $T(r)$ for the calculation of λ is infeasible since $T_i(r)$, for each process i, grows without bound in r, and thereby its state space is infinite. For this reason we introduce Markov chain $\Lambda(r)$ which optimizes $T(r)$ in two ways and which we use to compute λ: One can achieve a finite state space by considering differences of $T(r)$, instead of $T(r)$. This is one optimization we built into $\Lambda(r)$ and only by it are we enabled to use the computer to calculate the expected round duration. The

other optimization in $\Lambda(r)$, which is orthogonal to the first one, is that we do not record the local round finishing times (resp. the difference of local round finishing times) for every of the N processes, but only record the *number* of processes that are associated a given value. This reduces the size of the state space from M^N to $\binom{N+M-1}{M-1}$, which is significant, because in practical situations, it suffices to use modest values of M as will be shown in Section 4.

We are now ready to define $\Lambda(r)$. Its state space $\mathcal{L}$ is defined to be the set of M-tuples $(\sigma_1, \ldots, \sigma_M)$ of nonnegative integers such that $\sum_{z=1}^{M} \sigma_z = N$. The M-tuples from $\mathcal{L}$ are related to $T(r)$ as follows: Let $\#X$ be the cardinality of set X, and define

$$\sigma_z(r) = \#\big\{i \mid T_i(r) - L(r-1) = z\big\} \tag{2}$$

for $r \geqslant 1$, where we set $L(0) = 0$ to make the case $r = 1$ in (2) well-defined. Note that $T_i(r) - L(r-1)$ is always greater than 0, because $\delta_{j,i}(r)$ in Equation (1) is greater than 0. Finally, set

$$\Lambda(r) = \big(\sigma_1(r), \ldots, \sigma_M(r)\big) \ . \tag{3}$$

The intuition for $\Lambda(r)$ is as follows: For each z, $\sigma_z(r)$ captures the number of processes that start simulated round r, z steps after the last process started the last simulated round, namely $r-1$. For example, in case of the execution depicted in Figure 1, $\sigma_1(r) = 0$, $\sigma_2(r) = 1$ and $\sigma_3(r) = 2$. Since algorithm $A(B)$ always waits for the last simulated round message received, and the maximum number of tries until the message is correctly received is bounded by M, we obtain that $\sigma_z(r) = 0$ for $z < 1$ and $z > M$. Knowing $\sigma_z(r)$, for each z with $1 \leqslant z \leqslant M$, thus provides sufficient information (1) on the processes' states in order to calculate the probability of the next state $\Lambda(r+1) = (\sigma_1, \ldots, \sigma_M)$, and (2) to determine $L(r+1) - L(r)$ and by this the simulated round duration for the last process. We first obtain:

Proposition 2. *$\Lambda(r)$ is a Markov chain.*

In fact Proposition 2 even holds for a wider class of delay distributions $\delta_{j,i}(r)$; namely those invariant under permutation of processes. Likewise, many results in the remainder of this section are applicable to a wider class of delay distributions: For example, we might lift the independence restriction on the $\delta_{j,i}(r)$ for fixed r and assume strong correlation between the delays, i.e., for each process j and each round r, $\delta_{j,i}(r) = \delta_{j,i'}(r)$ for any two processes i, i'.[4]

Let $X(r)$ be a Markov chain with countable state space $\mathcal{X}$ and transition probability distribution P. Further, let π be a probability distribution on $\mathcal{X}$. We call π a *stationary distribution* for $X(r)$ if $\pi(X) = \sum_{Y \in \mathcal{X}} P_{X,Y} \cdot \pi(Y)$ for all $X \in \mathcal{X}$. Intuitively, $\pi(X)$ is the asymptotic relative amount of time in which Markov chain $X(r)$ is in state X.

Definition 1. *Call a Markov chain* good *if it is aperiodic, irreducible, Harris recurrent, and has a unique stationary distribution.*

[4] Rajsbaum and Sidi [17] call this "negligible transmission delays".

Proposition 3. *$\Lambda(r)$ is a good Markov chain.*

Denote by π the unique stationary distribution of $\Lambda(r)$, which exists because of Proposition 3. Define the function $\sigma : \mathcal{L} \to \mathbb{R}$ by setting $\sigma(\Lambda) = \max\{z \mid \sigma_z \neq 0\}$ where $\Lambda = (\sigma_1, \ldots, \sigma_M) \in \mathcal{L}$. By abuse of notation, we write $\sigma(r)$ instead of $\sigma\big(\Lambda(r)\big)$. From the next proposition follows that $\sigma(r) = L(r) - L(r-1)$, i.e., $\sigma(r)$ is the last process' duration of simulated round $r-1$. For example $\sigma(r+1) = 5$ in the execution in Figure 1.

Proposition 4. $L(r) = \sum_{k=1}^{r} \sigma(k)$

The following theorem is key for calculating the expected simulated round duration λ. We will use the theorem for the computation of λ starting in Section 3.2. It states that the simulated round duration averages $L(r)/r$ up to some round r converge to a finite λ almost surely as r goes to infinity. This holds even for $M = \infty$, that is, if no bound is assumed on the number of tries until successful reception of a message. The theorem further relates λ to the steady state π of $\Lambda(r)$. Let $\mathcal{L}_z \subseteq \mathcal{L}$ denote the set of states Λ such that $\sigma(\Lambda) = z$.

Theorem 1. *$L(r)/r \to \lambda$ with probability 1. It is $\lambda = \sum_{z=1}^{M} z \cdot \pi(\mathcal{L}_z) < \infty$.*

3.2 Using $\Lambda(r)$ to Compute λ

We now state an algorithm that, given parameters $M \neq \infty$, N, and p, computes the expected simulated round duration λ (see Theorem 1). In its core is a standard procedure to compute the stationary distribution of a Markov chain, in form of a matrix inversion. In order to utilize this standard procedure, we need to explicitly state the transition probability distributions P_{XY}, which we regard as a matrix P. For ease of exposition we state P for the system of processes with probabilistic loop-back links, i.e., we do not assume that $\delta_{i,i}(r) = 1$ holds. Later, we explain how to arrive at a formula for P in the case of the (more realistic) assumption of $\delta_{i,i}(r) = 1$.

A first observation yields that matrix P bears some symmetry, and thus some of the matrix' entries can be reduced to others. In fact we first consider the transition probability from *normalized* Λ states only, that is, $\Lambda = (\sigma_1, \ldots, \sigma_M)$ with $\sigma_M \neq 0$.

In a second step we observe that a non-normalized state Λ can be transformed to a normalized state $\Lambda' = \mathrm{Norm}(\Lambda)$ without changing its outgoing transition probabilities, i.e., for any state X in $\mathcal{L}$, it holds that $P_{X,\Lambda} = P_{X,\Lambda'}$: Thereby Norm is the function $\mathcal{L} \to \mathcal{L}$ defined by:

$$\mathrm{Norm}(\sigma_1, \ldots, \sigma_M) = \begin{cases} (\sigma_1, \ldots, \sigma_M) & \text{if } \sigma_M \neq 0 \\ \mathrm{Norm}(0, \sigma_1, \ldots, \sigma_{M-1}) & \text{otherwise} \end{cases}$$

For example, assuming that $M = 5$, and considering the execution in Figure 1, it holds that $\Lambda(r) = (0, 1, 2, 0, 0)$. Normalization, that is, right alignment of the last processes, yields $\mathrm{Norm}(\Lambda(r)) = (0, 0, 0, 1, 2)$.

Further, for any $\Lambda = (\sigma_1, \ldots, \sigma_M)$ in $\mathcal{L}$ with $\sigma_M \neq 0$, and any $1 \leqslant z \leqslant M$, let $P(\leqslant z \mid \Lambda)$ be the conditional probability that a specific process i is in the set $\{i \mid T_i(r+1) - L(r) \leqslant z\}$, given that $\Lambda(r) = \Lambda$. We easily observe that i is in the set if and only if all the following M conditions are fulfilled: for each u, $1 \leqslant u \leqslant M$: for *all* processes j for which $T_j(r) - L(r-1) = u$ (this holds for $\sigma_u(r)$ many) it holds that $\delta_{j,i}(r) \leqslant z + M - u$. Therefore we obtain:

$$P(\leqslant z \mid \Lambda(r)) = \prod_{1 \leqslant u \leqslant M} P(\delta \leqslant z + M - u)^{\sigma_u(r)} \ , \tag{4}$$

for all z, $1 \leqslant z \leqslant M$. Let $P(z \mid \Lambda)$ be the conditional probability that process i is in the set $\{i \mid T_i(r+1) - L(r) = z\}$, given that $\Lambda(r) = \Lambda$. From Equation (4), we immediately obtain:

$$\begin{aligned} &P(1 \mid \Lambda) = P(\leqslant 1 \mid \Lambda) \text{ and} \\ &P(z \mid \Lambda) = P(\leqslant z \mid \Lambda) - P(\leqslant z-1 \mid \Lambda) \ , \end{aligned} \tag{5}$$

for all z, $1 < z \leqslant M$. We may finally state the transition matrix P: for each $X, Y \in \mathcal{L}$, with $X = (\sigma_1, \ldots, \sigma_M)$ and $Y = (\sigma'_1, \ldots, \sigma'_M)$,

$$P_{XY} = \prod_{1 \leqslant z \leqslant M} \binom{N - \sum_{k=1}^{z-1} \sigma_k}{\sigma_z} P(z \mid \mathrm{Norm}(Y))^{\sigma_z} \ . \tag{6}$$

Note that for a system where $\delta_{i,i}(r) = 1$ holds, in Equation (4), one has the account for the fact that a process i definitely receives its own message after 1 step. In order to specify a transition probability analogous to Equation (4), it is thus necessary to know to which of the $\sigma_k(r)$ in $\Lambda(r)$, process i did count for, that is, for which k, $T_i(r) - L(r-1) = k$ holds. We then replace $\sigma_k(r)$ by $\sigma_k(r) - 1$, and keep $\sigma_u(r)$ for $u \neq k$. Formally, let $P(\leqslant z \mid \Lambda, k)$, with $1 \leqslant k \leqslant M$, be the conditional probability that process i is in the set $\{i \mid T_i(r+1) - L(r) \leqslant z\}$, given that $\Lambda(r) = \Lambda$, as well as $T_i(r) - L(r-1) = k$. Then:

$$P(\leqslant z \mid \Lambda(r), k) = \prod_{1 \leqslant u \leqslant M} P(\delta \leqslant z + M - u)^{\sigma_u(r) - \mathbf{1}_{\{k\}}(u)}$$

where $\mathbf{1}_{\{k\}}(u)$ is the indicator function, having value 1 for $u = k$ and 0 otherwise. Equation (5) can be generalized in a straightforward manner to obtain expressions for $P(z \mid \Lambda, k)$.

The dependency of $P(\leqslant z \mid \Lambda(r), k)$ on k is finally accounted for in Equation (6), by additionally considering all possible choices of processes whose sum makes up σ_z.

Let $\Lambda_1, \Lambda_2, \ldots, \Lambda_n$ be any enumeration of states in $\mathcal{L}$. We write $P_{ij} = P_{\Lambda_i \Lambda_j}$ and $\pi_i = \pi(\Lambda_i)$ to view P as an $n \times n$ matrix and π as a column vector. By definition, the unique stationary distribution π satisfies (1) $\pi = P \cdot \pi$, (2) $\sum_i \pi_i = 1$, and (3) $\pi_i \geqslant 0$. It is an elementary linear algebraic fact that these properties suffice to characterize π by the following formula:

$$\pi = \left(P^{(n \to 1)} - I^{(n \to 0)}\right)^{-1} \cdot e \tag{7}$$

where $e = (0, \ldots, 0, 1)^T$, $P^{(n \to 1)}$ is matrix P with its entries in the n^{th} row set to 1, and $I^{(n \to 0)}$ is the identity matrix with its entries in the n^{th} row set to zero.

After calculating π, we can use Theorem 1 to finally determine the expected simulated round duration λ. The time complexity of this approach is determined by the matrix inversion of P. Its time complexity is within $O(n^3)$, where n is the number of states in the Markov chain $\Lambda(r)$. Since the state space is given by the set of M-tuples whose entries are within $\{1, \ldots, M\}$ and whose sum is N, we obtain $n = \binom{N+M-1}{M-1}$. In Sections 3.4 and 4 we show that already small values of M yield good approximations of λ, that quickly converge with growing M. This leads to a tractable time complexity of the proposed algorithm.

3.3 Results

The presented algorithm allows to obtain analytic expressions for λ for fixed N and M in terms of probability p. Figure 2 contains the expressions of $\lambda(p, N)$ for $M = 2$ and N equal to 2 and 3, respectively. For larger M and N, the expressions already become significantly longer.

Figures 3(a) and 3(b) show solutions of $\lambda(p)$ for systems with $N = 2$ and $N = 4$, respectively. We observe that for high values of the probability of successful communication p, systems with different M have approximately same slope. Since real distributed systems typically have a high p value, we may approximate λ for higher M values with that of significantly lower M values. The effect is further investigated in Section 4 by means of Monte Carlo simulation.

3.4 Rate of Convergence

Theorem 1 states that $L(r)/r$ converges to λ with probability 1, however it does not give a rate of convergence. We now present a lower bound on the speed of this convergence.

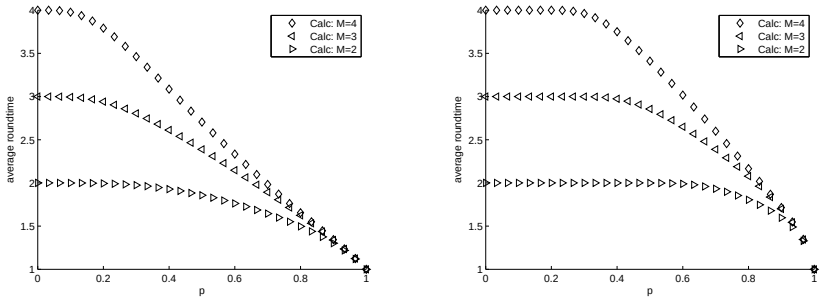
The fundamental facts regarding the convergence speed of $L(r)/r$ are: (1) The expected value of $L(r)/r$ is $\lambda + O(r^{-1})$ as $r \to \infty$. (2) The variance of $L(r)/r$ converges to zero; more precisely, it is $O(r^{-1})$ as $r \to \infty$. Chebyshev's inequality provides a way of utilizing these two facts, and yields the following corollary. It bounds the probability for the event $|L(r)/r - \lambda| \geqslant A$, where A is a positive real number. (A more general statement is [15, Theorem 5].)

Corollary 1. *For all $A > 0$, the probability that $|L(r)/r - \lambda| \geqslant A$ is $O\big(r^{-2}\big)$.*

$$\lambda(p,2) = \frac{6 - 6p + p^2}{3 - 2p}$$

$$\lambda(p,3) = \frac{2 - 8p + 18p^2 - 16p^3 + 12p^4 + 24p^5 - 64p^6 + 22p^7 + 30p^8 - 22p^9 + 3p^{10}}{1 - 4p + 9p^2 - 8p^3 + 6p^4 + 12p^5 - 27p^6 + 6p^7 + 12p^8 - 6p^9}$$

Fig. 2. Expressions for $\lambda(p, 2)$ and $\lambda(p, 3)$ in a system with $M = 2$



(a) $N = 2$ (b) $N = 4$

Fig. 3. λ versus p in a system with different choices of N

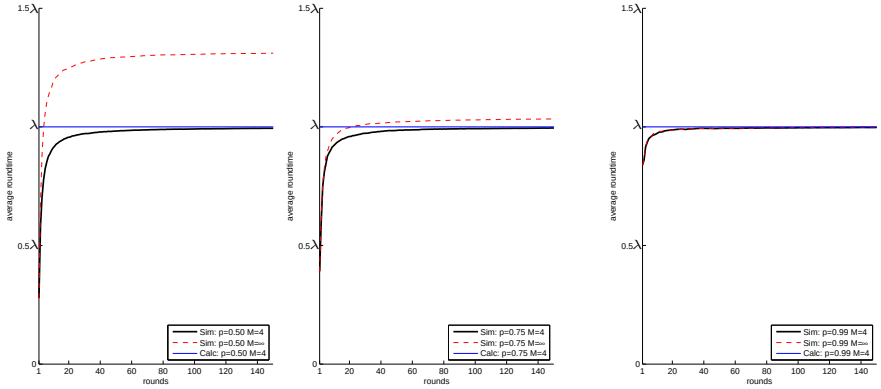
4 Simulations

In this section we study the applicability of the results obtained in the previous section to calculate the expected round duration the simulating algorithm in a distributed system with N processes in a p-lossy environment. The algorithm presented in Section 3.2, however, only yields results for $M < \infty$. Therefore, the question arises whether the solutions for finite M yield (close) approximations for $M = \infty$. Hence, we study the behavior of the random process $T(r)/r$ for increasing r, for different M, with Monte Carlo simulations carried out in Matlab.

We considered the behavior of a system of $N = 5$ processes, for different parameters M and p. The results of the simulation are plotted in Figures 4(a)–4(c). They show: (1) The expected round duration λ, computed by the algorithm presented in Section 3.2 for a system with $M = 4$, drawn as a constant function. (2) The simulation results of sequence $T_1(r)/r$, that is process 1's round starts, relative to the calculated λ, for rounds $1 \leqslant r \leqslant 150$, for two systems: one with parameter $M = 4$, the other with parameter $M = \infty$, averaged over 500 runs.

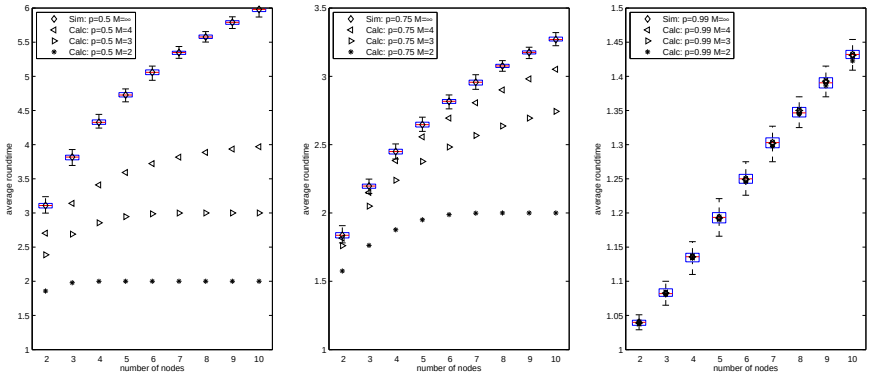
In all three cases, it can be observed that the simulated sequence with parameter $M = 4$ rapidly approximates the theoretically predicted rate for $M = 4$. From the figures we further conclude that calculation of the expected simulated round duration λ for a system with finite, and even small, M already yields good approximations of the expected rate of a system with $M = \infty$ for $p > 0.75$, while for practically relevant $p \geqslant 0.99$ one cannot distinguish the finite from the infinite case.

To further support this claim, we compared analytically obtained λ values for several settings of parameters p, N, and small M to the rates obtained from 100 Monte-Carlo simulation runs each lasting for 1000 rounds of the corresponding systems with $M = \infty$: The resulting Figures 5(a)–5(c) visualizes this comparison: the figures show the dependency of λ on the number of processes N, and present the statistical data from the simulations as boxplots. Note that for $p = 0.75$ the discrepancy between the analytic results for $M = 4$ and the simulation results for $M = \infty$ is already small, and for $p = 0.99$ the analytic results for all choices of M are placed inbetween the lower the upper quartile of the simulation results.



(a) $p = 0.5$ (b) $p = 0.75$ (c) $p = 0.99$

Fig. 4. $T_1(r)/r$ versus r in systems with different p



(a) $p = 0.5$ (b) $p = 0.75$ (c) $p = 0.99$

Fig. 5. λ versus N in systems with different p

5 Related Work

The notion of simulating a stronger system on top of a weaker one is common in the field of distributed computing [1, Part II]. For instance, Neiger and Toueg provide automatic translation technique that turns a synchronous algorithm B that tolerates benign failures into an algorithm $A(B)$ that tolerate more severe failures. Dwork, Lynch, and Stockmeyer [9] use the simulation of a round structure on top of a partially synchronous system, and Charron-Bost and Schiper [8] systematically study simulations of stronger communication axioms in the context of round-based models.

In contrast to randomized algorithms, like Ben-Or's consensus algorithm [5], the notion of a probabilistic *environment*, as we use it, is less common in distributed computing: One of the few exceptions is Bakr and Keidar [4] who provide practical performance results on distributed algorithms running on the Internet. On the theoretical side, Bracha and Toueg [7] consider the Consensus Problem in an environment, for which they assume a nonzero lower bound on the probability that a message m sent from process i to j in round r is correctly received, and that the correct reception of m is independent from the correct reception of a message from i to some process $j' \neq j$ in the same round r. While we, too, assume independence of correct receptions, we additionally assume a constant probability $p > 0$ of correct transmission, allowing us to derive exact values for the expected round durations of the presented retransmission scheme, which was shown to provide perfect rounds on top of fair-lossy executions. The presented retransmission scheme is based on the α-synchronizer introduced by Awerbuch [2] together with correctness proofs for asynchronous (non-faulty) communication networks of arbitrary structure. However, since Awerbuch did not assume a probability distribution on the message receptions, only trivial bounds on the performance could be stated. Rajsbaum and Sidi [17] extended Awerbuch's analysis by assuming message delays to be negligible, and processes' processing times to be distributed. They consider (1) the general case as well as (2) exponential distribution, and derive performance bounds for (1) and exact values for (2). In terms of our model their assumption translates to assuming maximum positive correlation between message delays: For each (sender) process j and round r, $\delta_{j,i}(r) = \delta_{j,i'}(r)$ for any two (receiver) processes i, i'. They then generalize their approach to the case where $\delta_{j,i}(r)$ comprises a dependent (the processing time) and an independent part (the message delay), and show how to adapt the performance bounds for this case. However, only bounds and no exact performance values are derived for this case. Rajsbaum [16] presented bounds for the case of identical exponential distribution of transmission delays and processing times. Bertsekas and Tsitsiklis [6] state bounds for the case of constant processing times and independently, exponentially distributed message delays. However, again, no exact performance values were derived.

Our model comprises negligible processing times and transmission faults, which result in a discrete distribution of the effective transmission delays $\delta_{j,i}(r)$. Interestingly, with one sole exception [18] which considers the case of a 2-processor system only, we did not find any published results on exact values of the expected round durations in this case. The nontriviality of this problem is indicated by the fact that finding the expected round duration is equivalent to finding the exact value of the *Lyapunov exponent* of a nontrivial stochastic max-plus system [10], which is known to be hard problem (e.g., [3]). In particular, our results can be translated into novel results on stochastic max-plus systems.

Acknowledgements The authors would like to thank Martin Biely, Ulrich Schmid, and Martin Zeiner for helpful discussions. The computational results presented have been achieved in part using the Vienna Scientific Cluster (VSC).

References

1. Attiya, H., Welch, J.: Distributed Computing: Fundamentals, Simulations, and Advanced Topics. Second edition. John Wiley & Sons (2004)
2. Awerbuch, B.: Complexity of Network Synchronization. J. ACM 32, 804–823 (1985)
3. Baccelli, F., Hong, D.: Analytic Expansions of Max-Plus Lyapunov Exponents. Ann. Appl. Probab. 10, 779–827 (2000)
4. Bakr, O., Keidar, I.: Evaluating the Running Time of a Communication Round over the Internet. In: 21st Annual ACM Symposium on Principles of Distributed Computing. ACM (2002)
5. Ben-Or, M.: Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols. In: 2nd Annual ACM Symposium on Principles of Distributed Computing. ACM (1983)
6. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice Hall (1989)
7. Bracha, G., Toueg, S.: Asynchronous Consensus and Broadcast Protocols. J. ACM 32, 824–840 (1985)
8. Charron-Bost, B., Schiper, A.: The Heard-Of Model: Computing in Distributed Systems with Benign Faults. Distrib. Comput. 22, 49–71 (2009)
9. Dwork, C., Lynch, N., Stockmeyer, L.: Consensus in the Presence of Partial Synchrony. J. ACM 35, 288–323 (1988)
10. Heidergott, B.: Max-Plus Linear Stochastic Systems and Pertubation Analysis. Springer (2006)
11. Lamport, L., Shostak, R., Pease, M.: The Byzantine Generals Problem. ACM T. Progr. Lang. Sys. 4, 382–401 (1982)
12. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann (1996)
13. Meyn, S., Tweedie, R.L.: Markov Chains and Stochastic Stability. Springer (1993)
14. Neiger, G., Toueg, S.: Automatically Increasing the Fault-Tolerance of Distributed Algorithms. J. Algorithm. 11, 374–419 (1990)
15. Nowak, T., Függer, M., Kößler, A.: On the Performance of a Retransmission-based Synchronizer. Research Report 9/2011, TU Wien, Inst. f. Technische Informatik, `http://www.vmars.tuwien.ac.at/documents/extern/2899/paper.pdf` (2011)
16. Rajsbaum, S.: Upper and Lower Bounds for Stochastic Marked Graphs. Inform. Process. Lett. 49, 291–295 (1994)
17. 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)
18. Resing, J.A.C., de Vries, R.E., Hooghiemstra, G., Keane, M.S., Olsder, G.J.: Asymptotic Behavior of Random Discrete Event Systems. Stochastic Process. Appl. 36, 195–216 (1990)