Research Report 9/2011, Technische Universität Wien, Institut für Technische Informatik April 18, 2011

On the Performance of a Retransmission-based Synchronizer

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

[1] LIX, Ecole polytechnique
[2] TU Wien

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.

*Partially supported by the FATAL project of the Austrian Science Fund (FWF).

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, 15]: 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 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 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.

1.1 Contributions

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.* [16]. (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 the proofs, some facts about Markov chains, and the details of the results on the convergence speed.

2 The Retransmission Scheme

We now 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 [9, 8]. 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 [2]. However, it does not use the acknowledgement 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* c_i which is assumed to be taken from a not necessarily finite set of possible local states $\mathcal{C}$. The *global state* of the distributed system is a collection of local states $(c_i)_{i\in\Pi}$. Processes communicate by message passing; a *message* m is taken from a possibly infinite set of messages $\mathcal{M}$. An *algorithm* A for the distributed system comprises the following parts:

(1) The *set of possible local states* $\mathcal{C}$, the *set of possible initial local states* $\mathcal{C}_0$, and the *set of possible messages* $\mathcal{M}$.

(2) A pair of functions $\left(\sigma_A^{(i)}, \gamma_A^{(i)}\right)$ for every process i: The *send function* $\sigma_A^{(i)}$ for every process i, is from $\mathcal{C}$ to $\mathcal{M}$, and maps a local state to the message to send. The *next state function* $\gamma_A^{(i)}$ for every process i, is from $\mathcal{C} \times 2^{\mathcal{M}\times\Pi}$ to $\mathcal{C}$, and maps a local state and a set $\mathcal{R} \subseteq \mathcal{M} \times \Pi$ of received messages, labeled with their respective sender, to the next local state.

Computation at processes is assumed to occur in sequences of events locally happening at the processes. In an event, a process atomically (1) receives a set of messages, (2) computes its next local state, and (3) sends (broadcasts) a message to all other processes. Formally an *event* is a tuple $(i, \mathcal{R})$, where i is a process and $\mathcal{R}$ is the set of messages, with their respective sender (i.e., $\mathcal{R} \subseteq \mathcal{M} \times \Pi$), that where 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 $E(i) = a_i(1), e_i(2), a_i(2), \ldots, e_i(k), a_i(k), \ldots$, such that (1) every $a_i(1)$ is an initial (local) state and (2) for each $k > 1$ with $e_i(k) = (i, \mathcal{R})$, it is $\gamma_A^{(i)}\big(a_i(k-1), \mathcal{R}\big) = a_i(k)$. 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 $(m, j) \in \mathcal{R}$ where $e_i(k) = (i, \mathcal{R})$. We further say that process i *sends* (broadcasts) message m in *step* k, if $\sigma_A^{(i)}\big(a_i(k)\big) = m$.

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 that 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 message m in infinitely many steps, then j receives m from i in infinitely many steps.

In the case of benign communication, which we consider in this work, it is reasonable to assume the *no message generation* axiom **NoGen**.

Further desirable axioms are that of *communication closedness* **CommClosed** [8],*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** and for each process i, $E(i)$ is infinite. A *fair-lossy execution* of an algorithm A is an admissible execution that satisfies axioms **FairLoss** and **PerfComm***. Likewise, a *perfect round execution* is an admissible execution that satisfies axioms **CommClosed** and **PerfComm**.

2.2 Simulating Perfect Round Executions

Our goal is to characterize 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 simulation relation. 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 $\mathcal{C}^{(B)}$ and $\mathcal{M}$ denote the set 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 is $\mathcal{C}^{(B)}$, 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:

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

(2) We define $E \restriction B$ to be the sequence arising from F when replacing each processor's first state, $a_i(1)$, by $a_i(1)[Bstate]$, and every but each processor's first state, $a_i(r)$, by the two elements $a_i(r)[Bevent]$, $a_i(r)[Bstate]$ 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, (1) $trigger = true$ in every initial state of A, (2) for every initial state $b_i(1)$ of B, there exists an initial state $a_i(1)$ of A such that $a_i(1)[Bstate] = b_i(1)$, and (3) for every fair-lossy execution E of A, execution $E \restriction B$ is a perfect round execution of B.

2.3 A Solution

We are now ready to state a retransmission based algorithm that simulates perfect round executions on top of fair-lossy ones, and formally prove it correct. We denote the value of variable X at process j by X^j.

For every algorithm B, consider algorithm $A = A(B)$ as presented in Figure 1. The idea for the simulation is simple: Each process steadily 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 waits in round Rnd until it has received all other processes' round Rnd messages. When it does, it starts (simulated) round $Rnd + 1$.

1: **VAR** $BState \leftarrow b_i(0)$; $trigger \leftarrow true$; $Bevent \leftarrow \perp$;
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 $\mathcal{R}$
4: **for** received message $(r, m, m_{old}) \in \mathcal{R}$ from process j **do**
5: $Rcv[j, r] \leftarrow m$;
6: $Rcv[j, r-1] \leftarrow m_{old}$;
7: **end for**
8: $trigger \leftarrow false$;
9: **if** for all j in Π: $Rcv[j, Rnd] \neq \perp$ **then**
10: $Bstate_{old} \leftarrow Bstate$;
11: $trigger \leftarrow true$;
12: $\mathcal{R}' \leftarrow \{(Rcv[j, Rnd], j) \mid j \in \Pi\}$;
13: $Bevent \leftarrow (i, \mathcal{R}')$;
14: $Bstate \leftarrow \gamma_B\big(Bstate, \mathcal{R}'\big)$;
15: $Rnd \leftarrow Rnd + 1$;
16: **end if**
17: **end next state function**

18: **send function**
19: broadcast $(Rnd, \sigma_B(Bstate), \sigma_B(Bstate_{old}))$;
20: **end send function**

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

Proposition 1. *In every fair-lossy execution of $A(B)$ holds: If there exists a process i such that Rnd^i is bounded by K for some $K \geqslant 1$, then Rnd is bounded by $K + 1$ on all processes.*

Proposition 2. *In every fair-lossy execution of $A(B)$ holds: If each process has $Rnd = K$ in one of its steps, then each process has $Rnd = K + 1$ in one of its steps.*

Proposition 3. *In every fair-lossy execution of $A(B)$, variable Rnd is unbounded on every process.*

From Propositions 1–3 we imediately 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.*

3 Round Durations under Probabilistic Message Loss

We have presented a simple algorithm to simulate perfect rounds on top of fair-lossy executions, and will next 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 $Rnd^i = r-1$ to $Rnd^i = r$ was triggered; formally, $T_i(r) = k$ if $E(i) = a_i(1), e_i(2), a_i(2), \ldots$ and k is the smallest index such that $a_i(k)[Rnd] = r$. $L(r)$ 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.[1]

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 (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. We thus obtain the following proposition relating the starts of the simulated rounds:

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

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

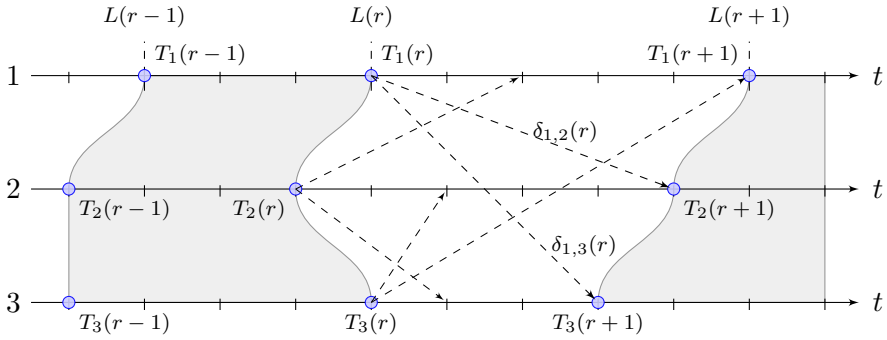


Figure 2: Fair-lossy execution of $A(B)$

Figure 2 depicts part of a fair-lossy execution of algorithm $A(B)$. Messages from process i to i are not depicted as they are always assumed to be received in the next step.

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: 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*.[2]

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 $i \neq j$, the probability that $\delta_{j,i}(r) = z$ is $(1-p)^{z-1} \cdot p$. Note that $\delta_{i,i}(r) = 1$, because of **PerfComm***.

For computational purposes we introduce the probability distribution **ProbLoss**(p, M), where $M \in \mathbb{N} \cup \{\infty\}$, on the set of fair-lossy executions, which is obtained from **ProbLoss**(p) by slightly changing the distribution of the $\delta_{j,i}(r)$: in contrast to **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 $i \neq j$, 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, **ProbLoss**(p, ∞) = **ProbLoss**(p).

[1] Relating the round duration to Newtonian real-time is thus reduced to relating the duration of a single step to real-time.

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

We will see in Sections 4.3 and 5, that the error we make when calculating the expected duration of the simulated rounds in **ProbLoss**(p, M) with finite M instead of **ProbLoss**(p) is small, even for small values of 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 **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 an algorithm 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. Due to space limitations, most of the proofs are postponed to Appendix A; supplemental facts and definitions about Markov chains can be found in Appendix B.

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 $T(r)$. 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 5.

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 the set X, and set

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

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 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.*

In fact Proposition 6 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'.[3]

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 $\pi(X) = \sum_{Y \in \mathcal{X}} \pi(Y) \cdot P_{X,Y}$ 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.[4]

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

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 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 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, $\lambda = \sum_{z=1}^{M} z \cdot \pi(\mathcal{L}_z) < \infty$.*

[3] This is the case of "negligible transmission delays" considered by Rajsbaum and Sidi [15].

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

4.2 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 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_{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 **PerfComm*** holds. Later, we explain how to arrive at a formula for P in the case of the (more realistic) assumption of **PerfComm***.

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 2, it holds that $\Lambda(r) = (0, 1, 2, 0, 0)$. Normalization, that is, right alignment of the last processes, yields $\mathrm{Norm}(\Lambda(r)) = (0, 0, 0, 1, 2)$.

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:

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

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 $Y = (\sigma'_1, \ldots, \sigma'_M)$,

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

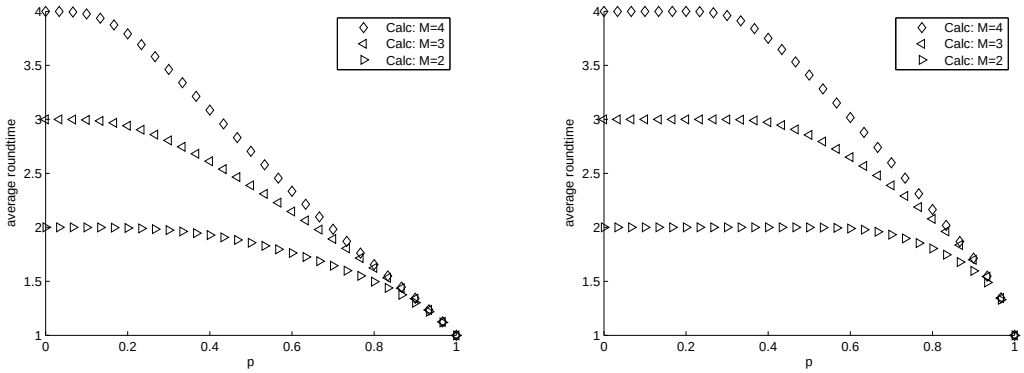
Note that for a system where **PerfComm*** holds, in Equation (4), one has the account for the fact that a process i definitely receives its own message after 1 step. In order to specify a transition probability analogous to Equation (4), it is thus necessary to know to which of the $\sigma_k(r)$ in $\Lambda(r)$, 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:

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

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

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

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



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

Figure 4: λ versus p in a system with different choices of N

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 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:

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

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 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 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 algorithm.

4.3 Results

The presented algorithm allows to obtain analytic expressions for λ for fixed N and M in terms of probability p. Figure 3 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 4(a) and 4(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 5 by means of Monte Carlo simulation.

4.4 Rate of Convergence

Theorem 2 states that $L(r)/r$ converges to λ with probability 1, however does not give a rate of convergence. We now present lower bounds on the speed of this convergence. Detailed proofs can be found in Appendix C.

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 Theorem 5 in the appendix.)

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

5 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 4.2, 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.

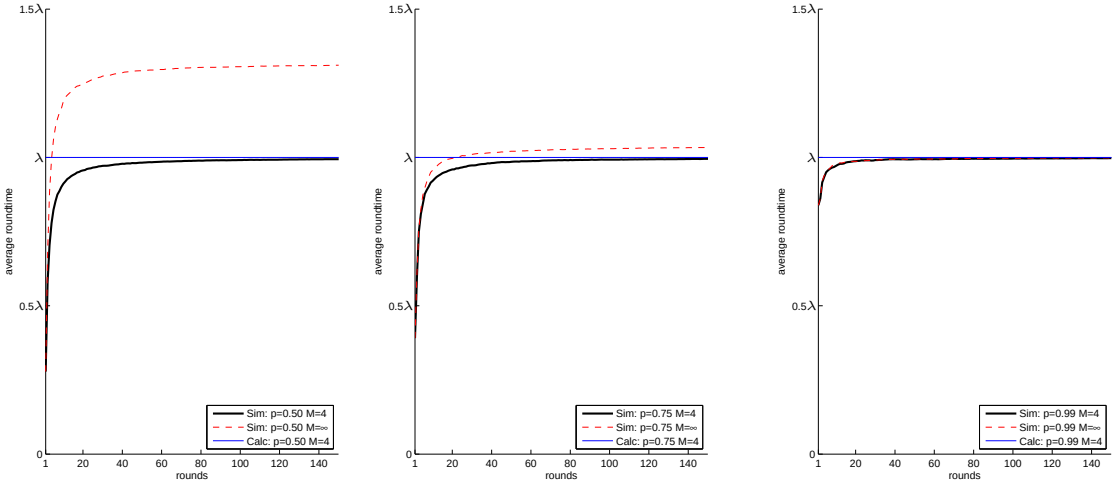
We considered the behavior of a system of $N = 5$ processes, for different parameters M and p. The results of the simulation are plotted in Figures 5(a)–5(c). Each of them shows: (1) The expected round duration λ, computed by the algorithm presented in Section 4.2 for a system with $M = 4$, drawn as 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 6(a)–6(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 in-between the lower quartile and the upper quartile of the simulation results.

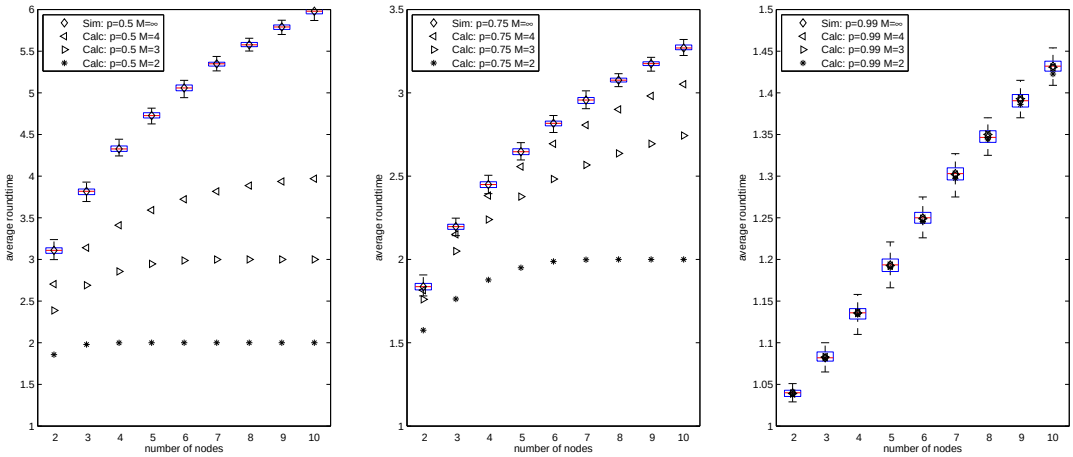
6 Related work

The notion of simulating a stronger system on top of a weaker one is common in the field of distributed computing [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.



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

Figure 5: $T_1(r)/r$ versus r in systems with different p



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

Figure 6: λ versus N in systems with different p

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 [15] 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. 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 [16] 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.

7 Conclusion

The paper considers 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.

The main contribution is 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 asymptotically bounded by $\binom{M+N-1}{M-1}^3$ where N is the number of processes and M is the maximum number of tries until a message is correctly received. We further show that each process i's $T_i(r)/r$ converges to λ with probability 1 and present analytical bounds on the convergence speed.

While this approach is applicable to finite M only, we further show 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.

In future work, we will consider non-fully connected communication networks and non-homogeneous probabilities of correct transmission.

Acknowledgements

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

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] 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)

[16] 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)

A Proofs

Proof of Proposition 1. Let i be a process such that $Rnd^i \leqslant K$ during the whole execution. Then, by code line 19, i never sends a message of the form (r, m, m') with $r > K$. By **NoGen**, no process receives a message of the form (r, m, m') with $r > K$ from process i. Hence all processes always have $Rcv[i, r] = \perp$ for all $r > K$, and, by lines 9 and 15, do not set Rnd to a higher value than $K + 1$. ■

Proof of Proposition 2. By contradiction: Suppose that there exists some process i such that always $Rnd^i \leqslant K$. By Proposition 1, $Rnd \leqslant K + 1$ on all processes. Hence by code line 19 and the fact that every process takes infinitely many steps, it follows that every process sends a message of the form (r, m, m') infinitely often where $r \in \{K, K + 1\}$. By **FairLoss**, all of these messages are received at least once. Then, by code line 19 and 4–7, all $Rcv^i[j, K] \neq \perp$ at

process i during some step of the execution. But then, by code line 15, also $Rnd^i = K + 1$. Contradiction. ■

Proof of Proposition 3. This is an immediate consequence of Proposition 2. ■

Proof of Theorem 1. It remains to show that (1) $E \restriction B$ is an execution of B and (2) $E \restriction B$ is perfect whenever E is fair-lossy. Property (1) mainly follows from code lines 8, 11, and 12–14. Property (2) follows from code line 9 and Proposition 3. ■

Proof of Proposition 6. On the set of collections (x_i) of reals 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 Λ_r, given $\Lambda(1) = \Lambda_1$, $\ldots$, $\Lambda(r-1) = \Lambda_{r-1}$, is the same as the conditional distribution for Λ_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. ■

Proof of Proposition 7. $\Lambda(r)$ is aperiodic because every state can be reached from every other in two and in three steps with nonzero probability. 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 [13, Theorem 10.0.1]. ■

Proof of Theorem 2. We use Theorem 3 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 4 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 $\exists i, j : (i \neq j) \wedge (\delta_{i,j}(r) \geqslant z)$, hence

$$\mathbb{P}\big(\sigma(r) = z\big) \leqslant 1 - \big(1 - (1-p)^{z-1}\big)^{N(N-1)} \tag{8}$$

for all $r \geqslant 1$.

We now manipulate the right-hand side of Equation (8) with operations that preserve the inequality. First, we substitute $N(N-1)$ by N^2. We then 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. ■

B 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 $\big(X(0), X(1), \ldots, X(r-1)\big)$, but only on the value of $X(r-1)$; more formally, $X(r)$'s conditional probability distribution for fixed values of $\big(X(0), \ldots, X(r-1)\big)$ 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 3. *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,*

$$\lim_{r\to\infty} \frac{1}{r} \sum_{k=1}^{r} g\big(X(k)\big) = \sum_{X\in\mathcal{X}} g(X) \cdot \pi(X)$$

almost surely for every initial distribution.

Proof. [13, Theorem 17.0.1(i)] ■

Theorem 4. *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. [13, Theorem 13.0.1(i)], [13, Theorem 16.0.2(iii)] ■

C 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 5).

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 4, 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 a convergence rate of $O(\log\log r/r)$ [13, 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 $k < \ell$.

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

$$\sum_{z=1}^{M} \sum_{u=1}^{M} z \cdot u \cdot \mathbb{P}\big(\Lambda(k) \in \mathcal{L}_z \wedge \Lambda(\ell) \in \mathcal{L}_u\big). \tag{9}$$

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

$$\sum_{\Lambda \in \mathcal{L}_z} \mathbb{P}\big(\Lambda(k) = \Lambda\big) \cdot \mathbb{P}\big(\Lambda(\ell) \in \mathcal{L}_u \mid \Lambda(k) = \Lambda\big). \tag{10}$$

Theorem 4 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 (10), and noting both $\pi(\mathcal{L}_z) = \sum_{\Lambda \in \mathcal{L}_z} \pi(\Lambda)$ and Theorem 2, yields that (9) 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 $k < \ell$.

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 $k < 2\ell$ and $C(r)$ to be the set of pairs (k,ℓ) in $A(r)$ that satisfy $k \geqslant 2\ell$. 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,

$$\sum_{(k,\ell) \in B(r)} a(k,\ell) \leqslant \sum_{\ell=1}^{r} \sum_{k=1}^{r} \rho^k. \tag{11}$$

We calculate $\sum_{k=1}^{r} \rho^k = (\rho - \rho^{r+1})/(1-\rho)$, which immediately implies that the right-hand sum in (11) is $O(r)$.

Similarly,

$$\sum_{(k,\ell)\in C(r)} a(k,\ell) \leqslant \sum_{\ell=1}^{r}\sum_{k=1}^{\ell} \rho^{\ell-k} \leqslant \sum_{\ell=1}^{r}\sum_{k=1}^{r} \rho^{k} \tag{12}$$

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. The inequality can be stated as $\mathbb{P}\bigl(|X - \mathbb{E}X| \geqslant A\bigr) \leqslant (\operatorname{Var} X)^2/A^2$.

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 5. *If $\alpha_r = \Omega(r^{-1+\varepsilon})$ for some $\varepsilon > 0$, 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, 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 $\alpha_r = \Omega(r^{-1+\varepsilon})$. ■