On the Performance of a Retransmission-Based Synchronizer

Thomas Nowak, Matthias Függer, Alexander Kößler

On the Performance of a Retransmission-based Synchronizer

Thomas Nowak[a,*], Matthias Függer[b], Alexander Kößler[b]

[a] *LIX, Ecole polytechnique, 91128 Palaiseau CEDEX, France*
Tel: +33 1 69 33 41 42
Fax: +33 1 69 33 40 49
Email: nowak@lix.polytechnique.fr
[b] *ECS Group, TU Wien, 1040 Vienna, Austria*

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. Further, 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 environments with probabilistic message loss, within which the synchronizer shows nontrivial timing behavior. We show how to make this calculation efficient, and present analytical results on the convergence speed. The theoretic results, based on Markov theory, are backed up with Monte Carlo simulations.

Keywords: synchronizer, round-based algorithms, probabilistic environment, simulation, Markov theory

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 [1, 2], 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 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* [3, 4, 5, 6]: 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 [7] in the context of state-machine replication [8]. 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 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 messages 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.

[*] Corresponding author

1.1. Contributions

We claim to make the following contributions in this paper: (1) We give a method to determine the expected round duration of a general retransmission scheme, thereby generalizing results concerning stochastic max-plus systems by Resing *et al.* [9]. (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.

1.2. Organization of the Paper

Section 2 introduces the retransmission algorithm in question and the computing system model. Section 3 introduces a 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 4 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 5 contains simulation results. We give an overview on related work in Section 6. Conclusions are found in Section 7. The appendix contains facts about Markov chains that are used in the paper. Table 1 contains a list of the symbolic notation used in the paper.

A preliminary version of this work was presented at the SIROCCO 2011 conference [10].

2. The Retransmission Scheme

In this section, we formally present the object of study: a general technique to cope with message loss in distributed systems by retransmissions. Instead of handling message loss directly in the algorithm, it is often more convenient for the algorithm's designer to separate concerns into (1) simulating perfect rounds, i.e., rounds *without* message loss, on top of a system with message loss, and (2) to run a simpler algorithm on top of the simulated perfect rounds. Simulations that provide stronger communication directives on top of a system satisfying weaker communication directives are commonly used in distributed computing [11, 5]. In this section we present one such simulation—a retransmission scheme—and prove it correct. Note that the proposed retransmission scheme is a modified version of the α synchronizer [3]. However, it does not use the acknowledgment message.

2.1. Computational Model

We assume a distributed system comprising a fully connected communication network between *processes* taken from the set $\Pi = \{1, 2, \ldots, N\}$. Each process i has a *local state* s_i; a *global state* of the distributed system is a collection of local states $(s_i)_{i\in\Pi}$. Processes communicate by message passing.

Formally, an *algorithm* A for the distributed system comprises the following parts:

(A1) For every process i, a *set of possible local states* $\mathcal{S}_i$, a *set of possible initial local states* $\mathcal{S}_i^0$, and the *set of possible messages* $\mathcal{M}$, not containing $\perp$. We assume without loss of generality that the sets $\mathcal{S}_i$ are pairwise disjoint.

(A2) A pair of functions $(\mathrm{Send}_i, \mathrm{Next}_i)$ for every process i: The *send function* Send_i for every process i, is from $\mathcal{S}_i$ to $2^{\mathcal{M}}$, and maps a local state to a nonempty finite set of messages to send. The *next state function* Next_i for every process i, is from $\mathcal{S}_i \times 2^{\mathcal{M}\times\Pi}$ to $\mathcal{S}_i$, and maps a local state and a set $R \subseteq \mathcal{M} \times \Pi$ of received messages, labeled with their respective sender, to the next local state.

Computation at processes is assumed to occur in sequences of steps locally happening at the processes. In a step, a process atomically (E1) receives a set of messages, (E2) computes its next local state, and (E3) sends (broadcasts) a nonempty finite set of messages to all other processes. Note that our definition of a step differs from classic definitions with respect to (E3), potentially allowing an algorithm to broadcast a set of messages instead of a single message per step. While in distributed systems without transmission failures, algorithms for both kinds of definitions can be easily reduced to each other by joining all messages to be sent in a step into a single message, this is not the case for distributed systems that have to cope with transmission failures, like those we consider in our work. There, the extension allows for finer grained modeling of benign transmission failures, i.e., failures where contents of messages are not changed: Instead of the single message, sent in a step, either being received in some other step or not, an arbitrary subset of messages sent in a step can be received in some other step.

symbol	meaning	first use
Π	set of processes	Section 2.1
N	number of processes	Section 2.1
s_i	local state of process i	Section 2.1
$\mathcal{S}_i$	set of possible local states of process i	Section 2.1
$\mathcal{S}_i^0$	set of possible initial states of process i	Section 2.1
$\mathcal{M}$	set of possible messages	Section 2.1
Send_i	send function of process i	Section 2.1
Next_i	next state function of process i	Section 2.1
$E(i)$	projection of execution E to process i's states and events	Section 2.1
$E \upharpoonright B$	B-projection of execution E	Section 2.2
$s[X]$	value of variable X in state s	Section 2.2
$T_i(r)$	start of simulated round r at process i	Section 3
$L(r)$	$\max_i T_i(r)$	Section 3
$\delta_{i,j}(r)$	effective transmission delay from i to j in round r	Section 3
p	probability of successful transmission	Section 3
M	maximum number of tries per round	Section 3
λ	expected round duration	Section 4
$P_{X,Y}$	transition probability from state Y to state X	Section 4.1
$\sigma_z(r)$	$\#\big\{i \mid T_i(r) - L(r-1) = z\big\}$	Section 4.1
$\Lambda(r)$	$\big(\sigma_1(r), \ldots, \sigma_M(r)\big)$	Section 4.1
$\mathcal{L}$	state space of Markov chain $\Lambda(r)$	Section 4.1
$\sigma(\Lambda)$	$\max\{z \mid \sigma_z \neq 0\}$	Section 4.1
$\sigma(r)$	$\sigma\big(\Lambda(r)\big)$	Section 4.1
$\mathcal{L}_z$	set of $\Lambda \in \mathcal{L}$ such that $\sigma(\Lambda) = z$	Section 4.1
π	stationary distribution of $\Lambda(r)$	Section 4.1
$\mathrm{Norm}(\Lambda)$	normalized state of Λ	Section 4.2
$P(\leqslant z \mid \Lambda)$	probability that $T_i(r+1) - L(r) \leqslant z$ for a fixed i, given $\Lambda(r) = \Lambda$	Section 4.2
$P(z \mid \Lambda)$	probability that $T_i(r+1) - L(r) = z$ for a fixed i, given $\Lambda(r) = \Lambda$	Section 4.2
$P(\leqslant z \mid \Lambda, k)$	probability that $T_i(r+1) - L(r) \leqslant z$ for a fixed i, given $\Lambda(r) = \Lambda$ and $T_i(r) - L(r-1) = k$	Section 4.2
$P(z \mid \Lambda, k)$	probability that $T_i(r+1) - L(r) = z$ for a fixed i, given $\Lambda(r) = \Lambda$ and $T_i(r) - L(r-1) = k$	Section 4.2
$\lambda_{\mathrm{prob}}(p, M, N)$	value of λ for probability space **ProbLoss**(p, M) with N processes	Section 4.3
$\lambda_{\mathrm{det}}(p, M, N)$	value of λ for probability space **ProbLoss**$^*(p, M)$ with N processes	Section 4.3

Table 1: List of notation

Formally we define: An *event* is a tuple (i, R), where i is a process and R is the set of messages, tagged with their respective senders (i.e., $R \subseteq \mathcal{M} \times \Pi$) that are received by process i in the event. An *execution* E of an algorithm A is a sequence of events and local states such that for every process i, the projection $E(i)$ to process i's events and states is an alternating sequence of local states and events $E(i) = s_i(1), e_i(2), s_i(2), \ldots, e_i(k), s_i(k), \ldots$, such that (Ex1) every $s_i(1)$ is an initial (local) state of i and (Ex2) for every $k > 1$ with $e_i(k) = (i, R)$, it is $s_i(k) = \mathrm{Next}_i\left(s_i(k-1), R\right)$. In execution E, event e is *before* event e' if e appears before e' in sequence E. We say that process i *receives* message m from j in *step* k if $(m, j) \in R$ where $e_i(k) = (i, R)$. We further say that process i *sends* (broadcasts) message m in *step* k, if $m \in \mathrm{Send}_i\left(s_i(k)\right)$.

It remains to specify the relation between message sends and receives that has to hold during an execution. We do this by means of communication axioms which denote a condition on the distributed system's communication behavior: The system can either satisfy an axiom or not. The following are communication axioms used in the sequel:

NoGen For all processes i and j, if j receives message m from i, then i broadcasted m before.

FairLoss For all processes i and j, if i broadcasts the same message m in infinitely many steps, then j receives m from i in infinitely many steps.

Further desirable axioms are that of *communication closedness* **CommClosed** [5], *perfect communication* **PerfComm**, and perfect communication for self loops, i.e., **PerfComm***. They are defined by:

CommClosed For all processes i and j, if j receives message m from i in step $k > 1$, then i broadcasted m in step $k - 1$.

PerfComm For all processes i and j, if i broadcasts message m in step $k - 1$, $k > 1$, then j receives m from i in step k.

PerfComm* For all processes i, if i broadcasts message m in step $k - 1$, $k > 1$, then i itself receives m from i in step k.

Call an execution *admissible* if it satisfies **NoGen**, which is reasonable to assume for benign communication, and for each process i, $E(i)$ is infinite.

A *fair-lossy execution* of an algorithm A is an admissible execution that satisfies axiom **FairLoss**. A *perfect round execution* is an admissible execution that satisfies axioms **CommClosed** and **PerfComm**.

2.2. Simulating Perfect Round Executions

Our goal is to determine the round duration of a retransmission scheme that simulates a perfect round execution on top of a fair-lossy execution. We thus proceed by introducing a notion of simulation. Let B be an algorithm (designed for perfect round executions). We define what it means for an algorithm A (designed for fair-lossy executions) to simulate algorithm B. The idea is that algorithm A's local state includes B's local state in a special variable *Bstate*. Further, in each event, algorithm A is allowed to trigger a local event of algorithm B. It does this by setting a local variable *trigger* to *true*, and handing over a set of received messages to its local instance of B. Algorithm B then makes a step and updates *Bstate*.

Formally we define: Let $\mathcal{S}_i^{(B)}$ and $\mathcal{M}^{(\mathcal{B})}$ denote the sets of local states and the set of messages of B, respectively. We demand of algorithm A that its local states contain the variables *Bstate*, *trigger*, and *Bevent*. Variable *Bstate*'s type at process i is $\mathcal{S}_i^{(B)}$, variable *trigger* is Boolean, and variable *Bevent*'s type is $\Sigma^{(B)}$, where $\Sigma^{(B)}$ is the set of events of algorithm B.

Given an execution E of algorithm A, we define the *B-projection $E \restriction B$ of E* in the following way:

(P1) Let F denote the subsequence of E that arises when (a) deleting all events, and (b) all states in which *trigger* = *false*.

(P2) We define $E \restriction B$ to be the sequence arising from F when replacing each processor's first state, $s_i(1)$, by $s_i(1)[\mathit{Bstate}]$, and every but each processor's first state, $s_i(r)$, by the two elements $s_i(r)[\mathit{Bevent}]$, $s_i(r)[\mathit{Bstate}]$ where $s[X]$ denotes the value of variable X in state s.

Definition 1. We say that algorithm A *simulates B in perfect rounds on top of fair-lossy executions* if, (S1) *trigger* = *true* in every initial state of A, (S2) for every initial state $s_i^{(B)}(1)$ of B, there exists an initial state $s_i(1)$ of A such that $s_i(1)[\mathit{Bstate}] = s_i^{(B)}(1)$, and (S3) for every fair-lossy execution E of A, execution $E \restriction B$ is a perfect round execution of B.

2.3. The Algorithm

We are now ready to formally state a retransmission-based algorithm that simulates perfect round executions on top of fair-lossy ones, and prove it correct.

For every algorithm B, consider algorithm $A = A(B)$ presented in Figure 1. The idea of the simulation is simple: Each process steadily broadcasts (B1) its current (simulated) round number *Rnd* together with algorithm B's messages for the current round (*Rnd*) and, (B2) the previous round number $Rnd - 1$ together with algorithm B's messages for the previous round ($Rnd - 1$). A process waits in round *Rnd* until it has received all processes' round *Rnd* messages. When it does, it starts (simulated) round $Rnd + 1$.

The intuition for a process sending both its current and its previous round messages is the following: At some point during the execution, the value of any two processes' *Rnd* variables may differ by one, because of transmission failures. That is, while some process i already started simulated round K, and therefore waits for messages with round number K, another process j may still be in simulated round $K-1$, waiting for messages with round number $K-1$. Clearly, process i therefore must still send round $K-1$ messages to j, until j, too, starts round K. Messages with round number less than $K-1$, however, need not be sent by process i: It can be shown that at any point during the execution, the values of any two processes' *Rnd* variables differ by at most one (cf. proof of Proposition 1).

```
 1: VAR $BState \leftarrow s_i^{(B)}(1)$; $trigger \leftarrow true$; $Bevent \leftarrow \perp$;
 2: VAR $BState_{old} \leftarrow \perp$; $\forall j\ \forall r$: $Rcv[j,r] \leftarrow \perp$; $Rnd \leftarrow 1$;

 3: next state function when receiving set of messages $R$
 4:   for received message $(r, m) \in R$ from process $j$ do
 5:     $Rcv[j,r] \leftarrow m$;
 6:   end for
 7:   $trigger \leftarrow false$;
 8:   if for all $j$ in $\Pi$: $Rcv[j, Rnd] \neq \perp$ then
 9:     $Bstate_{old} \leftarrow Bstate$;
10:     $trigger \leftarrow true$;
11:     $R' \leftarrow \{(Rcv[j, Rnd], j) \mid j \in \Pi\}$;
12:     $Bevent \leftarrow (i, R')$;
13:     $Bstate \leftarrow \mathrm{Next}_i^{(B)}\big(Bstate, R'\big)$;
14:     $Rnd \leftarrow Rnd + 1$;
15:   end if
16: end next state function

17: send function
18:   broadcast $\Big(Rnd - 1, \mathrm{Send}_i^{(B)}(Bstate_{old})\Big)$; broadcast $\Big(Rnd, \mathrm{Send}_i^{(B)}(Bstate)\Big)$;
19: end send function
```

Figure 1: Process i's code in simulation algorithm $A(B)$

Proposition 1. *In every fair-lossy execution E of $A(B)$ holds: If there exists a process $i \in \Pi$ such that $s_i(k)[Rnd] \leqslant K$ for all k, then $s_j(k)[Rnd] \leqslant K + 1$ for all k and all $j \in \Pi$.*

Proof. By code line 18, i never sends a message of the form (r, m) with $r > K$. By **NoGen**, no process receives a message of the form (r, m) with $r > K$ from process i. Hence, by lines 4–6, all processes always have $Rcv[i, r] = \perp$ for all $r > K$, and, by lines 8 and 14, do not set *Rnd* to a higher value than $K+1$. □

Proposition 2. *In every fair-lossy execution E of $A(B)$ holds: If for all $i \in \Pi$ there exists a k such that $s_i(k)[Rnd] = K$, then for all $i \in \Pi$ there exists a k' such that $s_i(k')[Rnd] = K + 1$.*

Proof. Suppose, by means of contradiction, that there exists some process i such that $s_i(k')[Rnd] \leqslant K$ for all k'. Then by Proposition 1, $s_j(k')[Rnd] \leqslant K + 1$ for all k' and all $j \in \Pi$. Hence by code line 18 and the facts that every process $j \in \Pi$ has $Rnd = K$ in one of its steps and takes infinitely many steps, it follows that every process sends a message of the form (r, m) infinitely often where $r \in \{K, K + 1\}$. By **FairLoss**, all of these messages are received at least once. Then, by code line 18 and 4–6, process i has $Rcv[j, K] \neq \perp$ for all processes $j \in \Pi$ during some step of the execution. But then, by code line 14, also $Rnd = K + 1$. Contradiction. □

Proposition 3. *In every fair-lossy execution E of $A(B)$, for every process $i \in \Pi$, the sequence $s_i(k)[Rnd]$ is unbounded as $k \to \infty$.*

Proof. This is an immediate consequence of Proposition 2. □

From Propositions 1–3 we immediately obtain the correctness of the retransmission scheme:

Theorem 1. *For every algorithm B, algorithm $A(B)$ simulates B in perfect rounds on top of fair-lossy executions.*

Proof. It remains to show that (S3a) $E \restriction B$ is an execution of B and (S3b) $E \restriction B$ is perfect whenever E is fair-lossy. Property (S3a) follows from code lines 7, 10, and 11–13. Property (S3b) follows from code line 8 and Proposition 3. □

3. Round Durations under Probabilistic Message Loss

We have presented a simple algorithm to simulate perfect rounds on top of fair-lossy executions. In the rest of this paper, we analyze the performance of this solution.

In a fair-lossy execution E of algorithm $A(B)$, we define the *start of simulated round* r at process i, denoted by $T_i(r)$, to be the number of the step in $E(i)$ in which the state change from $Rnd = r-1$ to $Rnd = r$ was triggered; formally, $T_i(r) = k$ if $E(i) = s_i(1), e_i(2), s_i(2), \ldots$ and k is the smallest index such that $s_i(k)[Rnd] = r$. $L(r)$ is the number of the step where the last process starts its 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 local process steps.

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 by i. Formally, for any two processes i and j, let $\delta_{j,i}(r) - 1$ be the smallest number $\ell \geqslant 0$ such that (D1) process j sends a message m in its $(T_j(r)+\ell)^{\text{th}}$ step and (D2) process i receives m from j in its $(T_j(r)+\ell+1)^{\text{th}}$ step. We thus obtain the following proposition relating the starts of simulated rounds:

Proposition 4. *Let E be a fair-lossy execution of $A(B)$. For each process i: $T_i(1) = 1$, and for each $r \geqslant 1$:*

$$T_i(r+1) = \max_{1 \leqslant j \leqslant N} \left(T_j(r) + \delta_{j,i}(r) \right) \qquad (1)$$

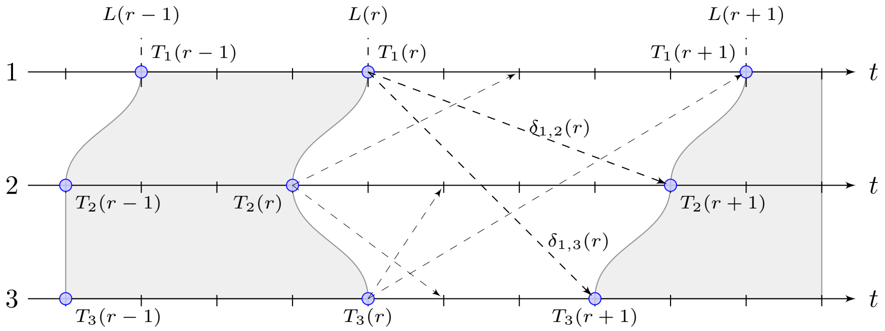


Figure 2: Fair-lossy execution of $A(B)$

Figure 2 depicts part of a fair-lossy execution of algorithm $A(B)$.

To allow for a quantitative assessment of the durations of the simulated rounds, besides the trivial bounds of $(0, \infty)$, we extend the modeling of the environment with a probability space: We introduce probability spaces **ProbLoss** and **ProbLoss***, for which we exemplarily calculate the expected average simulated round duration.

For all processes i and j, if process i sends message m in its $(k-1)^{\text{th}}$ step, $k > 1$, then process j receives m from i in its k^{th} step with probability p, where $0 < p \leqslant 1$, is called the *probability of successful transmission.*[1]

Formally, let **ProbLoss**(p) be the probability distribution on the set of fair-lossy executions defined by: The random variables $\delta_{j,i}(r)$ are pairwise independent, and for any two processes i, j, the probability that $\delta_{j,i}(r) = z$ is $(1-p)^{z-1} \cdot p$.

[1] In systems in which the probability of successful transmission is bounded from below by some $p > 0$, axiom **FairLoss** holds with probability 1.

For computational purposes we also introduce the probability distribution $\mathbf{ProbLoss}(p, M)$, where $M \in \mathbb{N} \cup \{\infty\}$, which is obtained from $\mathbf{ProbLoss}(p)$ by modifying the distribution of the $\delta_{j,i}(r)$: In contrast to $\mathbf{ProbLoss}(p)$ we bound the number of tries per simulated round message until it is successfully received 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, 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, $\mathbf{ProbLoss}(p, \infty) = \mathbf{ProbLoss}(p)$.

In order to describe systems satisfying the realistic assumption $\mathbf{PerfComm}^*$, we define $\mathbf{ProbLoss}^*(p)$ and $\mathbf{ProbLoss}^*(p, M)$ in the same way as $\mathbf{ProbLoss}(p)$ and $\mathbf{ProbLoss}(p, M)$, except that always $\delta_{i,i}(r) = 1$ for all r and processes i.

We will see in Sections 4.3 and 5, 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. It is further shown in these sections that for $M \geqslant 4$, $\mathbf{ProbLoss}(p, M)$ is a good approximation of $\mathbf{ProbLoss}^*(p, M)$.

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

Proposition 5. *If $T_i(r)/r$ 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 $r \to \infty$.

4. Calculating the Expected Round Duration

The expected round duration of the retransmission algorithm, in the case of fair-lossy executions distributed according to $\mathbf{ProbLoss}(p, M)$ or $\mathbf{ProbLoss}^*(p, M)$, 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 $\lambda_i = \mathbb{E} \lim_{r\to\infty} T_i(r)/r$. Because of Proposition 5, for any two processes i, j it holds that $\lambda_i = \lambda_j = \lambda$, where $\lambda = \mathbb{E} \lim_{r\to\infty} L(r)/r$.

The section is structured as follows: Section 4.1 provides the definition of the Markov chain $\Lambda(r)$. Section 4.2 develops a method to compute the expected round duration using $\Lambda(r)$. Section 4.3 shows the use of $\Lambda(r)$ by giving several examples. Section 4.4 presents lower bounds of the convergence speed of the round durations. A certain familiarity with basic notions of probability theory is assumed; however, no advanced knowledge is necessary for the comprehension of this section. Supplemental facts and definitions about Markov chains can be found in Appendix A.

4.1. 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 impossible 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 $T(r)$; for a process executing algorithm $A(B)$ decides to increment its variable *Rnd* in step k based only on the round numbers it receives in step k and the value of its variable *Rnd* in step $k-1$. Thus the probability that $T(r) = X$ given that $T(r-1) = Y$ is equal to the probability that $T(r) = X - c$ given that $T(r-1) = Y - c$, if $c \in \mathbb{N}_0$. Choosing $c = L(r-1)$, and observing that $T_i(r) - L(r-1)$ is upper bounded by M, this yields a finite state space for finite M, which enabled us to calculate the expected round duration.

Also, 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 is feasible because the system is invariant under permutation of processes: The probability that $T(r) = X$ given that $T(r-1) = Y$ is equal to the probability that $T(r) = X'$ given that $T(r-1) = Y'$, where $X'_i = X_{\phi(i)}$ and $Y'_i = Y_{\phi(i)}$ for an arbitrary permutation ϕ of Π. This optimization further reduces the size of the state space from M^N to $\binom{N+M-1}{M-1}$, which is polynomial in N; in practical situations, it suffices to use modest values of M as will be shown in Section 5. We show in Theorem 2 that the information recorded in the states of Markov chain $\Lambda(r)$ suffices to determine the expected round duration λ.

We are now ready to formally 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 in $\mathcal{L}$ are related to $T(r)$ as follows: Let $\#X$ be the cardinality of the set X, and set

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

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 2, $\sigma_1(r) = 0$, $\sigma_2(r) = 1$ 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 6. *$\Lambda(r)$ is a Markov chain.*

Proof. On the set of collections (x_i) of numbers indexed by $\Pi = \{1, 2, \ldots, N\}$, we introduce equivalence relation $\sim$ by defining $(x_i) \sim (y_i)$ if and only if there exists a bijection $\phi : \Pi \to \Pi$ such that $x_i = y_{\phi(i)}$ for every $i \in \Pi$. We have $(x_i) \sim (y_i)$ if and only if the multisets $\{x_i \mid i \in \Pi\}$ and $\{y_i \mid i \in \Pi\}$ are equal. Denote by $\big[(x_i)\big]$ the equivalence class of collection (x_i). Every state $\Lambda \in \mathcal{L}$ naturally corresponds to such an equivalence class.

Let $r > 0$ and $\Lambda_1, \Lambda_2, \ldots, \Lambda_{r-1} \in \mathcal{L}$. We need to show that the conditional distribution for $\Lambda(r)$, given $\Lambda(1) = \Lambda_1$, …, $\Lambda(r-1) = \Lambda_{r-1}$, is the same as the conditional distribution for $\Lambda(r)$, given only $\Lambda(r-1) = \Lambda_{r-1}$. By Equations (3) and (2), it suffices to show that the conditional distributions for $\mathcal{A}(r) = \big[\big(A_i(r)\big)\big]$ where $A_i(r) = T_i(r) - L(r-1)$, are equal.

We claim that the distribution of $\mathcal{A}(r)$ only depends on $\mathcal{B}(r) = \big[\big(B_i(r)\big)\big]$ where $B_i(r) = T_i(r-1) - L(r-1)$. From Equation (1) it follows that $A_i(r) = \max_j \big(B_j(r) + \delta_{j,i}(r-1)\big)$. Let $\widetilde{B}(r) \in \mathcal{B}(r)$, i.e., $\widetilde{B}_i(r) = B_{\phi(i)}(r)$ for a bijection $\phi : \Pi \to \Pi$ and define $\widetilde{A}_i(r) = \max_j \big(\widetilde{B}_j(r) + \delta_{j,i}(r-1)\big)$. We show that there exists a bijection $\psi : \Pi \to \Pi$ such that the distributions for $A_i(r)$ and $\widetilde{A}_{\psi(i)}(r)$ are equal. It suffices to set $\psi = \phi^{-1}$. Then, $\widetilde{A}_{\psi(i)}(r) = \max_j \big(B_{\phi(j)}(r) + \delta_{j,\psi(i)}(r-1)\big) = \max_j \big(B_j(r) + \delta_{\psi(j),\psi(i)}(r-1)\big)$. Since $(j,i) \mapsto \big(\psi(j), \psi(i)\big)$ is a permutation of Π^2, and $\delta_{\psi(j),\psi(i)}(r-1)$ and $\delta_{j,i}(r-1)$ are identically distributed for all $(j,i) \in \Pi^2$, the claim follows.

Equivalence class $\mathcal{B}(r)$, in turn, is completely determined by Λ_{r-1} because of the identity $B_i(r) = A_i(r-1) - \max_j A_j(r-1)$. This concludes the proof. □

In fact, Proposition 6 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 drop the independence assumption 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'.[2]

Let $X(r)$ be a Markov chain with countable state space $\mathcal{X}$ and transition probabilities P. A probability distribution π on $\mathcal{X}$ is a *stationary distribution* for $X(r)$ if $\pi(X) = \sum_{Y \in \mathcal{X}} \pi(Y) \cdot P_{X,Y}$ 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 2. Call a Markov chain *good* if it is aperiodic, irreducible, Harris recurrent, and has a unique stationary distribution.[3]

[2] This is the case of "negligible transmission delays" considered by Rajsbaum and Sidi [6].

[3] The notions "aperiodic", "irreducible", and "Harris recurrent" are standard in Markov theory and are recalled in the appendix.

Proposition 7. *$\Lambda(r)$ is a good Markov chain.*

Proof. $\Lambda(r)$ is aperiodic because every state can be reached from every other in two and in three steps with nonzero probability: The transition probability from every state to state $(N, 0, \ldots, 0)$ is nonzero, for this transition occurs if all messages arrive on their first try. Also, the transition probability from state $(N, 0, \ldots, 0)$ to every other state is nonzero.

Harris recurrence follows from the fact that every state can be reached in two steps with nonzero probability, together with the fact that the state space is finite.

Existence and uniqueness of the stationary distribution follows from recurrence [12, Theorem 10.0.1]. □

Denote by π the unique stationary distribution of $\Lambda(r)$, which exists because of Proposition 7. 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 it follows that $\sigma(r) = L(r) - L(r-1)$, i.e., knowing $\sigma(1)$ to $\sigma(r)$ suffices to determine $L(r)$. For example, $\sigma(r+1) = 5$ in the execution in Figure 2.

Proposition 8. *$L(r) = \sum_{k=1}^{r} \sigma(k)$*

Proof. The proof is by induction on r. The case $r = 1$ is trivial. We are done if we show $L(r) = L(r-1) + \sigma(r)$ for all $r > 1$. By definition, we have $L(r-1) + \sigma(r) = L(r-1) + \max_i \big(T_i(r) - L(r-1)\big)$. Noting the rule $A + \max_i B_i = \max_i(A + B_i)$ concludes the proof. □

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 4.2. The theorem 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 of $\Lambda(r)$. Let $\mathcal{L}_z \subseteq \mathcal{L}$ denote the set of states Λ such that $\sigma(\Lambda) = z$. Then:

Theorem 2. *$L(r)/r$ converges to λ with probability 1. Furthermore, $\lambda = \sum_{z=1}^{M} z \cdot \pi(\mathcal{L}_z) < \infty$.*

Proof. We use Theorem A.1 in the appendix and prove that its hypothesis holds by showing $\sum_{z \geqslant 1} z \cdot \pi(\mathcal{L}_z) \leqslant 2^{N^2} p^{-2}$.

As a first step, we show $\pi(\mathcal{L}_z) \leqslant 2^{N^2}(1-p)^{z-1}$. Because $\mathbb{P}\big(\sigma(r) = z\big)$ converges to $\pi(\mathcal{L}_z)$ as $r \to \infty$ (Theorem A.2 in the appendix), it suffices to prove this inequality for $\mathbb{P}\big(\sigma(r) = z\big)$. The event $\sigma(r) = z$ implies the event $\exists i, j : \delta_{i,j}(r) \geqslant z$, i.e., the complement of the event $\forall i, j : \delta_{i,j}(r) \leqslant z - 1$. The events $\delta_{i,j}(r) \leqslant z - 1$ each have probability $1 - (1-p)^{z-1}$. Hence

$$\mathbb{P}\big(\sigma(r) = z\big) \leqslant 1 - \big(1 - (1-p)^{z-1}\big)^{N^2} \tag{4}$$

for all $r \geqslant 1$.

We now manipulate the right-hand side of Equation (4) with operations that preserve the inequality. We invoke the binomial theorem and the triangle inequality, arriving at $\sum_{k=0}^{N^2} \binom{N^2}{k}(1-p)^{k(z-1)}$. Finally, we substitute $k(z-1)$ by $z-1$ and use the identity $\sum_k \binom{n}{k} = 2^n$ to prove the claimed inequality $\pi(\mathcal{L}_z) \leqslant 2^{N^2}(1-p)^{z-1}$.

Using the derivative of the geometric sum formula, we calculate $\sum_{z=0}^{\infty} z(1-p)^{z-1} = 1/p^2$. This concludes the proof. □

4.2. Using $\Lambda(r)$ to Compute λ

We now state a method that, given parameters $M \neq \infty$, N, and p, computes the expected simulated round duration λ (see Theorem 2). 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 $P_{X,Y}$, from each state Y to each state X, which we regard as a matrix P. We will do this using two different assumptions on the communication system: (i) for the simpler case $\mathbf{ProbLoss}(p, M)$ of a system with probabilistic loop-back links, i.e., where we do not assume that $\mathbf{PerfComm}^*$ holds, and (ii) for a system $\mathbf{ProbLoss}^*(p, M)$ with the (more realistic) assumption of $\mathbf{PerfComm}^*$.

A first observation, that is valid for both systems, 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, $\Lambda = (\sigma_1, \dots, \sigma_M)$ 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:

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

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

(i) Probabilistic loop-back links **ProbLoss**. For any $\Lambda = (\sigma_1, \dots, \sigma_M)$ 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 $\Lambda(r) = \Lambda$, i.e.,

$$P(\leqslant z \mid \Lambda) = \mathbb{P}(T_i(r+1) - L(r) \leqslant z \mid \Lambda(r) = \Lambda) \ . \tag{5}$$

Since the right-hand side is independent of i and r, $P(\leqslant z \mid \Lambda)$ is well-defined. We easily observe that $T_i(r+1) - L(r) \leqslant z$, given that $\Lambda(r) = \Lambda$, 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:

$$P(\leqslant z \mid \Lambda(r)) = \prod_{1 \leqslant u \leqslant M} \mathbb{P}(\delta \leqslant z + M - u)^{\sigma_u(r)} \ , \tag{6}$$

for all z, $1 \leqslant z \leqslant M$. Let $P(z \mid \Lambda)$ be the conditional probability that a specific process is in the set $\{i \mid T_i(r+1) - L(r) = z\}$, given that $\Lambda(r) = \Lambda$, i.e.,

$$P(z \mid \Lambda(r)) = \mathbb{P}(T_i(r+1) - L(r) = z \mid \Lambda(r) = \Lambda) \ . \tag{7}$$

From Equations (5) and (7), we immediately obtain:

$$\begin{aligned} &P(1 \mid \Lambda) = P(\leqslant 1 \mid \Lambda) \text{ and,} \\ &P(z \mid \Lambda) = P(\leqslant z \mid \Lambda) - P(\leqslant z-1 \mid \Lambda) \ , \end{aligned} \tag{8}$$

for all z, $1 < z \leqslant M$. We may finally state the transition matrix P: for each $X, Y \in \mathcal{L}$, the probability that the system makes a transition from state $Y = \Lambda(r) = (\sigma_1, \dots, \sigma_M)$ to state $X = \Lambda(r+1) = (\sigma'_1, \dots, \sigma'_M)$ is given by the probability that of the N processes, there are σ'_1 processes in the set $\{i \mid T_i(r+1) - L(r) = 1\}$, of the $N - \sigma'_1$ remaining processes, there are σ'_2 processes in the set $\{i \mid T_i(r+1) - L(r) = 2\}$, etc. Finally, the remaining $\sigma'_M = N - \sum_{z=1}^{M-1} \sigma'_z$ processes are in the set $\{i \mid T_i(r+1) - L(r) = M\}$. This yields,

$$P_{X,Y} = \binom{N}{\sigma'_1, \sigma'_2, \dots, \sigma'_M} \prod_{1 \leqslant z \leqslant M} P(z \mid \mathrm{Norm}(Y))^{\sigma'_z} \ , \tag{9}$$

where for any finite sequence $a_1, \dots, a_m$ with $m \geqslant 1$ and elements from $\mathbb{N}_0$, the multinomial coefficient $\binom{\sum_{\ell=1}^m a_i}{a_1, a_2, \dots, a_m}$ is equal to $\prod_{1 \leqslant \ell \leqslant m} \binom{\sum_{k=1}^{\ell} a_k}{a_\ell}$, i.e., the number of possibilities to distribute $\sum_{\ell=1}^m a_i$ processes into m bins of sizes $a_1, \dots, a_m$.

(ii) Deterministic loop-back links **ProbLoss***. Note that for a system where **PerfComm*** holds, in Equation (6), 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 (6), 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 $\{j \mid T_j(r+1) - L(r) \leqslant z\}$, 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 (8) can be generalized in a straightforward manner to obtain expressions for $P(z \mid \Lambda, k)$, i.e., for the conditional probability that process i is in the set $\{i \mid T_i(r+1) - L(r) = z\}$, given that $\Lambda(r) = \Lambda$, as well as $T_i(r) - L(r-1) = k$.

When stating a formula for $P_{X,Y}$ analogous to Equation (9), one has to account for the dependency of $P(z \mid \Lambda, k)$ on k. For that purpose let $P_{X,Y}(Q)$, where Q is an $M \times M$ matrix with elements from $\mathbb{N}_0$, be the transition probability from state $Y = \Lambda(r)$ with $\mathrm{Norm}(Y) = (\sigma_1, \ldots, \sigma_M)$ to state $X = \Lambda(r+1) = (\sigma'_1, \ldots, \sigma'_M)$, provided that $Q_{z,k}$ is the number of processes which are in both $\{i \mid T_i(r+1) - L(r) = z\}$ and $\{i \mid T_i(r) - L(r-1) = k\}$. By definition, $P_{X,Y}(Q)$ is nonzero only if $\sum_{z=1}^{M} Q_{z,k} = \sigma_k$, for $1 \leqslant k \leqslant M$, and $\sum_{k=1}^{M} Q_{z,k} = \sigma'_z$, for $1 \leqslant z \leqslant M$. We readily obtain,

$$P_{X,Y}(Q) = \prod_{1 \leqslant k \leqslant M} \left(\binom{\sigma_k}{Q_{1,k}, Q_{2,k}, \ldots, Q_{M,k}} \prod_{1 \leqslant z \leqslant M} P(z \mid \mathrm{Norm}(Y), k)^{Q_{z,k}} \right) \ . \tag{10}$$

To calculate $P_{X,Y}$ one has to account for all possible choices of Q, each of which occurs with probability $P_{X,Y}(Q)$. With $\mathcal{Q}$ being the set of $M \times M$ matrices with elements from $\mathbb{N}_0$ for which $\sum_{z=1}^{M} \sum_{k=1}^{M} Q_{z,k} = N$, we finally obtain

$$P_{X,Y} = \sum_{Q \in \mathcal{Q}} P_{X,Y}(Q) \ . \tag{11}$$

While the calculation of the transition probabilities $P_{X,Y}$ depends on the specific communication assumptions made, the method to obtain λ from the expressions for $P_{X,Y}$ is independent from all these assumptions. It is presented in the following. Let $\Lambda_1, \Lambda_2, \ldots, \Lambda_n$ be any enumeration of states in $\mathcal{L}$. We write $P_{i,j} = P_{\Lambda_i,\Lambda_j}$ and $\pi_i = \pi(\Lambda_i)$ to view P as an $n \times n$ matrix and π as a row vector. By definition, the unique stationary distribution π satisfies (1) $\pi = \pi \cdot P$, (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:

$$\pi = e \cdot \left(P^{(n \to 1)} - I^{(n \to 0)} \right)^{-1} \tag{12}$$

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

After calculating π, we can use Theorem 2 to finally determine the expected simulated round duration λ. The time complexity of this approach is determined by (T1) building transition matrix P, and (T2) the matrix inversion of P. For both probability spaces (i) $\mathbf{ProbLoss}(p, M)$ and (ii) $\mathbf{ProbLoss}^*(p, M)$, matrix P is of the same size $n \times n$, where $n = \binom{N+M-1}{M-1}$ is the number of states in the Markov chain $\Lambda(r)$. Thus the time complexity of (T2) is within $O(n^3)$, which is polynomial in N. With respect to (T1) a naïve implementation of the procedure presented in (ii) has time-complexity at least $\#\mathcal{Q} = \binom{N+M^2-1}{M^2-1}$, which outweighs (T2), in contrast to the method presented in (i).

In Sections 4.4 and 5 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 method.

4.3. Results

The presented method allows to obtain analytic expressions for λ for fixed M and N in terms of probability p. Denote by $\lambda_{\mathrm{prob}}(p, M, N)$ respectively $\lambda_{\mathrm{det}}(p, M, N)$ the value of λ for probability space $\mathbf{ProbLoss}(p, M)$ respectively $\mathbf{ProbLoss}^*(p, M)$ with N processes. Figure 3 contains $\lambda_{\mathrm{det}}(p, M, N)$ for $M = 2$ and N equal to 2 and 3. For larger M and N, the expressions already become significantly longer.

Clearly for all p, M and N, $\lambda_{\mathrm{det}}(p, M, N)$ is less or equal to $\lambda_{\mathrm{prob}}(p, M, N)$, since $\mathbf{ProbLoss}$ differs from $\mathbf{ProbLoss}^*$ only by restricting $\delta_{i,i}(r)$ to attain the minimum value of 1 for each process i in each simulated round r. So if one is interested in nontrivial upper bounds of deterministic loop-back systems,

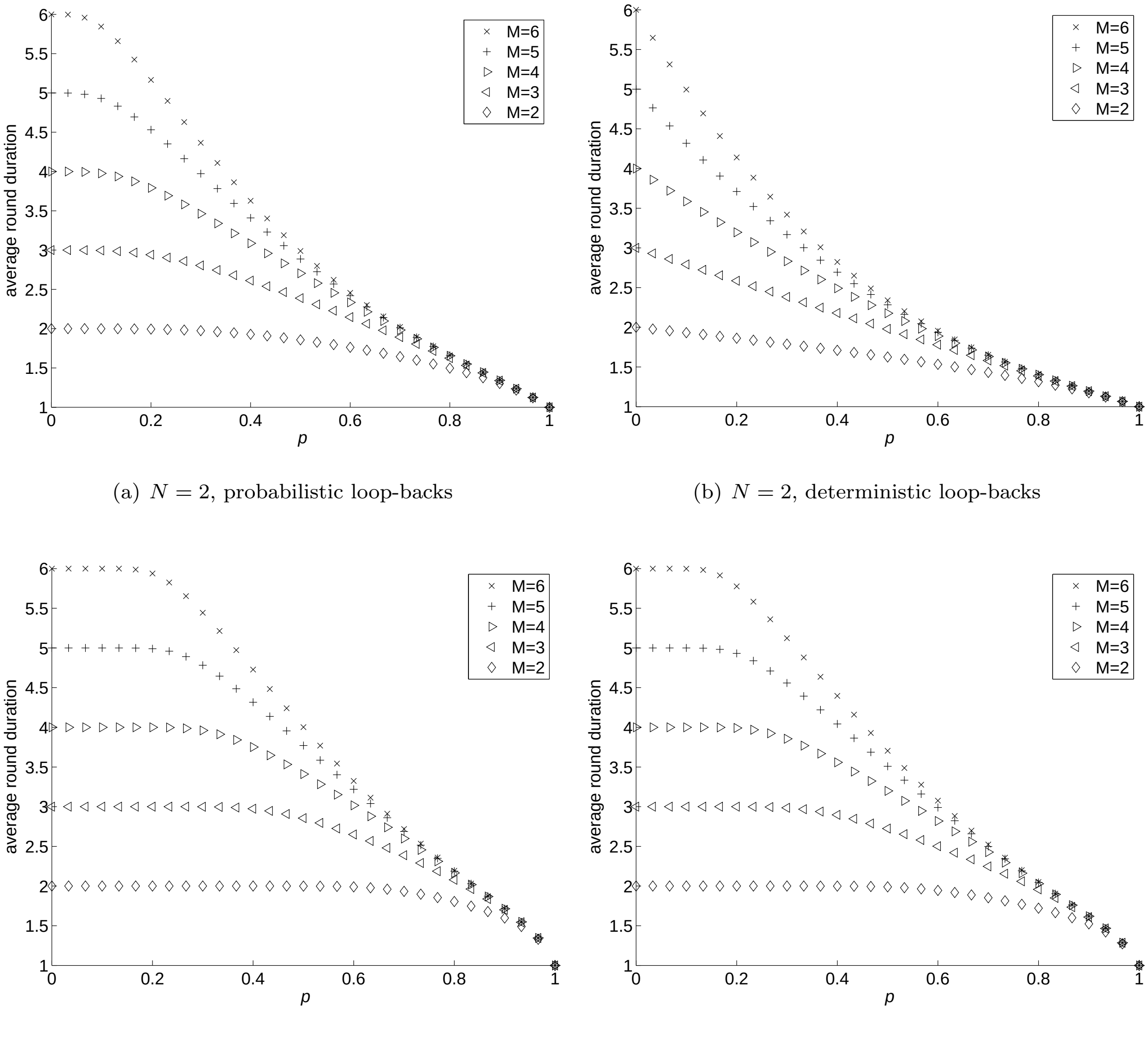
$$\lambda_{\mathrm{det}}(p,2,2) = \frac{6-6p+p^2}{3-2p}$$

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

Figure 3: Expressions for $\lambda_{\mathrm{det}}(p, M, N)$ with $M = 2$ and $N = 2, 3$

probabilistic loop-back systems are a good choice. Figures 4(a)–4(d) even suggest that $\lambda_{\text{prob}}(p, M, N)$ is a good approximation for $\lambda_{\text{det}}(p, M, N)$ for $N \geqslant 4$: Figures 4(a) and 4(b) show solutions of $\lambda_{\text{prob}}(p, M, 2)$ and $\lambda_{\text{det}}(p, M, 2)$ while Figures 4(c) and 4(d) show solutions for $\lambda_{\text{prob}}(p, M, 4)$ and $\lambda_{\text{det}}(p, M, 4)$ respectively.

We further observe that for high values of the probability of successful communication p, systems with different M have approximately the same slope. Since real distributed systems typically have a high p value, we may approximate $\lambda_{\text{det}}(p, M, N)$ as well as $\lambda_{\text{prob}}(p, M, N)$ for higher M values with that of significantly lower M values. The effect is further investigated in Section 5 by means of Monte Carlo simulation.



(a) $N = 2$, probabilistic loop-backs

(b) $N = 2$, deterministic loop-backs

(c) $N = 4$, probabilistic loop-backs

(d) $N = 4$, deterministic loop-backs

Figure 4: $\lambda_{\text{prob}}(p, M, N)$ and $\lambda_{\text{det}}(p, M, N)$ versus p for $N = 2, 4$ and $M \leqslant 6$

4.4. Rate of Convergence

We know from Theorem 2 that $L(r)/r$ converges to λ. The purpose of this section is to establish results on the rate of this convergence. As a particular result, we will see that also $\sigma(r)$ converges to λ. Our main result of this section will be a lower bound on the probability for the event $|L(r)/r - \lambda| < A$ (Theorem 3). We assume $M < \infty$ in this section.

The first proposition shows exponential convergence of $\sigma(r)$'s expected value to λ. It is the consequence of a standard result in Markov theory.

Proposition 9. *There exists some ρ, $0 < \rho < 1$, such that $\mathbb{E}\,\sigma(r) = \lambda + O\left(\rho^r\right)$ as $r \to \infty$.*

Proof. By definition of the expected value, $\mathbb{E}\,\sigma(r) = \sum_{z=1}^{M} z \cdot \mathbb{P}\big(\Lambda(r) \in \mathcal{L}_z\big)$. By Theorem A.2, it is $\mathbb{P}\big(\Lambda(r) \in \mathcal{L}_z\big) = \pi(\mathcal{L}_z) + O(\rho^r)$ for some ρ, $0 < \rho < 1$. Combining the two equations yields the claimed formula by Theorem 2. □

Having established the rate of convergence of $\sigma(r)$, we may conclude something about the rate of convergence of $L(r)/r$, i.e., its averages. However, we do not arrive at exponential convergence of $L(r)/r$ towards λ, but only $O(r^{-1})$. This can be seen as a consequence of the tendency of averages to even out drastic changes. The mathematical reason for it is that the sum $\sum_{k=1}^{r} \rho^k$ does not tend to zero as $r \to \infty$.

Proposition 10. *$\mathbb{E}\,L(r)/r = \lambda + O(1/r)$ as $r \to \infty$.*

Proof. By Proposition 8, we have $\mathbb{E}\,L(r)/r = 1/r \sum_{k=1}^{r} \mathbb{E}\,\sigma(k)$. Now, using Proposition 9 and noting that $\sum_{k=1}^{r} \rho^k = O(1)$ as $r \to \infty$ concludes the proof. □

Next, we investigate the *variance* of $\sigma(r)$.

Proposition 11. *There exists some ρ, $0 < \rho < 1$, such that* $\operatorname{Var}\big(\sigma(r)\big) = \beta - \lambda^2 + O(\rho^r)$ *as $r \to \infty$, where $\beta = \sum_{z=1}^{M} z^2 \cdot \pi(\mathcal{L}_z)$.*

Proof. The proposition follows by the same means as Proposition 9 after using the formula $\operatorname{Var}(X) = \mathbb{E}\,X^2 - (\mathbb{E}X)^2$. □

The next proposition provides two insights: (1) As r tends to infinity, the variance of $L(r)/r$ tends to zero; in contrast, the variance of $\sigma(r)$ tends to $\beta - \lambda^2$ (Proposition 11). This is a common phenomenon when considering averages of random variables (cf. Law of Large Numbers). (2) We show a rate of convergence of $O(1/r)$ for the variance of $L(r)/r$. This is an improvement over standard Markov theoretic results, which are able to show that the variance is $O(\log\log r/r)$ [12, Theorem 17.0.1(iv)-LIL].

Proposition 12. $\operatorname{Var}\big(L(r)/r\big) = O(1/r)$ *as $r \to \infty$.*

Proof. We subdivide the proof into a sequence of claims, which we prove separately.

Claim 1. $\mathbb{E}\,\sigma(k) \cdot \sigma(\ell) = \lambda^2 + O\big(\rho^{\min(k,\ell-k)}\big)$ uniformly for all $k < \ell$.

By definition of the expected value, $\mathbb{E}\,\sigma(k) \cdot \sigma(\ell)$ is equal to

$$\sum_{z=1}^{M} \sum_{u=1}^{M} z \cdot u \cdot \mathbb{P}\big(\Lambda(k) \in \mathcal{L}_z \wedge \Lambda(\ell) \in \mathcal{L}_u\big). \tag{13}$$

But $\mathbb{P}\big(\Lambda(k) \in \mathcal{L}_z \wedge \Lambda(\ell) \in \mathcal{L}_u\big)$ is equal to

$$\sum_{\Lambda \in \mathcal{L}_z} \mathbb{P}\big(\Lambda(k) = \Lambda\big) \cdot \mathbb{P}\big(\Lambda(\ell) \in \mathcal{L}_u \mid \Lambda(k) = \Lambda\big). \tag{14}$$

Theorem A.2 states that there exists a ρ, $0 < \rho < 1$ such that $\mathbb{P}\big(\Lambda(k) = \Lambda\big) = \pi(\Lambda) + O(\rho^k)$ and $\mathbb{P}\big(\Lambda(\ell) \in \mathcal{L}_u \mid \Lambda(k) = \Lambda\big) = \pi(\mathcal{L}_u) + O(\rho^{\ell-k})$.

Substituting this last equality into (14), together with $\pi(\mathcal{L}_z) = \sum_{\Lambda \in \mathcal{L}_z} \pi(\Lambda)$ and Theorem 2, yields that (13) is equal to $\lambda^2 + O\left(\rho^{\min(k,\ell-k)}\right)$. We have thus proved Claim 1.

Claim 2. $\operatorname{Cov}\big(\sigma(k), \sigma(\ell)\big) = O\big(\rho^{\min(k,\ell-k)}\big)$ uniformly for all $k < \ell$.

This claim follows from the formula $\operatorname{Cov}(X, Y) = \mathbb{E}(X \cdot Y) - \mathbb{E}X \cdot \mathbb{E}Y$, together with Claim 1 and Proposition 9.

Claim 3. $\displaystyle\sum_{1 \leqslant k < \ell \leqslant r} \rho^{\min(k,\ell-k)} = O(r)$

Define $a(k, \ell) = \rho^{\min(k,\ell-k)}$. Denote by $A(r)$ the set of pairs (k, ℓ) such that $1 \leqslant k < \ell \leqslant r$. Further define $B(r)$ to be the set of pairs (k, ℓ) in $A(r)$ that satisfy $2k < \ell$ and $C(r)$ to be the set of pairs (k, ℓ) in $A(r)$ that satisfy $2k \geqslant \ell$. It is $A(r) = B(r) \cup C(r)$. For $(k, \ell) \in B(r)$, we have $a(k, \ell) = \rho^k$ and for $(k, \ell) \in C(r)$, we have $a(k, \ell) = \rho^{\ell-k}$.

Hence,

$$\sum_{(k,\ell)\in B(r)} a(k,\ell) \leqslant \sum_{\ell=1}^{r}\sum_{k=1}^{r} \rho^k. \tag{15}$$

We calculate $\sum_{k=1}^{r} \rho^k = (\rho - \rho^{r+2})/(1-\rho) = O(1)$, which implies that the right-hand side of (15) is $O(r)$.

Similarly,

$$\sum_{(k,\ell)\in C(r)} a(k,\ell) \leqslant \sum_{k=1}^{r}\sum_{\ell=k+1}^{2k} \rho^{\ell-k} = \sum_{k=1}^{r}\sum_{\ell=1}^{k} \rho^{\ell} \leqslant \sum_{k=1}^{r}\sum_{\ell=1}^{r} \rho^{\ell} \tag{16}$$

is also $O(r)$. This proves Claim 3.

Claim 4. $\operatorname{Var}\bigl(L(r)/r\bigr) = O(1/r)$

We use the formulas $\operatorname{Var}\bigl(\sum_i X_i\bigr) = \sum_i \operatorname{Var}(X_i) + 2\sum_{i<j} \operatorname{Cov}(X_i, X_j)$ and $\operatorname{Var}(aX) = a^2 \cdot \operatorname{Var}(X)$, which, together with Proposition 11 and Claims 2 and 3, implies Claim 4. This concludes the proof. □

We can utilize the acquired knowledge about expected value and variance of $L(r)/r$ to explicitly state an asymptotic lower bound on the probability that $L(r)/r$ has distance at most α to the expected value λ. This is a standard procedure and uses Chebyshev's inequality, which can be stated as

$$\mathbb{P}\bigl(|X - \mathbb{E}X| \geqslant A\bigr) \leqslant (\operatorname{Var} X)^2/A^2 \ . \tag{17}$$

In our case, however, we do not have *one* random variable, but countably many. Thus, we do not limit ourselves to considering a single constant A, but we allow a sequence α_r instead of A. The case of a constant is a particular case.

Theorem 3. *If $M < \infty$ and $\alpha_r \cdot r \to \infty$ as $r \to \infty$, then*

$$\mathbb{P}\left(|L(r)/r - \lambda| \geqslant \alpha_r\right) = O\bigl(1/r^2\alpha_r^2\bigr)$$

as $r \to \infty$.

Proof. Let $\mathbb{E}\, L(r)/r = \lambda + g_r$. Then, by Proposition 10, we have $g_r = O(1/r)$. The condition $|L(r)/r - \lambda| \geqslant \alpha_r$ is equivalent to $|L(r)/r - \lambda| - |g_r| \geqslant \alpha_r - |g_r|$, which, by the triangle inequality, implies $|L(r)/r - (\lambda + g_r)| \geqslant \alpha_r - |g_r|$.

Hence, $\mathbb{P}\bigl(|L(r)/r - \lambda| \geqslant \alpha_r\bigr)$ is less or equal to $\mathbb{P}\bigl(|L(r)/r - (\lambda + g_r)| \geqslant \alpha_r - |g_r|\bigr)$, which, by Chebyshev's inequality (17), yields

$$\mathbb{P}\left(|L(r)/r - \lambda| \geqslant \alpha_r\right) \leqslant \frac{\operatorname{Var}(L(r)/r)^2}{(\alpha_r - |g_r|)^2},$$

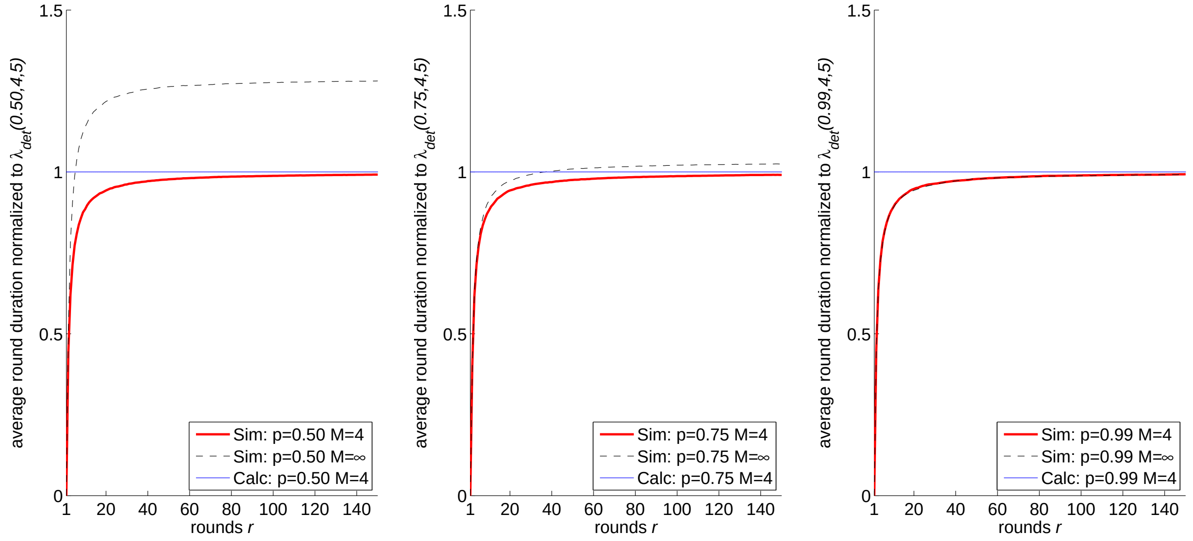
which is $O\bigl(1/r^2\alpha_r^2\bigr)$. Here we used Proposition 12 and the fact that $\alpha_r - |g_r| = \Omega(\alpha_r)$, which follows from $g_r = O(1/r)$ and $\alpha_r \cdot r \to \infty$. □

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

5. Simulations

The method presented in Section 4.2 allows to calculate $\lambda_{\mathrm{prob}}(p, M, N)$ and $\lambda_{\mathrm{det}}(p, M, N)$ if $M < \infty$. Therefore, the question arises whether the solutions for finite M yield good approximations for $M = \infty$. In this section, 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.

In Figure 5 we considered the behavior of deterministic loop-back systems with $N = 5$ processes, for different parameters M and p. The results of the simulation are plotted in Figures 5(a)–5(c). Each of them includes: (1) The expected round duration λ_{det}, computed by the method presented in Section 4.2 for a deterministic loop-back system with $M = 4$, drawn as a constant function. (2) The simulation results of sequence $T_1(r)/r$, that is process 1's average round duration, normalized to the calculated λ_{det}, for rounds $1 \leqslant r \leqslant 150$, for two systems: one with parameter $M = 4$, the other with parameter $M = \infty$, both averaged over 1000 runs. Considering λ_{prob} instead of λ_{det} resulted in similar graphs.

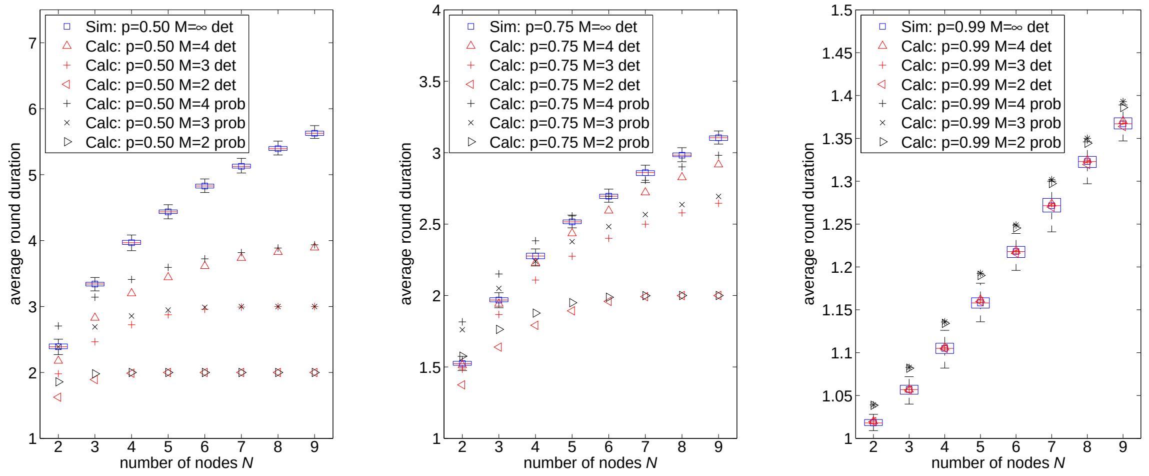


(a) $p = 0.5$

(b) $p = 0.75$

(c) $p = 0.99$

Figure 5: Simulated $T_1(r)/r$ versus r for $N = 5$ and $M = 4, \infty$ in deterministic loop-back systems with $p = 0.5, 0.75, 0.99$, normalized to $\lambda_{\text{det}}(p, 4, 5)$



(a) $p = 0.5$

(b) $p = 0.75$

(c) $p = 0.99$

Figure 6: λ_{prob}, λ_{det} for $M \leqslant 4$ and simulations (deterministic loop-backs, $M = \infty$) versus N for $p = 0.5, 0.75, 0.99$

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 $p \geqslant 0.75$, while for practically relevant $p \geqslant 0.99$ one cannot distinguish the finite from the infinite case.

In Figure 6 we compared the calculated values $\lambda_{\text{prob}}(p, M, N)$ and $\lambda_{\text{det}}(p, M, N)$ for $p = 0.5, 0.75, 0.99$, $N \leqslant 9$, and $M \leqslant 4$ to simulated values of $T_1(1000)/1000$ obtained from 100 Monte Carlo simulations of a deterministic loop-back system with $M = \infty$. The results of the simulation are depicted as box-plots. Note that for $p = 0.75$ the discrepancy between the analytic results for $\lambda_{\text{det}}(p, 4, N)$ and the simulation results for $M = \infty$ is already small, and for $p = 0.99$ the analytic results for all choices of M are in-between the lower quartile and the upper quartile of the simulation results.

6. Related work

The notion of simulating a stronger system on top of a weaker one is common in the field of distributed computing [2, Part II]. For instance, Neiger and Toueg [13] provide an 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 [11] use the simulation of a round structure on top of a partially synchronous system, and Charron-Bost and Schiper [5] 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 [14], 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 [15] 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 [3] 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 [6] extended Awerbuch's analysis by assuming message delays to be negligible, and a process i's processing time 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 [17] 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 [9] 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 [18], which is known to be a hard problem (e.g., [19]). In particular, our results can be translated into novel results on stochastic max-plus systems.

7. Conclusion

In this paper, we considered a retransmission-based algorithm that simulates a perfect round structure on top of a system with probabilistic message loss: Every message has probability p to arrive at its destination.

We devised a method, based on Markov theory, for calculating the exact value of a process i's expected round duration $\lambda = \mathbb{E} \lim_{r\to\infty} T_i(r)/r$, which was only known for a distributed system of size $N = 2$ until now. The running time of our method is polynomial in N, the number of processes. We further showed that $T_i(r)/r$ converges to λ with probability 1 and presented analytical bounds on the convergence speed.

While this approach is applicable to finite M only, simulations suggest that distributed systems with small values of M already yield very good approximations (with respect to the expected round duration) of the distributed system in which the number of retransmissions until a message is correctly received is not bounded.

A direct application of our results is that the computed expected round durations correspond to a lower bound on the expected rate of time-optimal algorithms that solve state-machine replication [20, 21, 22, 8] in the probabilistic systems under consideration; for a single perfect round of $A(B)$ suffices to solve distributed consensus.

Acknowledgments

The authors would like to thank Martin Biely, Ulrich Schmid, and Martin Zeiner for helpful discussions.

References

[1] N.A. Lynch, Distributed Algorithms, Morgan Kaufmann, San Francisco, 1996.

[2] H. Attiya, J. Welch, Distributed Computing: Fundamentals, Simulations, and Advanced Topics, second ed., John Wiley & Sons, Chichester, 2004.

[3] B. Awerbuch, Complexity of network synchronization, J. ACM 32 (1985) 804–823.

[4] O. Bakr, I. Keidar, Evaluating the running time of a communication round over the Internet, in: 21st Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 2002.

[5] B. Charron-Bost, A. Schiper, The heard-of model: computing in distributed systems with benign faults, Distrib. Comput. 22 (2009) 49–71.

[6] S. Rajsbaum, M. Sidi, On the performance of synchronized programs in distributed networks with random processing times and transmission delays, IEEE T. Parall. Distr. 5 (1994) 939–950.

[7] L. Lamport, R. Shostak, M. Pease, The Byzantine generals problem, ACM T. Progr. Lang. Sys. 4 (1982) 382–401.

[8] F.B. Schneider, Implementing fault-tolerant services using the state machine approach: a tutorial, ACM Comput. Surv. 22 (1990) 299–319.

[9] J.A.C. Resing, R.E. de Vries, G. Hooghiemstra, M.S. Keane, G.J. Olsder, Asymptotic behavior of random discrete event systems, Stochastic Process. Appl. 36 (1990) 195–216.

[10] T. Nowak, M. Függer, A. Kößler, On the performance of a retransmission-based synchronizer, in: A. Kosowski, M. Yamashita (Eds.), 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO), LNCS 6796, Springer, Heidelberg, 2011, pp. 234–245.

[11] C. Dwork, N. Lynch, L. Stockmeyer, Consensus in the presence of partial synchrony, J. ACM 35 (1988) 288–323.

[12] S. Meyn, R.L. Tweedie, Markov Chains and Stochastic Stability, Springer, Heidelberg, 1993.

[13] G. Neiger, S. Toueg, Automatically increasing the fault-tolerance of distributed algorithms, J. Algorithm. 11 (1990) 374–419.

[14] M. Ben-Or, Another advantage of free choice: completely asynchronous agreement protocols, in: 2nd Annual ACM Symposium on Principles of Distributed Computing (PODC), ACM, New York, 1983.

[15] G. Bracha, S. Toueg, Asynchronous consensus and broadcast protocols, J. ACM 32 (1985) 824–840.

[16] S. Rajsbaum, Upper and lower bounds for stochastic marked graphs, Inform. Process. Lett. 49 (1994) 291–295.

[17] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice Hall, Englewood Cliffs, 1989.

[18] B. Heidergott, Max-Plus Linear Stochastic Systems and Perturbation Analysis, Springer, Heidelberg, 2006.

[19] F. Baccelli, D. Hong, Analytic expansions of max-plus Lyapunov exponents, Ann. Appl. Probab. 10 (2000) 779–827.

[20] B. Charron-Bost, F. Pedone, A. Schiper (Eds.), Replication: Theory and Practice, LNCS 5959, Springer, Heidelberg, 2010.

[21] L. Lamport, The implementation of reliable distributed multiprocess systems, Comput. Netw. 2 (1978) 95–114.

[22] B.W. Lampson, How to build a highly available system using consensus, in: Ö. Babaoglu, K. Marzullo (Eds.), 10th International Workshop on Distributed Algorithms (WDAG), LNCS 1151, Springer, Heidelberg, 1996, pp. 1–17.

Appendix A. Markov Chain Facts

A *Markov chain* is a stochastic process, i.e., a sequence $\big(X(r)\big)_{r\geqslant 0}$ of random variables, such that the value of $X(r)$ does not depend on the value of the full history $\big(X(0), X(1), \dots, X(r-1)\big)$, but only on the value of $X(r-1)$; more formally, $X(r)$'s conditional probability distribution for fixed values of $\big(X(0), \dots, X(r-1)\big)$ is the same as for the sole fixed value $X(r-1)$. Given the set $\mathcal{X}$ of possible values for $X(r)$ (its *state space*) and a distribution for $X(0)$, the Markov chain $\big(X(r)\big)$ is fully determined once we fix a *transition probability distribution* P, i.e., a collection $(P_X)_{X\in\mathcal{X}}$ of probability distributions on $\mathcal{X}$.

Let $X(r)$ be a Markov chain with state space $\mathcal{X}$. We say that $X(r)$ is *aperiodic* if, for every $X \in \mathcal{X}$, the integers in the set $\big\{r\colon \mathbb{P}\big(X(r) = X \mid X(0) = X\big) > 0\big\}$ are relatively prime. We say that $X(r)$ is *irreducible* if for all $X, Y \in \mathcal{X}$, there exists an r such that $\mathbb{P}\big(X(r) = Y \mid X(0) = X\big) > 0$. We say that $X(r)$ is *Harris recurrent* if, for every $X \in \mathcal{X}$, we have $\mathbb{P}\big(X(r) = X \text{ for infinitely many } r\big) = 1$.

Theorem A.1. *Let $X(r)$ be good Markov chain with state space $\mathcal{X}$ and stationary 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. [12, Theorem 17.0.1(i)] □

Theorem A.2. *Let $X(r)$ be a good Markov chain with finite state space $\mathcal{X}$ and stationary distribution π. Then there exists some ρ, $0 < \rho < 1$, such that for all $X \in \mathcal{X}$:*

$$\mathbb{P}\big(X(r) = X\big) = \pi(X) + O\left(\rho^r\right)$$

as $r \to \infty$.

Proof. [12, Theorem 13.0.1(i)], [12, Theorem 16.0.2(iii)] □