Model Checking Lossy Channels Systems Is Probably Decidable

N. Bertrand and Ph. Schnoebelen

Lab. Spécification & Vérification
ENS de Cachan & CNRS UMR 8643
61, av. Pdt. Wilson, 94235 Cachan Cedex France
email: {bertrand|phs}@lsv.ens-cachan.fr

Abstract. Lossy channel systems (LCS's) are systems of finite state automata that communicate via unreliable unbounded fifo channels. We propose a new probabilistic model for these systems, where losses of messages are seen as faults occurring with some given probability, and where the internal behavior of the system remains nondeterministic, giving rise to a reactive Markov chains semantics. We then investigate the verification of linear-time properties on this new model.

1 Introduction

Verification of channel systems. Channel systems [BZ83] are systems of finite state automata that communicate via asynchronous unbounded fifo channels. They are a natural model for asynchronous communication protocols, used as the semantical basis of protocol specification languages such as SDL and Estelle. *Lossy channel systems* [Fin94,AJ96b] are a special class of channel systems where messages can be lost while they are in transit, without any notification. These lossy systems are the natural model for fault-tolerant protocols where the communication channels are not supposed to be reliable.

Surprisingly, while channel systems are Turing-powerful [BZ83], several verification problems become decidable when one assumes channels are lossy: reachability, safety properties over traces, inevitability properties over states, and fair termination are decidable for lossy channel systems [Fin94,CFP96,AJ96b,MS02].

This does not mean that lossy channel systems are an artificial model where, since no communication can be fully enforced, everything becomes trivial. To begin with, many important problems are undecidable: recurrent reachability properties are undecidable [AJ96a], so that model checking of liveness properties is undecidable too. Furthermore, boundedness is undecidable [May00], as well as all behavioral equivalences [Sch01]. Finally, none of the decidable problems listed in the previous paragraph can be solved in primitive recursive time [Sch02]!

Probabilistic losses. When modeling real-life protocols, it is natural to see message losses as some kind of faults having a probabilistic behavior. This idea led to the introduction of a Markov chain model for lossy channel systems [PN97].

Essentially the same model allowed Baier and Engelen to show that qualitative model checking is decidable, i.e. it can be decided whether a linear-time property holds *almost surely*, that is, with probability 1 [BE99]. This is a smart way of using randomization to circumvent the undecidability of temporal model checking in the non-probabilistic case. However, this result has several limitations: (l1) it requires that the channel system itself is seen as choosing probabilistically between its transitions, (l2) it assumes that there is a fixed probability p that "the current step is a loss", and (l3) it only gives decidability for $p \geq 0.5$, an unrealistically large value (using a slightly different model, [ABPJ00] shows that decidability is lost if p is not large enough).

Our contribution. We propose an improved approach that addresses the above-mentioned limitations. Our first idea is to use a more realistic probabilistic model for losses, where *any message* has a fixed probability $\tau > 0$ of being lost during the current step, independently of other messages possibly in transit at the same time. We call it the *local-fault model* (and refer to the proposal by [PN97] as the *global-fault model*). In our local-fault model, qualitative model checking is decidable whatever the value of τ (thus our solution to limitation (l2) solves (l3) as well).

Our second idea attacks limitation (l1): we move from Markov chains to *reactive Markov chains* (or, equivalently, *Markovian decision processes*) as the probabilistic model for lossy channel systems: this allows combining a probabilistic behavior for losses with a *nondeterministic* behavior for the channel system. The verification problems we investigate are whether a linear-time property holds almost surely *under any scheduling policy* (the *adversarial* viewpoint). We show that, while the problem is undecidable in general, there exist some decidable subcases (natural subsets of temporal properties). Furthermore, the problem becomes decidable when we restrict ourselves to *finite-memory* scheduling policies only. Finally, it turns out that these verification problems are insensitive to the precise value of the fault rate τ.

Since our decision procedures reduce probabilistic model checking to the kind of reachability questions that have been successfully verified in practice (e.g. [AAB99]), we believe our ideas will provide a nice way of verifying liveness properties on channel systems with probabilistic losses: the approximations "almost surely" and "under any finite-memory scheduling policy" are very reasonable and only retract minimally from the rigid "surely" and "for all scheduling policies" that are the standard goals in algorithmic verification.

Related work. Verifying probabilistic lossy channel systems combines issues from the verification of infinite-state systems and from the verification of probabilistic systems [1]. These two fields are technically quite involved and it seems that, to date, the only joint instance that has been investigated are the probabilistic lossy systems. We already explained how our work is a continuation

[1] Here we do not mean systems where the *timings* are probabilistic like, for example, continuous time Markov chains [BKH99].

of [PN97,BE99,ABPJ00] and depart from these earlier papers. The local-fault model has been independently proposed by Abdulla and Rabinovich [AR03] who proved a result essentially equivalent to our Theorem 5.4 (but did not investigate adversarial verification).

Outline of the paper. Section 2 sets the necessary background on the verification of infinite Markov chains. Channel systems are presented in Section 3, before we discuss probabilistic losses in Section 4 and study probabilistic lossy systems (PLCS's) in Section 5. Nondeterministic PLCS's are defined in Section 6 and their verification is studied in Section 7. For lack of space, many proofs have been omitted in this extended abstract: they can be found in the full version.

2 (Reactive) Markov chains and their verification

We assume some familiarity with Markov chains and only introduce the notations we need in the rest of the paper (we mostly follow [Var99]).

Definition 2.1. *A* Markov chain *is a tuple* $M = \langle W, P, P_0\rangle$ *of a countable set of* configurations $W = \{\sigma, \ldots\}$*, a* transition probability function $P : W^2 \mapsto [0, 1]$ *such that* $\sum_{\sigma' \in W} P(\sigma, \sigma') = 1$ *for all* $\sigma \in W$*, and an* initial probability distribution $P_0 : W \mapsto [0, 1]$.

M is *bounded* when there exists $e > 0$ s.t. $P(\sigma, \sigma') > 0$ entails $P(\sigma, \sigma') \geq e$ (i.e. probabilities are not arbitrarily low). M is *finite* when W is. Finite Markov chains are bounded.

A *run* of M is an infinite sequence $\pi \in W^\omega$ of configurations. The set of runs W^ω is turned into a probability space in the standard way: the measure μ of events is first defined on basic cylinders with:

$$\mu(\{\pi \mid \pi \text{ starts with } \sigma_0, \sigma_1, \ldots, \sigma_n\}) \stackrel{\text{def}}{=} P_0(\sigma_0)P(\sigma_0, \sigma_1) \cdots P(\sigma_{n-1}, \sigma_m) \quad (1)$$

and is then extended to the Borel field they generate (see [Var99,Pan01]).

Underlying any Markov chain M is the transition system (the directed graph) G_M where there is a transition $\sigma \rightarrow \sigma'$ iff $P(\sigma, \sigma') > 0$. This explains why we often rely on standard graph-theoretic terminology and write statements like "σ is reachable from σ_0", etc., for notions that do not depend on the precise values of the transition probability function. E.g. the measure (1) is non-zero iff σ_0 is a possible initial configuration and $\sigma_0 \rightarrow \sigma_1 \rightarrow \ldots \rightarrow \sigma_n$ is a path in G_M.

2.1 Reactive Markov chains

Reactive Markov chains [Var99], called "concurrent Markov chains" in [Var85,HSP83], were introduced for modeling systems whose behavior has both probabilistic and nondeterministic aspects. They are a special (and equivalent) form of *Markovian decision processes* [Der70], where the system nondeterministically picks what will be its next step, and the outcome of that step follows some probability law.

Definition 2.2. *A* reactive Markov chain *(a RMC) is a tuple* $M = \langle W, N, P, P_0 \rangle$ *s.t.* $\langle W, P, P_0 \rangle$ *is a Markov chain, and* $N \subseteq W$ *is the subset of* nondeterministic configurations.

The configurations in $W \setminus N$ are called *probabilistic*. For a nondeterministic σ, the exact value of $P(\sigma, \sigma') > 0$ has no importance (apart from being positive): it just means that, when in σ, σ' is a possible next configuration.

The behavior of a RMC $M = \langle W, N, P, P_0 \rangle$ is driven by the nondeterministic choices and the probabilistic behavior. This is formalized by introducing the notion of a *scheduler* (also called *adversary*, or *(scheduling) policy*) that is responsible for the nondeterministic choices. Formally, a scheduler for M is a mapping $u : W^*N \to W$ such that $u(\sigma_0 \ldots \sigma_n) = \sigma'$ implies $P(\sigma_n, \sigma') > 0$. The intuition is that, when the system is in a nondeterministic configuration σ_n, u selects a next configuration σ' among the allowed ones, based on the history $\sigma_0 \ldots \sigma_n$ of the computation (we do not consider more general notions of adversaries).

Combining a RMC M with a scheduler u gives a *bona fide* Markov chain $M^u = \langle W^+, P^u, P_0^u \rangle$ describing the stochastic behavior of M against u. Intuitively, M^u is obtained by unfolding M into a tree, with W^+ the set of non-empty histories [2], and pruning branches that do not obey u. Formally, for any $x \in W^+$

$$P^u(x\sigma, x\sigma\sigma') \stackrel{\text{def}}{=} \begin{cases} P(\sigma, \sigma') & \text{if } \sigma \notin N, \\ 1 & \text{if } \sigma \in N \text{ and } u(x\sigma) = \sigma', \\ 0 & \text{otherwise,} \end{cases}$$

and $P^u(x\sigma, y\sigma') = 0$ when $y \neq x\sigma$. Finally, P_0^u is like P_0 on histories having length 1, and zero on longer histories. It is readily verified that M^u is indeed a Markov chain.

2.2 Verification for Markov chains

We address verification of linear-time properties that can be expressed in temporal logic (TL), or second-order monadic logic on runs (MLO), and that do not refer to quantitative information.

Classically such properties can be given under the form of a Büchi automaton that recognizes exactly the correct runs, so that TL model checking reduces to repeated reachability of control states in a product system. This approach does apply to Markov chains if the property is represented by a *deterministic* ω-automaton: then the product system is again a Markov chain.

Since deterministic Büchi automata are not expressive enough for TL or MLO, we shall assume the properties are given by deterministic Street automata. Then, in order to check TL or MLO properties on Markov chains, it is enough to be able to check simpler behavioral properties of the form

[2] When describing the behavior of some M^u, it is customary to leave the histories implicit and only consider their last configuration: this informal way of speaking makes M^u look more like M.

$\alpha = \bigwedge_{i=1}^{n}(\Box\Diamond A_i \Rightarrow \Box\Diamond A'_i)$ where, for $i = 1, \ldots, n$, $A_i, A'_i \subseteq W$ (i.e. α is a Street acceptance condition). A run $\pi = \sigma_0, \sigma_1, \ldots$ satisfies such a condition, written $\pi \models \alpha$, if for all $i = 1, \ldots, n$, either $\sigma_j \in A_i$ for finitely many j, or $\sigma_j \in A'_i$ for infinitely many j. The following is standard:

Theorem 2.3. *Let M be a countable Markov chain and α be a Street acceptance condition. Then $\{\pi \mid \pi \models \alpha\}$ is measurable.*

We let $\mu_M(\alpha)$ denote this measure and say that M satisfies α *with probability* p when $\mu_M(\alpha) = p$. We often consider the probability, written $\mathbb{P}(M, \sigma \models \alpha)$ or $\mu_\sigma(\alpha)$, that a given configuration σ satisfies a property α: this is defined as $\mu_{M'}(\alpha)$ for a Markov chain M' obtained from M by changing the initial distribution.

We say that M satisfies α *almost surely* (resp. *almost never*, *possibly*) when M satisfies α with probability 1 (resp. with probability 0, with probability $p > 0$).

Remark 2.4. These notions are inter-reducible: M satisfies α almost surely iff it satisfies $\neg\alpha$ almost never iff it is not the case that it satisfies $\neg\alpha$ possibly. $\Box$

2.3 Verification for Markov chains with a finite attractor

Verifying that a *finite* Markov chain almost surely satisfies a Street property is decidable [CY95,Var99]. However, the techniques involved do not always extend to *infinite* chains, in particular to chains that are not bounded.

It turns out it is possible to extend these techniques to countable Markov chains *where a finite attractor exists*. We now develop these ideas, basically by simply streamlining the techniques of [BE99]. Below we assume a given Markov chain $M = \langle W, P, P_0 \rangle$.

Definition 2.5 (Attractors). *A non-empty set $W_a \subseteq W$ of configurations is an* attractor *when*

$$\mathbb{P}(M, \sigma \models \Box\Diamond W_a) = 1 \textit{ for all } \sigma \in W \tag{2}$$

The attractor is finite *when W_a is.*

Assume $W_a \subseteq W$ is a finite attractor. We define $G_M(W_a)$ as the finite directed graph $\langle W_a, \rightsquigarrow \rangle$ where the vertices are the configurations from W_a and where there is an edge $\sigma \rightsquigarrow \sigma'$ iff, in M, σ' is reachable from σ by a non-empty path. Observe that the edges in $G_M(W_a)$ are transitive.

In $G_M(W_a)$, we have the usual graph-theoretic notion of (maximal) strongly connected components (SCC's), denoted $B, B', \ldots$. A trivial SCC is a singleton without the self-loop. These SCC's are ordered by reachability and a minimal SCC (i.e. an SCC B that cannot reach any other SCC) is a *bottom SCC* (a BSCC). Observe that, in $G_M(W_a)$, a BSCC B cannot be trivial: since W_a is an attractor, one of its configurations must be reachable from B.

For a run π in M, we write $\lim_{W_a}(\pi)$ for the sets of configurations from W_a that appear infinitely often in π. Necessarily, if $\lim_{W_a}(\pi) = A$ then the configurations in A are inter-reachable and A is included in some SCC of $G_M(W_a)$.

Lemma 2.6. *If $\mu_\sigma(\{\pi \mid \lim_{W_a}(\pi) = A\}) > 0$ then A is a BSCC of $G_M(W_a)$.*

Assume the BSSC's of $G_M(W_a)$ are $B_1, \ldots, B_k$. Lemma 2.6 and Eq. (2) entail

$$\mu_\sigma(\{\pi \mid \lim_{W_a}(\pi) = B_1\}) + \cdots + \mu_\sigma(\{\pi \mid \lim_{W_a}(\pi) = B_k\}) = 1. \tag{3}$$

Therefore, for a BSCC B, $\sigma \in B$ entails $\mu_\sigma(\{\pi \mid \lim_{W_a}(\pi) = B\}) = 1$. Hence $\mu_{\sigma_0}(\{\pi \mid \lim_{W_a}(\pi) = B\}) > 0$ iff B is reachable from σ_0.

It is now possible to reduce the probabilistic verification of Street properties to a finite number of reachability questions:

Proposition 2.7. *Assume W_a is a finite attractor of M. Then for any $\sigma \in W$, $\mathbb{P}(M, \sigma \models \bigwedge_{i=1}^{n}(\Box\Diamond A_i \Rightarrow \Box\Diamond A'_i)) > 0$ iff there exists a BSCC B of $G_M(W_a)$ such that $\sigma \xrightarrow{*} B$ and, for all $i = 1, \ldots, n$ $B \xrightarrow{*} A_i$ implies $B \xrightarrow{*} A'_i$.*

2.4 Verification for reactive Markov chains

Verifying reactive Markov chains usually assumes an adversarial viewpoint on schedulers. Typical questions are whether, for all schedulers u, M^u satisfies α almost surely (resp. almost never, resp. possibly)? Cooperative viewpoints (asking whether for some u, M^u satisfies α almost surely ...) are possible but less natural in practical verification situations. We consider them since they appear through dualities anyway (Remark 2.4) and since presenting proofs is often easier under the cooperative viewpoint.

Technically, since we still use properties referring to states of W, one defines whether a path in M^u satisfies a property by projecting it from $(W^+)^*$ to W^* in the standard way [Var99].

One sometimes wants to quantify over a restricted set of schedulers, e.g. for checking that M almost surely satisfies α for all *fair* schedulers (assuming some notion of fairness) [HSP83,Var85]. Such a problem can usually be translated into an instance of the general adversarial problem by stating the fairness assumption in the α part.

However, not all restrictions can be transfered in the property to be checked. In particular we shall consider the restriction to *finite-memory* schedulers: this is a convenient way of ruling out infeasible or exaggeratedly malicious schedulers. Several definitions are possible: here we say that u is *finite-memory* if there is a morphism $h : W^* \to H$ that abstract histories from W^* into a finite monoid H and such that $u(\sigma_0 \ldots \sigma_n) = u'(h(\sigma_0, \ldots, \sigma_n), \sigma_n)$ for some $u' : H \times X \to W$. Thus H is the finite memory on which u', the true scheduler, is based. When H is a singleton, u is *memoryless.*

3 Channel systems

Perfect channel systems. In this paper we adopt the *extended* model of channel systems *where emptiness of channels can be tested for.*[3]

[3] Our *undecidability* proofs do not rely on the extension.

Definition 3.1 (Channel system). *A* channel system *(with m channels) is a tuple $S = \langle Q, C, \Sigma, \Delta, \sigma_0 \rangle$ where*
– $Q = \{r, s, \ldots\}$ *is a finite set of* control locations *(or* control states*),*
– $C = \{c_1, \ldots, c_m\}$ *is a finite set of m* channels*,*
– $\Sigma = \{a, b, \ldots\}$ *is a finite alphabet of* messages*,*
– $\Delta \subseteq Q \times Act_C \times Q$ *is a finite set of* rules*, where* $Act_C \stackrel{def}{=} (C \times \{?, !\} \times \Sigma) \cup (C \times \{= \varepsilon?\})$ *is a set of* actions *parameterized by C and Σ,*
– $\sigma_0 \in Q \times \Sigma^{*C}$ *is the* initial configuration *(see below).*

A rule $\delta \in \Delta$ of the form $(s, c, ?, a, r)$ (resp. $(s, c, !, a, r)$) is written "$s \xrightarrow{c?a} r$" (resp. "$s \xrightarrow{c!a} r$") and means that S can move from control location s to r by reading a from (resp. writing a to) channel c. Reading a is only possible if c is not empty and its first available message is a. A rule of the form $(s, c, = \varepsilon?, r)$ is written "$s \xrightarrow{c=\varepsilon?} r$" and means that S can move from s to r if channel c is empty.

Formally, the behavior of S is given via a transition system: a *configuration* of S is a pair $\sigma = \langle r, U \rangle$ where $r \in Q$ is a control location and $U \in \Sigma^{*C}$ is a *channel contents*, i.e. a C-indexed vector of Σ-words: for any $c \in C$, $U(c) = u$ means that c contains u. For $s \in Q$ we write $\Uparrow s$ for the set $\{s\} \times \Sigma^{*C}$ of all configurations based on s.

The possible moves between configurations are given by the rules of S. For $\sigma, \sigma' \in W$, we write $\sigma \xrightarrow{\delta}_{\text{perf}} \sigma'$ ("perf" is for *perfect steps*) when:

Reads: $\delta \in \Delta$ is some $s \xrightarrow{c?a} r$, σ is some $\langle s, U \rangle$, $U(c)$ is some $a_1 \ldots a_n$ with $a_1 = a$, and $\sigma' = \langle r, U\{c \mapsto a_2 \ldots a_n\} \rangle$ (using the standard notation $U\{c \mapsto u'\}$ for variants).

Writes: $\delta \in \Delta$ is some $s \xrightarrow{c!a} r$, σ is some $\langle s, U \rangle$, $U(c)$ is some $u \in \Sigma^*$, and $\sigma' = \langle r, U\{c \mapsto u.a\} \rangle$.

Tests: $\delta \in \Delta$ is some $s \xrightarrow{c=\varepsilon?} r$, σ is some $\langle s, U \rangle$, $U(c) = \varepsilon$, and $\sigma' = \langle r, U \rangle$.

Idling: Finally, we have idling steps $\sigma \xrightarrow{0}_{\text{perf}} \sigma$ in any configuration.

We write $En(\sigma)$ for the set of rules *enabled* in configuration σ. We consider idling as a rule and have $0 \in En(\sigma)$ for all σ. For $\delta \in En(\sigma)$, we further write $Succ_\delta(\sigma)$ to denote the (unique) successor configuration σ' obtained by applying δ on σ. We often omit the superscript δ in steps and only write $\sigma \to_{\text{perf}} \sigma'$.

Remark 3.2. Allowing the idling rule is a definitional detail that smoothes out Definitions 5.1 and 6.1 (deadlocks are ruled out). It also greatly simplifies the technical developments of section 7 (the possibility of idling gives more freedom to scheduling policies). □

Lossy channel systems. In the standard lossiness model, a lossy step is a perfect step possibly preceded and followed by arbitrary message losses. Here we allow losses only *after* the perfect step. This simplifies the construction of the probabilistic model and does not modify the semantics in any essential way [4] unlike, e.g., the notion of front-lossiness used in [Fin94,CFP96,ABPJ00,Sch01].

Formally, we write $u \sqsubseteq v$ when u is a subword of v, i.e. u can be obtained by erasing any number of letters (possibly zero) from v. When $u \sqsubseteq v$, it will be useful to identify the set $\rho \subseteq \{1, \ldots, |v|\}$ of positions in v where letters have been erased, and we use the notation "$u \sqsubseteq_\rho v$" for that purpose. E.g. $\texttt{aba} \sqsubseteq_{\{1,2,5\}} \texttt{baabba}$. Observe that, in general, $u \sqsubseteq v$ can be explained by several distinct erasures $\rho, \rho', \ldots$

The subword ordering extends to channel contents and to channel systems configurations in the standard way:

$$U \sqsubseteq V \stackrel{\text{def}}{\Leftrightarrow} U(c) \sqsubseteq V(c) \text{ for all } c \in C,$$
$$\langle r, U\rangle \sqsubseteq \langle s, V\rangle \stackrel{\text{def}}{\Leftrightarrow} r = s \text{ and } U \sqsubseteq V.$$

Erasures extend too: we still write $U \sqsubseteq_\rho V$ but now $\rho \subseteq C \times \mathbb{N}$.

It is now possible to define the lossy steps of channel systems: we write $\sigma \xrightarrow{\delta}_{\text{loss}} \sigma'$ when $\sigma' \sqsubseteq \sigma''$ for some σ'' s.t. $\sigma \xrightarrow{\delta}_{\text{perf}} \sigma''$. Perfect steps are a special case of lossy steps (they have $\sigma' = \sigma''$). Below we omit writing explicitly the "loss" subscript for lossy steps, and are simply careful of writing $\rightarrow_{\text{perf}}$ for all perfect steps.

As usual, $\sigma \xrightarrow{*} \sigma'$ denotes that σ' is *reachable* from σ. We write $\sigma \xrightarrow{+} \sigma'$ when σ' is reachable via a non-empty sequence of steps. The *reachability problem for lossy channel systems* is, given S, σ and σ', to say if σ' is reachable from σ in S. It is know that this problem is decidable (even if testing channels for emptiness is allowed) [AJ96b,CFP96,May00].

A set $A \subseteq W$ of configurations is *reachable from* σ if some $\sigma' \in A$ is. This is denoted $\sigma \xrightarrow{*} A$. One can decide whether $\sigma \xrightarrow{*} A$ just by looking at the minimal elements of A. Since $\sqsubseteq$ is a wqo, any $A \subseteq W$ only has a finite number of minimal elements. Therefore it is decidable whether $\sigma \xrightarrow{*} A$.

4 The local-fault model for probabilistic losses

Earlier proposals for probabilistic lossy channels assume there is a fixed probability that the next step is the loss of a message [PN97,BE99]. We argued in the introduction that this model is not very realistic. We prefer a viewpoint where the fixed fault rate is associated with every single message. Then, the probability that a given message is lost at the next step is not influenced by the presence or identity of other messages.[5]

[4] The modification only has to do with where we separate a step from its predecessor and successor steps, i.e. with the granularity of the operational semantics.

[5] This agrees with the actual behavior of many lossy fifo links where each message is handled individually by various components (switches, routers, buffers, . . .). Admit-

Formally, we assume given a fixed *fault rate* $\tau \in [0,1]$ that describes the probability that any given message will be lost during the next step. From τ, one derives $p_\tau(U,U')$, the probability that channels with contents U will have contents U' after one round of probabilistic losses:

$$p_\tau(U,U') \;=\; \sum_{\rho \text{ s.t. } U' \sqsubseteq_\rho U} \tau^{|\rho|}(1-\tau)^{|U'|}. \tag{4}$$

Then $p_\tau(U,U') > 0$ iff $U' \sqsubseteq U$ (assuming $0 < \tau < 1$), and $\sum_{U'} p_\tau(U,U') = 1$ for any U. It will be convenient to extend this probability distribution from channel contents to configurations with

$$p_\tau(\langle s,U\rangle,\langle r,V\rangle) \stackrel{\text{def}}{=} \begin{cases} p_\tau(U,V) & \text{if } s=r, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Example 4.1. Assume we have a single channel c that contains $u = \mathtt{aab}$. Assume $\tau = 0.1$. Then

$$\begin{array}{llll} p_\tau(\mathtt{aab},\varepsilon) = \tau^3 = 0.001 & & p_\tau(\mathtt{aab},\mathtt{aa}) = \tau(1-\tau)^2 = 0.081 \\ p_\tau(\mathtt{aab},\mathtt{b}) = \tau^2(1-\tau) = 0.009 & & p_\tau(\mathtt{aab},\mathtt{ab}) = 2\tau(1-\tau)^2 = 0.162 \\ p_\tau(\mathtt{aab},\mathtt{a}) = 2\tau^2(1-\tau) = 0.018 & & p_\tau(\mathtt{aab},\mathtt{aab}) = (1-\tau)^3 = 0.729 \end{array}$$

Observe that $\sum_{u'} p_\tau(u,u') = 1$. The difference between, e.g., $p_\tau(u,\mathtt{a})$ and $p_\tau(u,\mathtt{b})$, comes from the fact that, starting from u, there are two distinct ways of getting $\mathtt{a}$ by losses, while there is only one way of getting $\mathtt{b}$. $\square$

5 Probabilistic lossy channel systems

A *probabilistic lossy channel system* (PLCS) is a tuple $S = \langle Q, C, \Sigma, \Delta, \sigma_0, D\rangle$ s.t. $\langle Q, C, \Sigma, \Delta, \sigma_0\rangle$ is a channel system, and $D : (\Delta \cup \{0\}) \mapsto (0,\infty)$ is a *weight function* of its rules.

Definition 5.1 (Markov chain semantics of PLCS's). *The Markov chain M^τ_S associated with a PLCS S as above, and a fault rate $\tau \in (0,1)$ is $M^\tau_S \stackrel{def}{=} \langle W, P, P_0\rangle$ where W is the set of configurations of S, $P_0(\sigma_0) \stackrel{def}{=} 1$, and where $P(\sigma,\sigma')$, the probability that S moves from σ to σ' in one step, is given by*

$$P(\sigma,\sigma') \stackrel{def}{=} \frac{\displaystyle\sum_{\delta\in En(\sigma)} D(\delta) \times p_\tau(Succ_\delta(\sigma),\sigma')}{\displaystyle\sum_{\delta\in En(\sigma)} D(\delta)}. \tag{6}$$

tedly, there are situations calling for yet other models: e.g. [ABPJ00] assumes losses only occur when a message enters the queue.

It is readily seen that M_S^τ is indeed a Markov chain. It is usually infinite and not bounded.

An important consequence of the local-fault model is that the more messages are in the queue, the more likely some losses will make the number of messages decrease. We formalize this by introducing a partition $W = W_0 + W_1 + \cdots + W_n + \cdots$ of the set of configurations of M_S^τ given by $W_n \stackrel{\text{def}}{=} \{\sigma \in W \mid |\sigma| = n\}$, with $|\langle r, U\rangle| \stackrel{\text{def}}{=} \sum_c |U(c)|$. Then for any S and τ we have

Lemma 5.2. *For all $e > 0$ there is a rank $I \in \mathbb{N}$ s.t. for all $i \geq I$ and $\sigma \in W_i$*

$$\sum\{P(\sigma, \sigma') \mid \sigma' \in W_0 \cup W_1 \cup \cdots \cup W_{i-1}\} > 1 - e. \tag{7}$$

Corollary 5.3. *In M_S^τ, W_0 is a finite attractor.*

Theorem 5.4. (Decidability of model checking for PLCS's) *The problem of checking whether M_S^τ almost-surely (resp. almost-never, resp. possibly) satisfies a Street property α is decidable.*

Proof. Since reachability is decidable for lossy channel systems, the graph $G_{M_S^\tau}(W_0)$ can be built effectively. Since W_0 is a finite attractor, the graph can be used to check whether $\mathbb{P}(M, \sigma_0 \models \alpha) > 0$ by using the criterion provided by Proposition 2.7 (again, using decidability of reachability). Thus it is decidable whether M_S^τ possibly satisfies α. Now, by Remark 2.4, this entails the decidability of checking whether α is satisfied almost surely (resp. almost never). $\square$

Remark 5.5. From this algorithm, we deduce that, for a PLCS S, whether $\mathbb{P}(M_S^\tau, \sigma_0^\tau \models \alpha) = 1$ does not depend on the exact value of the fault rate τ or the weight function D of S. $\square$

6 Lossy channel systems as reactive Markov chains

Seeing LCS's as Markov chains requires that we see the nondeterministic choice between enabled transitions as being made probabilistically (witness the D weight function in PLCS's).

However, it is more natural to see these choices as being made nondeterministically: this nondeterminism models the lack of any assumption regarding scheduling policies or relative speeds (in concurrent systems), or the lack of any information regarding values that have been abstracted away (in abstract models), or the latitude left for later implementation decisions (in early designs). In all these situations, it is not natural to assume the choices are probabilistic. Even if, for qualitative properties, the exact values in the weight function are not relevant (Remark 5.5), the probabilistic viewpoint enforces a very strong fairness hypothesis on the nondeterministic choices, something which is not suitable except perhaps for concurrent systems.

For these reasons, it is worthwhile to go beyond the Markov chain model and use reactive Markov chains. Below, a *nondeterministic probabilistic lossy channel system* (a NPLCS) is simply a LCS where losses are probabilistic so that the semantics is given under the form of a RMC instead of a transition system.

Definition 6.1. (Reactive Markov chain semantics of NPLCS's) *The RMC associated with a NPLCS S and a fault rate τ is $M^\tau_S = \langle W_+ \cup W_-, W_+, P, P_0\rangle$ where W_+ and W_- are two copies of the set $Q \times \Sigma^{*C}$. P_0 selects $\sigma_{0,+}$, the initial configuration, and P is given by*

$$P(\langle q, U\rangle_+, \langle q', U'\rangle_-) > 0 \textit{ iff } \langle q, U\rangle \rightarrow_{\text{perf}} \langle q', U'\rangle, \tag{8}$$

$$P(\langle q, U\rangle_-, \langle q', U'\rangle_+) = \begin{cases} p_\tau(U, U') & \textit{if } q = q', \\ 0 & \textit{otherwise.} \end{cases} \tag{9}$$

Thus positive configurations are nondeterministic and implement perfect steps of S, reaching negative configurations where message losses are probabilistic. Note that the precise value of $P(\sigma_+, \sigma'_-)$ in (8) is not relevant.

Since the probabilistic configurations are only used as some intermediate steps between nondeterministic configurations, it is tempting to omit mentioning them altogether when discussing the behavior of NPLCS's. Indeed, in the next sections, we rarely write configurations with the "$-$" or "$+$" subscript they require: unless explicitly stated, we always refer to the nondeterministic configurations.

7 Model checking NPLCS's

Model checking for NPLCS's is trickier than model checking PLCS's, and the existence of the finite attractor does not always allow reducing to a decidable finite problem. The decidability results we provide below rely on the finite attractor and downward-closure of reachability sets.

We considered the general case (checking for a Street property) as well as restricted cases where only properties of the form $\Diamond A$ (reachability), $\Box A$ (invariant), $\bigwedge_i \Box\Diamond A_i$ (conjunction of Büchi properties), and $\bigvee_i \Diamond\Box A_i$. Below we adopt the simplifying assumption that all sets A used in properties either are singletons or have the form $\Uparrow\{s_1, \ldots, s_k\}$ for a set of control states $s_1, \ldots, s_k$.

We exhibit some decidable cases and some undecidable ones. Most problems are studied under a cooperative "$\exists u? \ldots$" form because this is easier to reason about, but all results are summarized in the adversarial form in Fig. 2 below.

7.1 Some decidable problems

We start by consider properties of the simple form $\alpha = \Diamond A$. We say a set $X \subseteq Q$ is *safe for* α if $\langle x, \varepsilon\rangle \xrightarrow{*}_X A$ for all $x \in X$. Here the notation "$\sigma \xrightarrow{*}_X \sigma'$" denotes reachability under the constraint that only control states from X are used (the constraint applies to the endpoints σ and σ' as well). This coincides with reachability in the LCS $S_{|X}$ obtained from S by deleting control states from $Q \setminus X$, and is thus decidable.

Lemma 7.1. *There exists a scheduler u such that $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \Diamond A) = 1$ iff s belongs to a safe X.*

Corollary 7.2. *It is decidable whether there exists a scheduler u s.t. $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \Diamond A) = 1$.*

We now consider properties of the special form $\alpha = \bigwedge_{i=1}^{n} \Box\Diamond A_i$. We say $x \in Q$ is *allowed* if for all $i = 1, \ldots, n$, $\langle x, \varepsilon\rangle \xrightarrow{*} A_i$. Otherwise x is *forbidden.* It is decidable whether x is allowed or forbidden.

Lemma 7.3. *Assume all states in S are allowed. Then there exists a scheduler u s.t. $\mathbb{P}(M_S^{\tau,u}\langle s, \varepsilon\rangle \models \alpha) = 1$.*

Lemma 7.4. *Assume x is forbidden and define $S - x$ as the LCS where control state x has been removed. Then the following are equivalent:*
1. There exists a scheduler u s.t. $\mathbb{P}(M_S^{\tau,u}\langle s, \varepsilon\rangle \models \alpha) = 1$.
2. $x \neq s$ and there exists a scheduler u' s.t. $\mathbb{P}(M_{S-x}^{\tau,u'}\langle s, \varepsilon\rangle \models \alpha) = 1$.

Corollary 7.5. *It is decidable whether there exists a scheduler u s.t. $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \bigwedge_{i=1}^{n} \Box\Diamond A_i) = 1$.*

7.2 An undecidable problem

Theorem 7.6. *The problem of checking whether, given a NPLCS S and a Street property α, $\mathbb{P}(M_S^{\tau,u} \models \alpha) = 1$ for all schedulers u, is undecidable.*

The proof is by reduction from the boundedness problem for LCS's, shown undecidable in [May00].

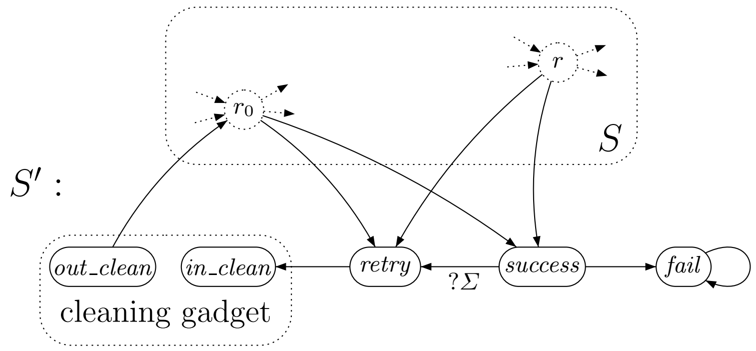

Fig. 1. The NPLCS S' associated with LCS S

Let $S = \langle Q, \{c\}, \Sigma, \Delta, \sigma_0\rangle$ be a single-channel LCS that does not use emptiness tests, and where $\sigma_0 = \langle r_0, \varepsilon\rangle$. We modify S to obtain S', a new LCS. Fig. 1, where S is in the dashed box, illustrates the construction: S' is obtained by adding three control states *retry*, *success* and *fail*, rules allowing to jump from any S-state $r \in Q$ to *retry* or *success*,[6] and some additional rules between the new states, as depicted in Fig. 1. The "$?\Sigma$" label is a shorthand for all $?a$ where $a \in \Sigma$. We further insert a cleaning gadget (not described) that allows to move from *retry* to r_0 (in a non-blocking way) but empties the channel in the process: this ensures we only jump back to S via its initial configuration $\langle r_0, \varepsilon\rangle$.

If we now see S' as a NPLCS with some fault rate τ, we have

Proposition 7.7. *S is unbounded iff $\mathbb{P}(M_{S'}^{\tau,u}, \sigma_0 \models \Box\Diamond{\uparrow}success \wedge \Box\Diamond{\uparrow}retry) > 0$ for some scheduler u.*

Since boundedness of LCS's is undecidable, we obtain Theorem 7.6 even for NPLCS's without emptiness tests.[7]

7.3 More decidability with finite-memory schedulers!

The scheduler we build in the proof of Proposition 7.7 is not finite-memory. By contrast, all the schedulers exhibited in the proofs in Section 7.1 are finite-memory, so that these decidable problems do not depend on whether we restrict schedulers to the finite-memory ones only.

This observation suggests investigating whether some of our undecidable problems remain undecidable when we restrict to finite-memory schedulers. It turns out this is indeed the case.

We consider a NPLCS S and a Büchi property $\alpha = \bigwedge_{i=1}^{n} \Box\Diamond A_i$. For finite-memory schedulers, cooperative possibly and cooperative almost-sure are related by the following fundamental lemma:

Lemma 7.8. *There exists a finite-memory scheduler u s.t. $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \alpha) > 0$ iff there is some $s' \in Q$ and a finite-memory scheduler u' s.t $\langle s, \varepsilon\rangle \xrightarrow{*} \langle s', \varepsilon\rangle$ and $\mathbb{P}(M_S^{\tau,u'}, \langle s', \varepsilon\rangle \models \alpha) = 1$.*

Combining with Corollary 7.5, and the decidability of reachability, we obtain:

Corollary 7.9. *It is decidable whether there exists a finite-memory scheduler u s.t. $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \bigwedge_{i=1}^{n} \Box\Diamond A_i) > 0$.*

Hence the impossibility stated in Theorem 7.6 can be circumvented with:

Theorem 7.10. *The problem of checking whether, given a NPLCS S and a Street property α, $\mathbb{P}(M_S^{\tau,u} \models \alpha) = 1$ for all finite-memory schedulers u, is decidable.*

---

[6] Via some internal rule where no reading or writing or test takes place. Since such rules are easily simulated by writing to a dummy channel, we simplified Def. 3.1 and omitted them.

[7] Observe that, because of the idling rule, formulae of the form $\Box\Diamond A$ where A is some ${\uparrow}\{s_1, \ldots, s_n\}$, lead to decidable adversarial problems! This is an artifact and undecidability reappears as soon as we consider slightly more general sets A, e.g. $A \stackrel{\text{def}}{=} {\uparrow}success \setminus \langle success, \varepsilon\rangle$.

8 Conclusions and perspectives

When verifying lossy channel systems, adopting a probabilistic view of losses it is a way of enforcing progress and ruling out some unrealistic behaviors (under probabilistic reasoning, it is extremely unlikely that all messages will be lost). Progress could be enforced with fairness assumptions, but assuming fair losses makes verification undecidable [AJ96a,MS02]. It seems this undecidability is an artifact of the standard rigid view asking whether no incorrect behavior exists, when we could be content with the weaker statement that incorrect behaviors are extremely unlikely.

We proposed a model for channel systems where at each step losses of messages occur with some fixed probability $\tau \in (0, 1)$, and where the nondeterministic nature of the channel systems model is preserved. This model is more realistic than earlier proposals since it uses the local-fault model for probabilistic losses, and since it does not require to view the rules of the system as probabilistic. (Picking a meaningful value for τ is not required since the qualitative properties we are interested in do not depend on that value.)

We advocate a specific approach to the verification of these systems: *check that properties hold almost surely under any finite-memory scheduling policy*. It seems this approach is feasible: these adversarial model checking questions can be reduced to the decidable reachability questions that are usually verified on channel systems.

Several questions remain unanswered, and they are good candidates for further work:
1. Fig. 2, summarizing our results on the decidability of adversarial verification *when there is no restriction to finite-memory schedulers*, should be completed. In the table, "D" and "U" stand for decidable and undecidable. Some decidability results are trivial and labeled with "d".
2. Allowing idling makes our decidability proofs much easier (Remark 3.2). We believe this is just a technical simplification that has no impact on decidability, but this should be formally demonstrated.
3. On theoretical grounds, it would be interesting to try to extend our work and consider RMC models of LCS's where the probabilistic states are not limited to message losses but could accommodate probabilistic choices between some rules.

References

[AAB99] P. A. Abdulla, A. Annichini, and A. Bouajjani. Symbolic verification of lossy channel systems: Application to the bounded retransmission protocol. In *Proc. 5th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'99)*, vol. 1579 of *Lect. Notes Comp. Sci.*, pages 208–222. Springer, 1999.

	$\mathbb{P}(\ldots) = 1$	$\mathbb{P}(\ldots) = 0$	$\mathbb{P}(\ldots) < 1$	$\mathbb{P}(\ldots) > 0$
$\Diamond A$	d	d	D	d
$\Box A$	d	d	d	D
$\bigwedge_i \Box\Diamond A_i$	?	U	D	?
$\bigvee_i \Diamond\Box A_i$	U	?	?	D
$\bigwedge_i(\Box\Diamond A_i \Rightarrow \Box\Diamond A'_i)$	U	U	?	?

Fig. 2. (Un)Decidability of adversarial model checking in the unrestricted case

[ABPJ00] P. A. Abdulla, C. Baier, S. Purushothaman Iyer, and B. Jonsson. Reasoning about probabilistic lossy channel systems. In *Proc. 11th Int. Conf. Concurrency Theory (CONCUR'2000)*, vol. 1877 of *Lect. Notes in Computer Sci.*, pages 320–333. Springer, 2000.

[AJ96a] P. A. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *Information and Computation*, 130(1):71–90, 1996.

[AJ96b] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, 1996.

[AR03] P. A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In *Proc. FOSSACS'2003 (this volume)*. Springer, 2003.

[BE99] C. Baier and B. Engelen. Establishing qualitative properties for probabilistic lossy channel systems: An algorithmic approach. In *Proc. 5th Int. AMAST Workshop Formal Methods for Real-Time and Probabilistic Systems (ARTS'99)*, vol. 1601 of *Lect. Notes Comp. Sci.*, pages 34–52. Springer, 1999.

[BKH99] C. Baier, J.-P. Katoen, and H. Hermanns. Approximate symbolic model checking of continuous-time Markov chains. In *Proc. 26th Int. Coll. Automata, Languages, and Programming (ICALP'99)*, vol. 1644 of *Lect. Notes Comp. Sci.*, pages 142–162. Springer, 1999.

[Bré99] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, vol. 31 of *Texts in Applied Mathematics*. Springer, 1999.

[BZ83] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323–342, 1983.

[CFP96] G. Cécé, A. Finkel, and S. Purushothaman Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, 1996.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[Der70] C. Derman. *Finite-State Markovian Decision Processes*. Academic Press, 1970.

[Fin94] A. Finkel. Decidability of the termination problem for completely specificied protocols. *Distributed Computing*, 7(3):129–135, 1994.

[HSP83] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent programs. *ACM Transactions on Programming Languages and Systems*, 5(3):356–380, 1983.

[May00] R. Mayr. Undecidable problems in unreliable computations. In *Proc. 4th Latin American Symposium on Theoretical Informatics (LATIN'2000)*, vol. 1776 of *Lect. Notes Comp. Sci.*, pages 377–386. Springer, 2000.

[MS02] B. Masson and Ph. Schnoebelen. On verifying fair lossy channel systems. In *Proc. 27th Int. Symp. Math. Found. Comp. Sci. (MFCS'2002)*, vol. 2420 of *Lect. Notes Comp. Sci.*, pages 543–555. Springer, 2002.

[Pan01] P. Panangaden. Measure and probability for concurrency theorists. *Theoretical Computer Sci.*, 253(2):287–309, 2001.

[PN97] S. Purushothaman Iyer and M. Narasimha. Probabilistic lossy channel systems. In *Proc. 7th Int. Joint Conf. Theory and Practice of Software Development (TAPSOFT'97)*, vol. 1214 of *Lect. Notes Comp. Sci.*, pages 667–681. Springer, 1997.

[Sch01] Ph. Schnoebelen. Bisimulation and other undecidable equivalences for lossy channel systems. In *Proc. 4th Int. Symp. Theoretical Aspects of Computer Software (TACS'2001)*, vol. 2215 of *Lect. Notes Comp. Sci.*, pages 385–399. Springer, 2001.

[Sch02] Ph. Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

[Var85] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. 26th IEEE Symp. Foundations of Computer Science (FOCS'85)*, pages 327–338, 1985.

[Var99] M. Y. Vardi. Probabilistic linear-time model checking: An overview of the automata-theoretic approach. In *Proc. 5th Int. AMAST Workshop Formal Methods for Real-Time and Probabilistic Systems (ARTS'99)*, vol. 1601 of *Lect. Notes Comp. Sci.*, pages 265–276. Springer, 1999.

This technical appendix contains the proofs omitted in the proceedings version.

A Proof of Proposition 2.7

We start by spelling out a classical lemma that will prove useful at several places.

Lemma A.1. *If $P(\sigma_{i-1},\sigma_i) > 0$ for $i = 1,\dots,n$, then $\mu_M(\Box\Diamond\sigma_0 \Rightarrow \Box\Diamond\sigma_n) = 1$.*

This holds for infinite Markov chains and states that if σ_n is reachable from σ_0, and σ_0 is visited infinitely often, then almost surely σ_n is visited infinitely often.

We now proceed with the Proof of Proposition 2.7:
($\Leftarrow$): Assume $\sigma \xrightarrow{*} B$ for a BSCC B such that for all $i = 1,\dots,n$ $B \xrightarrow{*} A_i$ implies $B \xrightarrow{*} A'_i$. Then, since B is finite, Lemma A.1 applies, and for any $\sigma' \in B$, $B \xrightarrow{*} A_i$ implies $\mathbb{P}(M,\sigma' \models \Box\Diamond A_i) = 1$. Thus $\mathbb{P}(M,\sigma \models \bigwedge_{i=1}^{n}(\Box\Diamond A_i \Rightarrow \Box\Diamond A'_i)) = \mathbb{P}(M,\sigma \models \Diamond B)$ which is positive since $\sigma \xrightarrow{*} B$.

($\Rightarrow$): Assume $\mathbb{P}(M,\sigma \models \bigwedge_{i=1}^{n}(\Box\Diamond A_i \Rightarrow \Box\Diamond A'_i)) > 0$. Then (3) implies that $\mu_\sigma(\{\pi \mid \pi \models \bigwedge_{i=1}^{n}(\Box\Diamond A_i \Rightarrow \Box\Diamond A'_i) \wedge \lim_{W_a}(\pi) = B\}) > 0$ for at least one BSCC B (entailing $\sigma \xrightarrow{*} B$). Now, for any $i = 1,\dots,n$, if $B \xrightarrow{*} A_i$ then $\Box\Diamond B$ almost surely implies $\Box\Diamond A_i$ (again by Lemma A.1), so that $\mu_\sigma(\{\pi \mid \pi \models \Box\Diamond A'_i \wedge \lim_{W_a}(\pi) = B\}) > 0$. Hence A'_i must be reachable from B.

B Proof of Corollary 5.3

Corollary 5.3 is a standard but tedious exercise (see e.g. [Bré99, § 2.5]). We give a detailed solution for the sake of completeness.

For any $\sigma \in W$, let $\mathbb{P}_\sigma \stackrel{\text{def}}{=} \mathbb{P}(M^\tau_S,\sigma \models \Diamond W_0)$. This is well-defined by Theorem 2.3. We further have

$$\mathbb{P}_\sigma = \begin{cases} \sum_{\sigma' \in W} P(\sigma,\sigma') \times \mathbb{P}_{\sigma'} & \text{if } \sigma \notin W_0, \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

Define $\mathbb{P}_n \stackrel{\text{def}}{=} \min\{\mathbb{P}_\sigma \mid \sigma \in W_n\}$ and $\mathbb{Q}_n \stackrel{\text{def}}{=} \min\{\mathbb{P}_i \mid i = 0,1,\dots,n\}$: the sequence $(\mathbb{Q}_n)_{n\in\mathbb{N}}$ is positive and decreasing.

For any $n > 0$ and $\sigma \in W_n$, equation (10) entails

$$\mathbb{P}_\sigma \geq \overbrace{\sum_{\sigma' \in W_{\leq n-1}} P(\sigma,\sigma')}^{p_\sigma} \times \mathbb{Q}_{n-1} + \overbrace{\sum_{\sigma' \in W_n} P(\sigma,\sigma')}^{q_\sigma} \times \mathbb{Q}_n + \overbrace{\sum_{\sigma' \in W_{n+1}} P(\sigma,\sigma')}^{r_\sigma} \times \mathbb{Q}_{n+1}.$$

From $\mathbb{P}_\sigma \geq p_\sigma \mathbb{Q}_{n-1} + q_\sigma \mathbb{Q}_n + r_\sigma \mathbb{Q}_{n+1}$, we derive $\mathbb{P}_n \geq p_n \mathbb{Q}_{n-1} + q_n \mathbb{Q}_n + r_n \mathbb{Q}_{n+1}$ by picking one of the σ's in W_n that make $\mathbb{P}_\sigma$ minimal and letting $p_n \stackrel{\text{def}}{=} p_\sigma$, etc. Using $\mathbb{Q}_{n-1} \geq \mathbb{Q}_n \geq \mathbb{Q}_{n+1}$ and $\mathbb{Q}_n = \min(\mathbb{P}_n, \mathbb{Q}_{n-1})$, we obtain $\mathbb{Q}_n \geq p_n \mathbb{Q}_{n-1} + q_n \mathbb{Q}_n + r_n \mathbb{Q}_{n+1}$ (since $p_n + q_n + r_n = 1$), and thus

$$r_n(\mathbb{Q}_n - \mathbb{Q}_{n+1}) \geq p_n(\mathbb{Q}_{n-1} - \mathbb{Q}_n) \tag{11}$$

Define now $\lambda_n \stackrel{\text{def}}{=} \mathbb{Q}_n - \mathbb{Q}_{n+1}$: since $(\mathbb{Q}_n)_{n\in\mathbb{N}}$ is decreasing but stays positive, $(\lambda_n)_{n\in\mathbb{N}}$ is positive with $\lim_{n\to\infty} \lambda_n = 0$. Equation (11) rewrites as $\lambda_{n-1} \leq \frac{r_n}{p_n}\lambda_n$, entailing, for any n and k

$$\lambda_n \leq \frac{r_{n+1}}{p_{n+1}} \frac{r_{n+2}}{p_{n+2}} \cdots \frac{r_{n+k}}{p_{n+k}} \lambda_{n+k} \quad (\stackrel{\text{def}}{=} \gamma_{n,k}\, \lambda_{n+k}). \tag{12}$$

Let us now consider a fixed n. We already know $\lim_{k\to\infty} \lambda_{n+k} = 0$. But we further have $\lim_{k\to\infty} \gamma_{n,k} = 0$: Lemma 5.2 says that $\lim_{k\to\infty} p_{n+k} = 1$ and we know $0 \leq p_{n+k} + r_{n+k} \leq 1$. Thus $\lambda_n = 0$.

This holds for all n so that $\mathbb{Q}_n = 1$ for all n, and hence $\mathbb{P}_\sigma = 1$ for all σ. In other words, starting from any configuration, W_0 is eventually visited almost surely. Finally, for any $k \in \mathbb{N}$ and starting from any $\sigma \in W$, W_0 will be visited k times almost surely. This entails

$$\mathbb{P}(M_S^\tau, \sigma \models \Box\Diamond W_0) = 1. \tag{13}$$

C Proofs for Section 7.1

C.1 Proof of Lemma 7.1

($\Leftarrow$): Assume $s \in X$ for some safe X. We build a scheduler u that provides almost-sure satisfaction of $\Diamond A$ from any $\langle x, \varepsilon\rangle$ with $x \in X$. For this purpose, u simply picks rules with the aim of performing $\langle x, \varepsilon\rangle \xrightarrow{*}_X A$. As long as the probabilistic losses occur as requested by $\langle x, \varepsilon\rangle \xrightarrow{*}_X A$, u goes on. When a probabilistic loss is not as expected, u stops, idles until the channels are empty (which is bound to eventually happen), and then aims again for A from the $\langle y, \varepsilon\rangle$ that is then current. Since u never stepped out of X, we have $y \in X$. With these repeated retries, A is reached almost surely (again by Lemma A.1, using the fact that we only apply it to a finite number of configurations).

($\Rightarrow$): Assume there is a scheduler u ensuring $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \Diamond A) = 1$. We say a configuration σ is *required by u* if $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models (\neg A)\, \mathit{Until}\, \sigma) > 0$, i.e. it is possible that $M_S^{\tau,u}$ visits σ without having crossed A. If $\sigma \neq \langle s, \varepsilon\rangle$ is required, there must be a path $\langle s, \varepsilon\rangle \to \ldots \to \sigma$ in $M_S^{\tau,u}$ where A is not visited, so that any $\sigma' \sqsubseteq \sigma$ is required too. Thus the set of required configurations is downward-closed and contains $\langle s, \varepsilon\rangle$. Furthermore, we have $\sigma \xrightarrow{*} A$ for any required σ, otherwise u could not ensure $\Diamond A$ almost surely. Let now X be the set of control states that appear in the required configurations: X is safe and contains s.

C.2 Proof of Lemma 7.3

We build u like in the proof of the ($\Leftarrow$) direction of Lemma 7.1. Here u aims for the A_is in turn, retries as long as the current A_i has not been reached, then moves to the next target A_{i+1}, and starts the whole thing again once all targets have been reached. Since all states are allowed, u is never blocked. $\square$

C.3 Proof of Lemma 7.4

(2. $\Rightarrow$ 1.): using u' on S will work.

(1. $\Rightarrow$ 2.): assume $\mathbb{P}(M_S^{\tau,u}\langle s,\varepsilon\rangle \models \alpha) = 1$. Then $s \xrightarrow{*} A_i$ for all i, hence $s \neq x$. Observe that u never actually picks a rule that moves to x (or else there is a non-zero probability to end up in $\langle x,\varepsilon\rangle$ and hence to fail fulfilling α). Thus u can be used on $S - x$ and provides $\mathbb{P}(M_{S-x}^{\tau,u}\langle s,\varepsilon\rangle \models \alpha) = 1$.

C.4 Proof of Corollary 7.5

A possible algorithm is to apply Lemma 7.4 repeatedly, removing the forbidden states one by one (this may create new forbidden states in the pruned system). If at any moment s is a forbidden state, then we answer negatively. Otherwise we end up with some smaller system where all states are allowed and Lemma 7.3 concludes positively.

D Proof of Proposition 7.7

($\Rightarrow$): assume S is unbounded. Then there exists configurations reachable from σ_0 and with arbitrarily large channel contents.

Let $(p_n)_{n=1,2,\ldots}$ be a sequence of numbers in $(0,1)$ s.t. $\prod_{n=1}^{\infty} p_i > 0$, e.g. $p_n \stackrel{\text{def}}{=} \frac{n^2}{n^2+1}$, and pick a sequence $(\langle r_n, w_n\rangle)_{n=1,2,\ldots}$ of reachable configurations s.t. $p_\tau(w_n,\varepsilon) \leq 1 - p_n$: this can be obtained by choosing w_n large enough.

Based on the $\langle r_n, w_n\rangle$'s, we build a scheduler u for $M_{S'}^{\tau}$. u works in phases numbered $1, 2, \ldots$ When phase n starts, $M_{S'}^{\tau}$ is in configuration σ_0 and u attempts to reach $\langle r_n, w_n\rangle$. In principle, this can be achieved (since $\langle r_n, w_n\rangle$ is reachable) but it requires that the right messages are lost at the right time. These losses are probabilistic and u cannot control them. Thus u aims for $\langle r_n, w_n\rangle$ and hopes that the right losses will occur. u goes on according to plan as long as losses occur as hoped.

When a “wrong” loss occurs, u resigns temporarily, jumps directly to *retry*, then reaches σ_0 via the cleaning gadget, and then it can have another go at trying to reach $\langle r_n, w_n\rangle$.

When $\langle r_n, w_n\rangle$ is eventually reached (which will happen almost surely), u jumps to *success*, from there to *retry*, and starts phase $n+1$. With these successive phases, u tries to visit *success* an infinite number of times.

It may happen that, when jumping from $\langle r_n, w_n\rangle$ to *success*, all messages in the channel are lost in one fell swoop, leaving us in $\langle \mathit{success}, \varepsilon\rangle$. In that case u is not able to initiate phase $n+1$ (at least one message must be consumed to move from *success* to *retry*) and will fail satisfying $\Box\Diamond A$. However, the probability of losing all messages in phase n is $p_\tau(w_n, \varepsilon) \leq 1 - p_n$, so that when phase n is initiated, it will lead to phase $n+1$ with probability $\geq p_n$. Finally, the probability that u ends up failing is $\prod_{n=1}^{\infty} p_\tau(w_n, \varepsilon) \geq \prod_{n=1}^{\infty} p_n > 0$.

Hence u has $\mathbb{P}(M_{S'}^{\tau;u}, \sigma_0 \models \Box\Diamond A) > 0$.

($\Leftarrow$): we prove that if S is bounded then $\mathbb{P}(M_{S'}^{\tau;u}, \sigma_0 \models \Box\Diamond A) = 0$ for all schedulers u.

Assume then that S is bounded and u is some scheduler for $M_{S'}^{\tau}$. First observe that, since S is bounded, S' is bounded too (except perhaps within the cleaning gadget).

We consider the set of runs $\{\pi \mid \pi \models \Box\Diamond A\}$ in the Markov chain $M_{S'}^{\tau;u}$. Since S' is bounded, a run visiting *success* infinitely often must visit some $\langle \mathit{success}, w\rangle_-$ infinitely often. But the probabilistic transition $\langle \mathit{success}, w\rangle_- \to \langle \mathit{success}, \varepsilon\rangle_+$ has strictly positive probability so that $\mathbb{P}(M_{S'}^{\tau;u}, \sigma_0 \models \Box\Diamond\langle \mathit{success}, w\rangle_- \Rightarrow \Box\Diamond\langle \mathit{success}, \varepsilon\rangle_+) = 1$ (Lemma A.1). But from $\langle \mathit{success}, \varepsilon\rangle_+$ we cannot go back to A. Thus, for any $w \neq \varepsilon$, $\mathbb{P}(M_{S'}^{\tau;u}, \sigma_0 \models \Box\Diamond\langle \mathit{success}, w\rangle_-) = 0$. Finally, $\mathbb{P}(M_{S'}^{\tau;u}, \sigma_0 \models \Box\Diamond A) = 0$.

E Proof of Lemma 7.8

For simplification purposes, we give our proofs for memoryless schedulers but they extend to finite-memory schedulers with some additional work (mostly of the tedious bookkeeping kind),

Assume u is a finite-memory scheduler and $x \in Q$ is some control state.

Lemma E.1. *If* $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \Box\Diamond{\uparrow}x) > 0$ *then* $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \Box\Diamond\langle x, \varepsilon\rangle) > 0$.

Proof. (We gives the proof for a memoryless u.) Since W_0 is an attractor, we get $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \Box\Diamond{\uparrow}x \ \wedge\ \Box\Diamond W_0) > 0$. Since W_0 is finite, there must be some $\langle y, \varepsilon\rangle \in W_0$ s.t. $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \Box\Diamond{\uparrow}x \ \wedge\ \Box\Diamond\langle y, \varepsilon\rangle) > 0$. Now consider a path π where $\langle y, \varepsilon\rangle$ and ${\uparrow}x$ are visited infinitely often and pick a subsequence of π where one goes from $\langle y, \varepsilon\rangle$ to some $\langle x, w\rangle$. It has the form:

$$\begin{aligned}\langle y, \varepsilon\rangle_+ = \langle s_0, V_0'\rangle_+ &\xrightarrow{\delta_1} \langle s_1, V_1\rangle_- \xrightarrow{p_1} \langle s_1, V_1'\rangle_+ \xrightarrow{\delta_2} \cdots \\ \cdots &\xrightarrow{\delta_n} \langle s_n, V_n\rangle_- \xrightarrow{p_n} \langle s_n, V_n'\rangle_+ = \langle x, w\rangle_+,\end{aligned}$$

where the δ_i's are the rules picked by u, and the p_i's are the probabilities that message losses transform V_i in V_i'.

Since u is memoryless, whenever $\langle y, \varepsilon\rangle_+$ or any other $\langle s_i, V_i'\rangle_+$ is visited, u picks the same δ_{i+1} rule. Thus if $\langle s_i, V_i'\rangle_+$ is visited infinitely often, then

$\langle s_{i+1}, V_{i+1}\rangle_-$ too is visited infinitely often. Furthermore, since there is a positive probability that $\langle s_i, V_i\rangle_-$ is immediately followed by $\langle s_i, V_i'\rangle_+$, if $\langle s_i, V_i\rangle_-$ is visited infinitely often, then almost surely $\langle s_i, V_i'\rangle_+$ is visited infinitely often.

Therefore, since there is a positive probability that u makes S visit $\langle y, \varepsilon\rangle_+$ infinitely often, there is the same positive probability that it makes S visit $\langle s_n, V_n\rangle_-$, i.e. $\langle x, V_n\rangle_-$, infinitely often. Almost surely infinitely many of these $\langle x, V_n\rangle_-$ will be immediately followed by $\langle x, \varepsilon\rangle_+$. $\square$

Remark E.2. Lemma E.1 does not hold for schedulers that are not finite-memory. This can be seen by looking at the scheduler we built in the Proof of the ($\Rightarrow$) direction of Proposition 7.7. If S is unbounded, there is a strictly positive probability that *success* is visited infinitely often, but of course it is impossible to visit *success*, ε infinitely often. $\square$

Lemma E.3. *If* $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \bigwedge_{i=1}^n \Box\Diamond A_i) > 0$ *then there exist control states* $x_1, \ldots, x_n$ *s.t.* $\langle x_i, \varepsilon\rangle \in A_i$ *for all* i *and* $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \bigwedge_{i=1}^n \Box\Diamond\langle x_i, \varepsilon\rangle) > 0$.

Proof (Idea). As for Lemma E.1. $\square$

Lemma E.4. *If* $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \Box\Diamond\langle x, \varepsilon\rangle) > 0$ *then* $\mathbb{P}(M_S^{\tau,u}, \langle x, \varepsilon\rangle \models \Box\Diamond\langle x, \varepsilon\rangle) = 1$.

Proof. (We gives the proof for a memoryless u.) For $k = 0, 1, \ldots$, let Π_k be the set of paths where $\langle x, \varepsilon\rangle$ is visited at least k times. Let $\mathbb{P}_k^u(\sigma)$ be the value of $\mu_\sigma(\Pi_k)$ in $M_S^{\tau,u}$: clearly $\mathbb{P}_k(\sigma) \geq \mathbb{P}_{k+1}(\sigma) \geq 0$. Let $\mathbb{P}_\omega(\sigma) \stackrel{\text{def}}{=} \lim_{k\to\infty} \mathbb{P}_k(\sigma)$.

Since u is memoryless, reaching $\langle x, \varepsilon\rangle$ from σ and then reaching a second time $\langle x, \varepsilon\rangle$ from $\langle x, \varepsilon\rangle$ are independent events. Hence $\mathbb{P}_{k+1}(\sigma) = \mathbb{P}_k(\sigma) \times \mathbb{P}_2(\langle x, \varepsilon\rangle)$. Here $\mathbb{P}_2(\langle x, \varepsilon\rangle)$ is the probability that u, started from $\langle x, \varepsilon\rangle$, makes S visit $\langle x, \varepsilon\rangle$ a second time. Finally we get

$$\mathbb{P}_{k+1}(\sigma) = \mathbb{P}_1(\sigma) \times \mathbb{P}_2(\langle x, \varepsilon\rangle)^k. \tag{14}$$

The hypothesis that $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \Box\Diamond\langle x, \varepsilon\rangle) > 0$ means that $\mathbb{P}_\omega(\sigma) > 0$. But, in view of (14), $\mathbb{P}_\omega(\sigma) > 0$ implies $\mathbb{P}_2(\langle x, \varepsilon\rangle) = 1$. Hence $\mathbb{P}_\omega(\langle x, \varepsilon\rangle) = 1$. $\square$

Corollary E.5. *If* $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \bigwedge_{i=1}^n \Box\Diamond\langle x_i, \varepsilon\rangle) > 0$ *then for any* $j = 1, \ldots, n$, $\mathbb{P}(M_S^{\tau,u}, \langle x_j, \varepsilon\rangle \models \bigwedge_{i=1}^n \Box\Diamond\langle x_i, \varepsilon\rangle) = 1$.

We now prove Lemma 7.8:

($\Leftarrow$): This direction is easy: let u have one try at reaching $\langle s', \varepsilon\rangle$ from $\langle s, \varepsilon\rangle$ (which succeeds with some positive probability) and then behave as u' when/if $\langle s', \varepsilon\rangle$ is reached.

($\Rightarrow$): Assume u is a finite-memory scheduler and $\mathbb{P}(M_S^{\tau,u}, \langle s, \varepsilon\rangle \models \alpha) > 0$. By Lemma E.3 there are some $(\langle x_i, \varepsilon\rangle)_{i=1,\ldots,n}$ reachable from $\langle s, \varepsilon\rangle$ s.t. $\mathbb{P}(M_S^{\tau,u}, \sigma_0 \models \bigwedge_{i=1}^n \Box\Diamond\langle x_i, \varepsilon\rangle) > 0$. But then for any i, $\mathbb{P}(M_S^{\tau,u}, \langle x_i, \varepsilon\rangle \models \alpha) = 1$ by Corollary E.5. Hence any x_i can be chosen for s'.