Model Checking Probabilistic Lossy Channel Systems *

Purush Iyer    Murali Narasimha
Dept of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
Ph: +1 919 515 7291, +1 919 515 6014
Fax: +1 919 515 7925
EMail: {purush, mnarasi}@eos.ncsu.edu

NCSU-CS TECH REPORT
TR-98-08
June 8, 1998

Abstract

Lossy channel systems model a set of finite state processes interacting with each other over unbounded, lossy FIFO channels. This computational model is an abstraction of protocols in the lower layers of the network protocol hierarchy. In spite of its unbounded FIFO queues the Lossy channel system model is not turing-powerful. It has been shown that the reachability problem is decidable [1]. However, the model-checking problem, against specifications in linear time temporal logic (LTL), is known to be undecidable [2]. Given that the rate of message loss in communication systems can be probabilistically characterized we consider a probabilistic version of Lossy channel systems. We show that the problem of checking whether a LTL requirement holds almost always, i.e., with probability 1, of probabilistic lossy channel systems (PLCS) is decidable. As can be expected the probability of message loss does not play a part in the model-checking procedure.

1 Introduction

Finite state machines which communicate over unbounded channels have been used as an abstract model of computation for reasoning about communication protocols [8, 15] and form the backbone of ISO protocol specification languages Estelle [21] and SDL [10]. Ever since the publication of the Alternating bit protocol [7] (the first ever computer communication protocol) it has been customary to assume, while modeling a protocol, that the communication channels between processes are free of errors. Possible errors in the communication channels are treated separately, or are completely ignored.

Over the past few years there have been attempts to rectify this situation to allow modeling of imperfections in the communication medium. In what might be termed as a surprising result it was shown [1] that the reachability problem is decidable for a model of processes communicating over unbounded FIFO buffers in which message losses are implicitly model-led. By capturing the notion of message loss the Lossy channel system model is closer to reality than the CFSM model. However, it turns out that the model-checking problem \(^1\) is undecidable [2]. As model-checking, which permits

---

*Supported in part by NSF under grant CCR 9404619

\(^1\)Given a lossy channel system \(L\) and a linear temporal logic formula \(\phi\), check whether the execution sequences of \(L\) satisfy \(\phi\).
checking of arbitrary safety and liveness properties of systems (expressible in a temporal logic), is now used in most software tools for concurrency [12, 20] the undecidability result would make it difficult to build design tools around the lossy channel system model.

In this paper we consider a probabilistic version of Lossy channel systems, where the message loss is characterized probabilistically. For such systems we show that probabilistic model-checking problem is decidable. Technically, we show that there is an algorithm which when given \( L_{Pr} \) (PLCS) and \( \phi \) (a LTL formula) can report whether the set of execution sequences of \( L_{Pr} \) that satisfy \( \phi \) has a probability 1, i.e., whether \( \phi \) holds of \( L_{Pr} \) almost always. Given that the cell loss rates (i.e., message loss rates) of communication channels in a network are readily available from vendors, the probabilistic lossy channel system model and the attendant probabilistic model-checking algorithm could be very useful in practise for designing, and validating, protocols that have to deal with communication errors.

**Related work:** We will now draw the reader's attention to recent related work on infinite state systems, and to the literature on probabilistic model-checking. Lossy channel systems can be thought of as an instance of the infinite state systems. With the success of formal methods based techniques for finite state systems there has been a spate of attention lately on reasoning about infinite state systems [3, 9]. However, decidable subclasses of processes communicating over unbounded FIFO queues have been investigated for a long time [14, 11, 15, 25]. To our knowledge, there are very few results on reasoning about probabilistic infinite state systems [18, 22]. Hart and Sharir consider the satisfiability problem for two probabilistic logics, and show that one of them has a finite-model property. In an earlier paper [22] we considered the following problem:

\[
\text{given a probabilistic lossy channel system } L_{Pr}, \text{ a LTL formula } \phi, \text{ a probability } p \text{ in the open interval } (0, 1), \text{ and a tolerance } \epsilon > 0, \text{ is the probability with which } \phi \text{ holds of } L_{Pr} \text{ at least } p \text{ (within the tolerance of } \epsilon).\]

Our solution in [22] involved an elaborate numerical-approximation scheme to answer that question. In contrast, the algorithm for the problem considered in this paper uses graph-theoretic properties of the infinite graph (i.e., infinite Labeled Markov chain) capturing the semantics of PLCS specifications.

There have been several efforts recently on model-checking probabilistic systems [4, 6, 13, 16, 17, 27]. However, all of them deal with finite state systems. The algorithm engendered by our decidability result uses automata-theoretic techniques that are similar to those used by Vardi [27] and by Courcoubetis and Yannakakis [13].

**Outline** The paper is organized as follows: in Section 2 we introduce the computation model PLCS, its semantics (Labeled Markov chains), and the definition of Muller automata (automata-theoretic version of LTL). In Section 3 we prove certain essential properties of the Markov chains which form the semantics of our specifications, and in Sections 4 and 5 we provide the proof of decidability of probabilistic model-checking of PLCS specifications, against LTL formulae, and our algorithm. We conclude, in Section 6, with a discussion on the implications of our result and explain why probabilistic model-checking is decidable but non-probabilistic model-checking is not.

## 2 Computational model and Definitions

Let \( W(Ch, \Sigma) \) be the set of \( Ch \) indexed vectors over the set of strings \( \Sigma^* \). We will write \( w(\epsilon) \) to denote the string at index \( \epsilon \). Let \( w[\epsilon := y] \) denote a vector of strings identical to \( w \) except at \( \epsilon \), where it is \( y \). A vector of empty strings and an empty string will both be written as \( \varepsilon \); which of the
two is intended should be clear from the context. Let \( \text{Comm}(Ch, \Sigma) = \{ e?\sigma, e!\sigma | e \in Ch, \sigma \in \Sigma \} \) be the set of receive and send commands over the channel set \( Ch \) and the set of message types \( \Sigma \).

**Definition 2.1** A lossy channel system is a structure of the form \( L = (S, Ch, \Sigma, Act, \rightarrow, s_0) \) where

- \( S \) is a finite set of control states,
- \( Ch \) is a finite set of channels,
- \( \Sigma \) is a finite set of message types,
- \( Act \) is a finite set of actions which includes \( \tau \) as the special silent action.
- \( \rightarrow \subseteq S \times \text{Comm}(Ch, \Sigma) \times Act \times S \) is the transition relation. An instance of the transition relation would be written as \( s \overset{c?\sigma, a}{\rightarrow} s' \).
- \( s_0 \in S \) is the start state.

**Definition 2.2** Given a Lossy channel system \( L = (S, Ch, \Sigma, Act, \rightarrow, s_0) \) define the labeled transition system of \( L \) as

\[
\mathcal{L}(L) = (?, \rightarrow, (s_0, \varepsilon))
\]

where

- \( ? \subseteq S \times W(Ch, \Sigma) \) is a (possibly infinite) set of states of the labeled transition system,
- \( (s_0, \varepsilon) \in ? \) is the start state of the system, and
- \( \rightarrow \subseteq ? \times Act \times ? \) is the transition relation.

\( ? \) and \( \rightarrow \) are inductively defined (as the least sets) satisfying the following rules.

- if \( \gamma = (s, w[c := \sigma x]) \in ? \) and \( s \overset{c?\sigma, a}{\rightarrow} s' \) then \( (s', w[c := x]) \in ? \) and \( (s, w[c := \sigma x]) \overset{a}{\rightarrow} (s', w[c := x]) \). This instance of \( \overset{a}{\rightarrow} \) captures the act of receiving a message.
- if \( \gamma = (s, w) \in ? \) and \( s \overset{c!\sigma}{\rightarrow} s' \) then \( (s, w[c := w(c)\sigma]) \in ? \) and \( (s, w) \overset{\sigma}{\rightarrow} (s', w[c := w(c)\sigma]) \). This instance captures the act of sending a message.
- if \( (s, w[c := \sigma y]) \in ? \) then \( (s, w[c := xy]) \in ? \) and \( (s, w[c := \sigma y]) \overset{\tau}{\rightarrow} (s, w[c := xy]) \). This instance of the transition relation denotes the silent action of losing a message.

Note that the system does not change state during a message loss.

Given a Lossy channel system \( L \), an execution sequence of \( L \) is a path through the labeled transition system of \( L \) starting from \( (s_0, \varepsilon) \); it is written in the form \( \tau = (s_0, \varepsilon) \overset{a_0}{\rightarrow} (s_1, w_1) \overset{a_1}{\rightarrow} \ldots \) ...

The trace of such an execution sequence is the sequence of non-\( \tau \) actions in an execution sequence, and is defined by the homomorphism induced by the map \( \mu(\alpha) = \begin{cases} \varepsilon & \alpha = \tau \\ \alpha & \text{otherwise} \end{cases} \).

Since message losses do not happen arbitrarily, and can generally be quantified probabilistically, we now introduce the notion of a Probabilistic lossy channel system.

**Definition 2.3** A Probabilistic lossy channel system \( L^Pr = (L; Pr; \varphi) \) where \( L \) is a lossy channel system, \( \varphi \in (0, 1) \) captures the probability of message loss in any state, and \( Pr : (\rightarrow) \rightarrow (0, 1) \) assigns a (relative) probability to the transitions of the system. We will write \( s \overset{c?\sigma, a}{\rightarrow}_p s' \) instead of \( Pr(s \overset{c?\sigma, a}{\rightarrow} s') = p \).
Consider a state \( \langle s, w \rangle \). We will say that a transition \( s \xrightarrow{c} s' \) is enabled in state \( \langle s, w \rangle \) provided \( w = w'[c := \sigma x] \) for some \( x \) and \( w' \). An output transition (of the form \( s \xrightarrow{c} s' \)) is always enabled in \( \langle s, w \rangle \) for all \( w \). Define \( R \), the total relative probability, of a state as follows:

\[
R(\langle s, w \rangle) = \sum \{ p | s \xrightarrow{c} s' is enabled in \langle s, w \rangle \} + \sum \{ p | s \xrightarrow{c} s' is enabled in \langle s, w \rangle \}
\]

Given a probabilistic lossy channel system \( I_{L, P} \) its semantics, a labeled Markov chain, based on \( L(L) \), is defined as follows:

**Definition 2.4** Let \( I_{L, P} \) be a Probabilistic lossy channel system, and let \( L(L) = (\langle ?, \rightarrow, \{ s_0, \varepsilon \} \rangle) \). The probability distribution function is defined as follows (note: it distributes \( 1 - \varphi \) among non-lossy transitions enabled at a state and \( \varphi \) among the lossy transitions):

\[
P(\langle s, w \rangle \xrightarrow{\alpha} \langle s', w' \rangle) = \begin{cases} 
\frac{\varepsilon}{w} + \frac{(1-\varepsilon)w}{R(s, w)} & \text{if } s \xrightarrow{c} s', w = w'[c := \sigma w'(c)], \alpha = \tau, s = s' \\
\frac{(1-\varepsilon)w}{R(s, w)} & \text{if } s \xrightarrow{c} s', w = w'[c := \sigma w'(c)], (\alpha \neq \tau, or s \neq s') \\
\frac{(1-\varepsilon)w}{R(s, w)} & \text{if } s \xrightarrow{c} s', w' = w[c := w'(c)\sigma] \\
\frac{w}{R(s, w)} & \text{otherwise}
\end{cases}
\]

We will write \( \langle s, w \rangle \xrightarrow{\alpha} p \langle s', w' \rangle \) instead of \( P(\langle s, w \rangle \xrightarrow{\alpha} \langle s', w' \rangle) = p \).

The probabilities arising in Markov chains are explained with reference to a measure space, defined as follows:

**Definition 2.5** ([23, 27]) Given a labeled Markov chain \( M(I_{L, P}) = (L(L); \mu) \) define a measure space \( S(M(I_{L, P})) = (\Omega, F, \mu) \), for assigning probabilities, where

- \( \Omega \) is the set of all infinite execution sequences of \( L \) starting at \( \gamma_0 \),
- \( F \) is a Borel field generated from the basic cylindric sets
  \[
  F(\gamma_0 \xrightarrow{a_1} \gamma_1 \xrightarrow{a_1} \cdots \gamma_n) = \{ \pi \in \Omega | \pi = \gamma_0 \xrightarrow{a_1} \gamma_1 \xrightarrow{a_2} \cdots \gamma_n \}
  \]
- \( \mu \) is a probability function defined by
  \[
  \mu(F(\gamma_0 \xrightarrow{a_1} \gamma_1 \cdots \xrightarrow{a_n} \gamma_n)) = p_1 \times p_2 \times \cdots p_n
  \]
  where \( \gamma_0 \xrightarrow{a_1} p_1 \gamma_1 \xrightarrow{a_2} p_2 \cdots \gamma_n \).

We will implicitly assume that any finite execution sequence terminating in a dead state is extended to an infinite execution sequence by adding a sink state, as it customarily done in Markov chain literature.

**Automaton characterization of Linear Temporal Logic** Given that we propose to use automata-theoretic techniques for model-checking we will make use of the following known facts:

- The set of sequences (or models) that satisfy a Linear temporal logic formula are describable by non-deterministic Büchi automaton, a certain kind of finite automata on infinite words [5, 26], and
• Non-deterministic Büchi automata are equivalent in expressive power to Deterministic Muller automata and there exist effective translations between the two classes of automata [26]. Furthermore, there is an effective translation from LTL to deterministic Mueller automaton.

Consequently, in the rest of this paper we will concentrate on (Deterministic) Muller automata.

Formally, we have:

**Definition 2.6** A Muller automaton is a tuple \( B = (T, q_0, F) \) where:

- \( T = (\Sigma, Q, \Delta) \) is a table where \((q \in) Q\) is a set of states, \( \Sigma \) is the alphabet, which does not contain \( \tau \), and \( \Delta : Q \times \Sigma \rightarrow Q \) is the transition function,

- \( F \subseteq 2^Q \) is a set of sets of accepting states.

Given an infinite word \( \eta = \sigma_0 \sigma_1 \ldots \in \Sigma^\omega \) a run of the automata \( B \) on \( \eta \) is the infinite sequence of states \( r = q_0 q_1 \ldots \) such that \( q_{i+1} = \Delta(q_i, \sigma_i) \) for \( i \geq 0 \). Let \( \inf(r) \) be the set of states that appear infinitely often in \( r \). \( B \) accepts an infinite word \( \eta \) iff the run \( r \) of \( B \) on \( \eta \) satisfies the constraints that \( \inf(r) \subseteq F \). Let \( \text{Lang}(B) \) be the set of all infinite words accepted by an automaton \( B \).

We will consider a Muller automaton \( B \) to be complete provided its transition function \( \Delta \) is total - i.e., at every state of the automaton there is transition for each \( \sigma \in \Sigma \). Note that for every Muller automaton \( B \) there is a complete Muller automaton \( \hat{B} \) such that the languages they accept are the same. \( \hat{B} \) can be constructed from \( B \) by adding a dead state \(-\) and defining \( \Delta_B(q, \sigma) = -\) provided \( \Delta_B(q, \sigma) \) is not defined, and \( \Delta_B(-, \sigma) = -\). In the rest of the paper when we specify \( B \) we implicitly use \( \hat{B} \).

**Satisfaction and its probability** Consider a lossy channel system \( L \). An execution sequence \( \pi \) in \( \mathcal{L}(L) \) satisfies a Muller automaton \( B \) provided \( \text{tr}(\pi) \in \text{Lang}(B) \). This definition applies equally well to probabilistic lossy channel systems. Consequently, given a Probabilistic lossy channel system \( L_{\psi}^{Pr} \) and a Muller automaton \( B \), define the probability of satisfaction as

\[
\mu(L_{\psi}^{Pr}, B) = \mu\{\pi \mid \pi \text{ is an execution sequence in } \mathcal{M}(L_{\psi}^{Pr}) \text{ and } \text{tr}(\pi) \in \text{Lang}(B)\}\).

The problem we shall solve is, therefore, the following:

Given a probabilistic channel system \( L_{\psi}^{Pr} \) and a Muller automaton \( B \) is \( \mu(L_{\psi}^{Pr}, B) = 1? \)

### 3 Properties of labeled transition systems

We will now introduce some basic facts about the states that appear in the labeled transition system of a lossy channel system and prove properties about execution paths in the labeled transition system.

Let \( \preceq \subseteq \Sigma^* \times \Sigma^* \) be the partial order defined as \( x \preceq y \) iff \( x = \sigma_0 \ldots \sigma_n \) and \( y = y_0 \sigma_0 \ldots y_n \sigma_n y_{n+1} \), for some \( y_i \in \Sigma^* \), \( 0 \leq i \leq n + 1 \). The ordering \( \preceq \) can be lifted easily to vectors of strings \( W(C h, \Sigma) \) and to states \( S \times W(C h, \Sigma) \) as (a) \( w_1 \preceq w_2 \) iff \( \forall c \in C h : w_1(c) \preceq w_2(c) \) and (b) \( \langle s, w \rangle \preceq \langle s', w' \rangle \) iff \( s = s' \) and \( w_1 \preceq w_2 \).

The ordering \( \preceq \subseteq S \times W(C h, \Sigma) \) satisfies an important property generally referred to as Higman’s property, stated as follows:

**Theorem 3.1 (Higman [19])** In every infinite sequence \( \gamma_0, \gamma_1 \ldots \) of states there is an infinite subsequence \( \gamma_{i_0}, \gamma_{i_1}, \ldots \) such that \( \forall j : \gamma_{i_j} \preceq \gamma_{i_{j+1}} \).
Returning now to the states of the labeled transition system an important observation is that the set of states of any lossy channel system are downward closed, i.e.,

$$\text{if } \gamma \in ? \text{ then } \forall \gamma' \leq \gamma : \gamma' \in ?$$

Note that this captures the notion that a message can always be lost from any state.

Given a lossy channel system $L$, its labeled transition system $L(L)$ can be looked upon as a directed (labeled) graph. A strongly connected component (scc) $C$ of $L(L)$ is a maximal subset of nodes (i.e., $C \subseteq ?$) such that all nodes in $C$ are reachable from each other. We will consider only those sccs which are reachable from the start state. We will call an scc $C$ a closed strongly connected component (cscc) provided there are no paths from a state in $C$ to a state in $? - C$.

The following lemma is a consequence of the above theorem.

**Lemma 3.1** Every cscc $C$ of a lossy channel system is downward closed.

Consider the empty buffer states (of the form $(s, \varepsilon)$) in a cscc $C$. Clearly they should be reachable from each other. More formally, we have:

**Lemma 3.2** Let $L$ be a lossy channel system. Let $?_{\text{empty}}$ be the set of all empty buffer states. $\gamma \in ?_{\text{empty}} \cap C$, for a cscc $C$, iff

1. $\gamma$ is reachable from $(s_0, \varepsilon)$.
2. $\forall \gamma' \in ?_{\text{empty}} \cap C$, $\gamma$ and $\gamma'$ are reachable from each other.
3. $\forall \gamma' \in ?_{\text{empty}} - C$, $\gamma'$ is not reachable from $\gamma$.

**Proof** Both if and only if parts follow from the definition of closed strongly connected component. ■

Since the set of empty buffer states of $L_{\psi}^+$ is finite and can, thus, be partitioned in only finitely many ways, we have the following:

**Lemma 3.3** The set of csccs of the labeled transition system associated with a lossy channel system is finite.

The previous lemma would be useless if there are no closed strongly connected components of a system. An example where there are no closed strongly connected components is a graph where (a) each natural number is a node, and (b) there is an edge from node $n$ to node $n + 1$. However, such graphs can not arise from lossy channel systems; this is due to the fact that there will always be back edges due to message losses. To that end we now prove that

**Lemma 3.4** Every infinite execution path in $L(L)$ visits only a finite number of sccs

**Proof** Given an infinite execution sequence $\pi = \gamma_0 \gamma_1 \ldots$ there is an infinite subsequence of related states, i.e., there is an infinite subsequence $\gamma_{i_0} \gamma_{i_1} \ldots$ such that $\forall j \geq 0 : i_j < i_{j+1}$ and $\gamma_{i_j} \leq \gamma_{i_{j+1}}$. Since there is a path in the labeled transition system from $\gamma_{i_j}$ to $\gamma_{i_{j+1}}$ and from $\gamma_{i_{j+1}}$ to $\gamma_{i_j}$ (by message losses) both of them are in the same scc. Thus all of the states in the infinite subsequence of $\pi$ (starting from $\gamma_{i_0}$ are in the same scc). Consequently, there can only be a finite number of sccs visited on any infinite execution path. ■

Given the previous lemma we can now state (formally) that pathological cases of graphs do not arise in our context.

**Lemma 3.5** From every state in $L(L)$ at least one cscc is reachable.
Proof Assume to the contrary. Then there is an infinite execution sequence visiting an infinite number of sccs, violating the previous lemma. 

We will now define the empty buffer states as the representatives of csccs, and show that they can be computed:

Definition 3.1 Given a labeled transition system $L(L)$, of a lossy channel system $L$, and a cscc $C$ of $L(L)$ define $\text{rep}(C) = (V_C, E_C)$ as the representative vertices and edges of $C$, where $V_C = \{s \in S | (s, \varepsilon) \in C\}$ and $E_C = \{s \xrightarrow{c'}_a s' | s \in V_C\} \cup \{s \xrightarrow{c} \varepsilon s' \exists x: \langle s, w'[c := \sigma x] \rangle \xrightarrow{a} \langle s', w[c := x] \rangle\}$.

That the representatives of the closed connected components can be computed follows from the decidability of reachability problem for lossy channel systems [1], which allows one to check for any pair of states, whether one is reachable from the other.

Theorem 3.2 ([1]) There is an algorithm which given a lossy channel system $L$, and two states $\gamma, \gamma'$, can decide whether $\gamma'$ is reachable from $\gamma$.

Given this algorithm one could easily identify those subsets of empty buffer states that form the representatives for csccs.

Theorem 3.3 There is an algorithm which given any lossy channel system $L$ will compute the representatives of the csccs of $L(L)$.

3.1 Sequences visiting csccs

The main reason for identifying csccs is that they provide the basis for sets of sequences whose probability is non-zero.

Definition 3.2 A set of execution sequences $\Pi$ of $L(L)$ is said to end in a cscc $C$ provided there exists a finite sequence $\pi'$ from $\langle s_0, \varepsilon \rangle$ to a state $\langle s, w \rangle$ of $C$ such that (a) $\langle s, w \rangle$ is the only state from $C$ in $\pi'$ and (b) all sequences $\pi \in \Pi$ are of the form $\pi'\pi''$ and all states visited on $\pi''$ are in $C$.

Note that in the definition given above it is implicit that all sequences in the set share a common prefix. A commonly used theorem in the context of (finite) Markov chains is that a set of sequences that do not visit all of the states of closed strongly connected component infinitely often has a zero measure. This fact is implicitly made use of by Courcoubetis and Yannakakis [13]. However, we are dealing with infinite state Markov chains. Fortunately, the underlying graph (i.e., the control graph) is finite which provides us with a similar result.

Lemma 3.6 Let $\Pi$ be a set sequences that end in a cscc $C$ but do not visit a state $e \in E_C$ infinitely often. The measure of $\Pi$ is 0.

Proof According to finite markov chain theory a set of sequences that end in a cscc and which does not visit a state infinitely often has measure zero (as the states visited form a transient set). Since from every state $\langle s, w \rangle$ in $C$ we can reach every other control state $s'$ represented in $C$, the control component is not dependent on the buffer content. Consequently, the result from finite state markov chain theory applies to the control states of the Markov chain associated with a PLCS $L$.

To transfer this result to edges of a PLCS $L$ let an edge $\epsilon : \langle s, p, a \rangle \xrightarrow{c} s'$ be given, where $c \# p$ could be either $c \# p$ or $c \# p$. Consider a new structure (say PLCS with silent actions) $L'$ with the edge $\epsilon$ replaced by edges $e_1 : \langle s, p, a \rangle \xrightarrow{s_{new}} s_{new}$ and $e_2 : s_{new} \xrightarrow{r_{new}} s'_{new}$; the second edge signals no change to the buffer contents. A Markov chain $L(L')$ (very much in the spirit of Definition 2.2) can
be constructed. Clearly, execution sequences and basic cylindrical sets of $L(L)$ and $L(L')$ are in correspondence with each other, and so are their measures. Since edge $\epsilon$ is absent infinitely often in a set of sequences $\Pi$ of $L(L)$ if the control node $s_{new}$ is absent infinitely often in the corresponding sequences of $L(L')$, we can infer that $\mu(\Pi) = 0$.

A direct implication of the lemma above is the following:

**Lemma 3.7** Let $\Pi$ be the set of all sequences, with a common prefix, that end in a cscC C of a lossy channel system $L(L)$, such that every sequence $\pi \in \Pi$ uses each edge $\epsilon \in E_G$ infinitely often. Then $\mu(\Pi) > 0$.

## 4 The cross product

In a manner similar to Vardi [27] and Courcoubetis and Yannakakis [13] we will take a cross-product of the control graphs of a lossy channel system and a Muller automaton, such that new automaton encodes both the sequences of the original lossy channel system and also its run on the Muller automaton. In particular, the product graph makes a move on both the lossy channel system and the Muller automaton on non-$\tau$ actions.

**Definition 4.1** Let $L = (S, Ch, \Sigma, \text{Act}, \rightarrow, s_0)$ and $B = (T, q_0, F)$, where $\tau \in \text{Act}$, and $T = (\text{Act} - \{\tau\}, Q, \Delta)$. Define $L \times B$ as a lossy channel system with the structure $(S_T, Ch, \Sigma, \text{Act}, \rightarrow_T, (s_0, q_0))$, where the set of states $S_T$ and the transition relation $\rightarrow_T$ are the least sets satisfying the following constraints:

- $\langle s_0, q_0 \rangle \in S_T$.
- If $\langle s, q \rangle \in S_T$ and $s \overset{\sigma, \tau}{\rightarrow} s'$ then $\langle s', q \rangle \in S_T$ and $\langle s, q \rangle \overset{\sigma, \tau}{\rightarrow} T \langle s', q \rangle$.
- If $\langle s, q \rangle \in S_T$ and $s \overset{\sigma, \alpha}{\rightarrow} s'$ then $\langle s', q \rangle \in S_T$ and $\langle s, q \rangle \overset{\sigma, \alpha}{\rightarrow} T \langle s', q \rangle$.
- If $\langle s, q \rangle \in S_T$, $s \overset{\sigma, \alpha}{\rightarrow} s'$, $\alpha \neq \tau$ and $(q, \alpha, q') \in \Delta$ then $\langle s', q' \rangle \in S_T$ and $\langle s, q \rangle \overset{\sigma, \alpha}{\rightarrow} T \langle s', q' \rangle$.
- If $\langle s, q \rangle \in S_T$, $s \overset{\sigma, \alpha}{\rightarrow} s'$, $\alpha \neq \tau$ and $(q, \alpha, q') \in \Delta$ then $\langle s', q' \rangle \in S_T$ and $\langle s, q \rangle \overset{\sigma, \alpha}{\rightarrow} T \langle s', q' \rangle$.

The product machine $L \times B$ is a lossy channel system in its own right, and thus $L(L \times B)$ is well defined. Consider an execution sequence of the form $\zeta = \langle \langle s_0, q_0 \rangle, \varepsilon \rangle \overset{\sigma_0}{\rightarrow} \langle \langle s_1, q_1 \rangle, w_1 \rangle \overset{\alpha_1}{\rightarrow} \ldots$. The sequence captures two pieces of information: the execution $\pi = \langle \langle s_0, \varepsilon \rangle \overset{\sigma_0}{\rightarrow} \langle s_1, w_1 \rangle \ldots \rangle$ of $L$ and the run of $\text{tr}(\pi)$ on the Muller automaton $B$, which is $q_0 q_1 \ldots$. To formally reason about these execution sequences define the notion of projection as follows:

**Definition 4.2** For every state $\kappa = \langle \langle s, w \rangle \rangle$ define $\kappa \mid L = \langle s, w \rangle$ and $\kappa \mid B = q$. For every execution sequence $\zeta = \langle \langle s_0, q_0 \rangle, \varepsilon \rangle \overset{\sigma_0}{\rightarrow} \langle \langle s_1, q_1 \rangle, w_1 \rangle \overset{\alpha_1}{\rightarrow} \ldots$ of $L \times B$, define $\zeta \mid L = \langle s_0, \varepsilon \rangle \overset{\sigma_0}{\rightarrow} \langle s_1, w_1 \rangle \ldots \rangle \mid B$ is defined similarly but with the proviso that states $\langle \langle s, q \rangle, w \rangle$ in a maximal sequence of $\tau$ transitions contribute only one instance of $q$.

The following facts follow from the definitions:

**Lemma 4.1** If $\langle s, w \rangle$ is a reachable state of $L(L)$ then there exists a state $\langle \langle s, q \rangle, w \rangle$ which is reachable in $L(L \times B)$.

**Proof (sketch)** The statement follows by an inductive argument on the length of the execution sequence which establishes that $\langle s, w \rangle$ is a reachable state of $L(L)$.■
Lemma 4.2 A sequence $\zeta$ is an execution sequence of $L(L+B)$ iff $\pi = \zeta \mid L$ is an execution sequence of $L$ and $\zeta \mid B$ is a run of $\text{tr}(\pi)$ on $B$.

Furthermore, for every sequence $\pi'$ of $L(L)$ there is an unique execution sequence $\zeta$ of $L(L+B)$ such that $\zeta \mid B = \pi'$.

Proof The first part is obvious, and the second follows from the fact that the Muller automaton $B$ is deterministic.

5 Compatibility of CSCCs and the main theorem

Since $L * B$ is a lossy channel system in its own right we can talk about the cscCs of $L(L+B)$. In particular, the representatives of the cscCs of $L(L+B)$ can be easily computed. Given a cscC $C'$ of $L(L*B)$ the structure $\text{rep}(C')$ is well defined. In the following we need the notion of a final cscC of $L(L+B)$ being compatible with a cscC of $L(L)$.

Definition 5.1 A cscC $C'$ of $L(L+B)$ is final provided its representative $\text{rep}(C') = (V_{C'}, E_{C'})$ is such $\{q \langle s, q \rangle \in V_{C'}\}$ is a final state set of the Muller automaton $B$.

The notion of compatibility is as follows:

Definition 5.2 Let $L$ be a lossy channel system and $B$ be a Muller automaton. A cscC $C'$ of $L(L+B)$ is compatible with a cscC $C$ of $L(L)$ provided

- A node $\langle s, q \rangle$, for some $q \neq -$, is in $V_{C'}$ iff $s$ is a node in $V_C$.
- An edge $\langle s, q \rangle \xrightarrow{c_{\sigma, a}} \langle s', q' \rangle$, for some $q \neq -$ and $q' \neq -$, is in $E_{C'}$ iff $s \xrightarrow{c_{\sigma, a}} s'$ is in $E_C$.
- An edge $\langle s, q \rangle \xrightarrow{c_{\sigma, a}} \langle s', q' \rangle$, for some $q \neq -$ and $q' \neq -$, is in $E_{C'}$ iff $s \xrightarrow{c_{\sigma, a}} s'$ is in $E_C$.

Sufficiency of compatibility Before we can show that compatibility is sufficient we provide a sufficient characterization of sequences that are accepted by the Muller automaton.

Lemma 5.1 Let $C$ be a cscC of $L(L)$ which is compatible with a final cscC $C'$ of $L(L+B)$. Let $\pi$ be an execution sequence of $L(L)$ which ends in $C$ and which takes all of the edges in $E_C$ infinitely often. Then $\text{tr}(\pi) \in \text{Lang}(B)$.

Proof Given that $B$ is a deterministic, and that $C$ and $C'$ are compatible, the sequence $\pi$ has a unique run $r$ on $B$. Furthermore there is a unique sequence $\zeta$ of $L(L+B)$ that ends in $C'$ such that $\zeta \mid B = r$ and $\zeta \mid L = \pi$.

The set of states $V_{C'} \mid B = Q_{C'}$, is one of the final state sets of $B$, as $C'$ is a final cscC. Since $r = \zeta \mid B$ and $Q_{C'} = V_{C'} \mid B$ we have $\text{inf}(r) \subseteq Q_{C'}$. To show that $\text{tr}(\pi) \in \text{Lang}(B)$ we need to demonstrate that $Q_{C'} = \text{inf}(r)$. Suppose $Q_{C'} - \text{inf}(r) \neq \emptyset$. We will show that this contradicts our assumption that $\pi$ visits all edges in $E_C$ infinitely often.

Let $q \in \text{inf}(r)$. This implies that there is some control state $\langle s, q \rangle$ that is visited infinitely often on $\zeta$. Let $q' \in Q_{C'} - \text{inf}(r)$. There exists a control state $\langle s', q' \rangle$ of $C'$ and $\langle s', q' \rangle$ does not appear infinitely often on $\zeta$. But there is a path from $\langle s, q \rangle$ to $\langle s', q' \rangle$ in the representative of the cscC $C'$, $\text{rep}(C')$. Let the path be $\langle s, q \rangle \xrightarrow{x_1, a_1} \langle s_1, q_1 \rangle \xrightarrow{x_2, a_2} \langle s_2, q_2 \rangle \ldots \xrightarrow{x_n, a_n} \langle s', q' \rangle$, where $x_i, s$ are of the form $c^m$ or $c!m$. Wlog, we assume that all of $a_1, \ldots, a_{n-1}$ are $\tau$ (if that is not the case, then pick the smallest $i \geq 1$ such that $a_i \neq \tau$, truncate the path at $\langle s_i, q_i \rangle$ and set $s'$ to $s_i$ and $q'$ to $q_i$). Recall that, by the definition of the cross product, the automaton does not change state on a $\tau$ action. Thus we have $q = q_1 = \ldots = q_{n-1}$. Since $(q =)q_{n-1} \neq q'$ we have that in $\langle s_{n-1}, q_{n-1} \rangle \xrightarrow{x_n, a_n} \langle s', q' \rangle$. 

Before we can get to the necessity of the notion of compatibility, we consider the proof of Lemma 5.2.

Lemma 5.2 Let $C$ be a csc$\infty$ of $L(L)$ and $C'$ a final csc$\infty$ of $L(L*B)$ such that $C$ and $C'$ are compatible. Then we have $\mu(\{\pi | \pi \in \text{Lang}(B)\}) > 0$.

Proof Consider a set of sequences $\Pi$ of $L(L)$ with a common prefix and which visit $C$. $\Pi$ can be partitioned into $\Pi_1$ and $\Pi_2$ where $\Pi_1$ contains sequences that visit all of the edges in $E_C$ infinitely often and $\Pi_2$ contains sequences that do not visit at least one edge of $E_C$ infinitely often. Since the two sets are complementary, the measure of $\Pi$ is the sum of the measures of $\Pi_1$ and $\Pi_2$. By Lemma 3.7, $\mu(\Pi_1) > 0$. By Lemma 5.1 all sequences in $\Pi_1$ are accepting. Thus $0 < \mu(\Pi_1) < \mu(\{\pi | \text{tr}(\pi) \in \text{Lang}(B)\})$, as needed.

Necessity of compatibility Before we can get to the necessity of the notion of compatibility, we will characterize the csc$\infty$s of $L*B$. The following is possible because $B$ is deterministic and is complete (i.e., every state has a transition on every action label).

Lemma 5.3 If $C'$ is a csc$\infty$ of $L*B$ then there is a csc$\infty$ $C$ of $L$ such that $C'|L \subseteq C$.

Proof (sketch) Assume there is no csc$\infty$ of $L$ such that the required relation holds. Since $C'$ is a csc$\infty$ the set of states $C'|L$ are connected, i.e., there is a path between every pair of states in the set. Clearly, this implies that $C'|L$ is a part of $\text{scc}(L)$. If $C'|L$ is not a csc$\infty$ then there is a state $\gamma \in C'|L$ and $\gamma' \not\in C'|L$ such that there is a path from $\gamma$ and $\gamma'$, but not the other way around. This implies that there is a state $\kappa \in C'$ and a state $\kappa' \in C'$ such that there is a path from $\kappa$ to $\kappa'$, but not the other way around – a contradiction. Consequently, the projection of a csc$\infty$ of $L*B$ is a csc$\infty$ of $L$.

Lemma 5.4 If $\mu(\{\pi | \text{tr}(\pi) \in \text{Lang}(B)\}) > 0$ then there is a csc$\infty$ $C$ of $L(L)$ which is compatible with a final csc$\infty$ $C'$ of $L(L*B)$.

Proof Let $\{\pi | \text{tr}(\pi) \in \text{Lang}(B)\} = \Pi$. Assume $\mu(\Pi) > 0$. Every infinite sequence has to end in some scc (by Lemma 3.4). One can write $\Pi = \Pi_{\text{scc}} \cup \Pi_{\text{acc}}$ where $\Pi_{\text{scc}}$ is the set of sequences that end in some scc and $\Pi_{\text{acc}}$ is the set of sequences that end in some scc (which is not closed). Since $\Pi_{\text{scc}} \cap \Pi_{\text{acc}} = \emptyset$, and since $\mu(\Pi_{\text{scc}}) > 0$ (an scc is a set of transient states of the Markov chain) we have $\mu(\Pi) = \mu(\Pi_{\text{scc}})$. Since there are only a finite number of csc$\infty$s there is at least one csc$\infty$ $C$, and a finite prefix $\pi'$, where $\Pi_C \subseteq \Pi_{\text{acc}}$ ends in $C$ (note all sequences in $\Pi_C$ have the same prefix $\pi'$) such that $\mu(\Pi_C) > 0$.

Pick a sequence $\pi$ in $\Pi_C$ that visits all of the edges of $E_C$. Clearly, this sequence has an accepting run on $B$. Consequently, there is a sequence $\zeta \in L(L*B)$ such that $\zeta|L = \pi$, $\zeta|B = r$ and $\inf(r)$ is a final state set of $B$. We first argue that $\zeta$ ends in a csc$\infty$ (say $C'$). Then we show that $C$ and $C'$ are compatible and that $C'$ is final.

Let $\zeta$ be such that it ends in an scc $C'$. If $C'$ is not a csc$\infty$, then there must be a path from a control state $\langle s_1, q_1 \rangle$ that appears on $\zeta$ to a state $\langle s_2, q_2 \rangle$ outside the scc. Since $\zeta|L = \pi$, we can infer that $s_1$ is a control state of $C$. As $\langle s_2, q_2 \rangle$ lies outside $C'$, there is a path from $\langle s_1, q_1 \rangle$ to $\langle s_2, q_2 \rangle$ but none from $\langle s_2, q_2 \rangle$ to $\langle s_1, q_1 \rangle$ in $C'$. This implies that in $C'$, there is a path from $s_1$ to $s_2$ but there is no path from $s_2$ to $s_1$, contradicting our assumption that $C$ is a csc$\infty$. Therefore $C'$ is a csc$\infty$.
From Lemma 5.3 we have $V_C \cap L \subseteq V_C$. Consider $V_C \cap L$, the projection of control states of $C'$ on $C$. Given that $\pi$ visits all states in $V_C$, $\zeta$ visits some subset of $V_C$, and $\zeta \not\subseteq L = \pi$ we have $V_C \subseteq V_C \cap L$. Thus $V_C = V_C \cap L$. In very much the same manner, by considering all of the sequences in $\Pi_L$ that visit all of the edges in $E_C$, we can also show that $E_C = E_C \cap L$. Thus $C$ and $C'$ are compatible. We are now left with showing that $C'$ is final.

Consider $V_C \cap B$. Let $q \in \text{inf}(r)$ and $q' \in V_C \cap B - \text{inf}(r)$. Just as in the proof of Lemma 5.1, it is can be shown that this leads to a contradiction to our assumption that $\pi$ visit all edges of $E_C$. Thus there can be no such $q'$. We conclude that $V_C \cap B = \text{inf}(r)$ and that $C'$ is final.

**Main theorem** Given proofs to both directions we can now infer the following:

**Theorem 5.1**

$$\mu(\{\pi \mid r(\pi) \in \text{Lang}(B)\}) > 0$$

iff there is a cscc $C$ of $\mathcal{L}(L)$ which is compatible with a final cscc $C'$ of $\mathcal{L}(L \ast B)$.

**The Algorithm** Given the Main theorem our algorithm can now be explained as follows: given a PLCS $L^P_\phi$ and a Muller automaton $B$ (a) construct the complement of $B$ (say, neg$(B)$), (b) construct $L \ast \text{neg}(B)$, and (c) check whether there is final cscc of $\mathcal{L}(L \ast B)$ which is compatible with a cscc of $\mathcal{L}(L)$. If the answer is yes then report that $\mu(L^P_\phi, B) \neq 1$ else report $\mu(L^P_\phi, B) = 1$.

Clearly, the preceding algorithm terminates, and by our Main theorem it is correct. We thus have the following:

**Theorem 5.2** There exists an algorithm which when given an PLCS $L^P_\phi$ and a Muller automaton $B$, it can decide whether $\mu(L^P_\phi, B) = 1$.

**Corollary 5.1** There exists an algorithm which when given an PLCS $L^P_\phi$ and a LTL formula $\phi$ it can decide whether the probability of execution sequences that satisfy $\phi$ is 1.

### 6 Discussion

The central result of our paper is that it is possible to decide whether the set of sequences of a probabilistic lossy channel system $L^P_\phi$ satisfying a LTL formula $\phi$, or a Muller automaton $B_\phi$, has probability 1. This might be surprising given that the traditional model-checking problem for non-probabilistic Lossy channel systems is undecidable, and given that our algorithm does not depend upon the probabilities involved. Clearly, the only explanation is that the two problems are not equally powerful. To illustrate this, consider the LTL specification $\phi = \neg \Box \Diamond @ s$ ($\Diamond s$ is an atomic proposition that holds in $s$ and does not hold in any $s' \neq s$) which is true of sequences in which the control state $s$ is not visited infinitely often. Assume that for a Probabilistic lossy channel system $L^P_\phi$ the probability of $\phi$ being true is 1. Does this mean that there are no infinite execution sequences of $L^P_\phi$ in which the control state $s$ is visited infinitely often? This is not the case. By saying that the probability of $\phi$ is 1, we are essentially saying that there are no csccs of $L^P_\phi$ which contains a control state $s$. Consequently, it is possible that there is a scc $C$, which is not a cscc, containing the control state $s$; this could engender an execution sequence in which the control state $s$ is visited infinitely often. However, the measure of such sequences is 0. Consequently, our decidability results are not at odds with the undecidability of the Recurrent Path Problem (which checks whether a control state $s$ can be visited infinitely often) for Lossy channel systems in [2].
Given the afore-mentioned distinction between model-checking and probabilistic model-checking, one could ask - "so what is the use of probabilistic model-checking?" The use of probabilistic model-checking comes from the fact that the theory of performance analysis describes satisfaction of properties in terms of steady state probabilities. The work in this paper can be seen as "describing satisfaction of LTL properties in terms of steady control-state probabilities." This vibes well with the notion that reactive systems (such as network protocols) are supposed to work forever, and that their long time behavior is what is important. Furthermore, ignoring non-closed SCCs is tantamount to making fairness assumptions (such as that with enough number of tries a message sent will be received by the intended recipient); a fact captured by the Kooman's fair abstraction rule [24].

Finally, the utility of probabilistic model-checking, and the results presented here, is that it provides a combined approach to formal methods based reasoning and performance evaluation. The interpretation of our result is that such a combined analysis could possibly be carried out when formal methods based reasoning (by itself) might be impossible.

We close the paper with a final remark that Probabilistic lossy channel systems can be used to specify lower level protocols (and network programs) that have to cope with failure in message delivery more realistically, and that having a model-checking procedure for it increases its utility.

References


