Electronic Communications of the EASST
Volume 70 (2014)

Proceedings of the
14th International Workshop on
Automated Verification of Critical Systems (AVoCS 2014)

Reachability and Reward Checking for Stochastic Timed Automata

E. Moritz Hahn, Arnd Hartmanns and Holger Hermanns

15 pages

Guest Editors: Marieke Huisman, Jaco van de Pol
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Reachability and Reward Checking for Stochastic Timed Automata[†]

E. Moritz Hahn[1], **Arnd Hartmanns**[2] **and Holger Hermanns**[2]

[1] State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, China

[2] Saarland University – Computer Science
Saarbrücken, Germany

Abstract: Stochastic timed automata are an expressive formal model for hard and soft real-time systems. They support choices and delays that can be deterministic, nondeterministic or stochastic. Stochastic choices and delays can be based on arbitrary discrete and continuous distributions. In this paper, we present an analysis approach for stochastic timed automata based on abstraction and probabilistic model checking. It delivers upper/lower bounds on maximum/minimum reachability probabilities and expected cumulative reward values. Based on theory originally developed for stochastic hybrid systems, it is the first fully automated model checking technique for stochastic timed automata. Using an implementation as part of the MODEST TOOLSET and four varied examples, we show that the approach works in practice and present a detailed evaluation of its applicability, its efficiency, and current limitations.

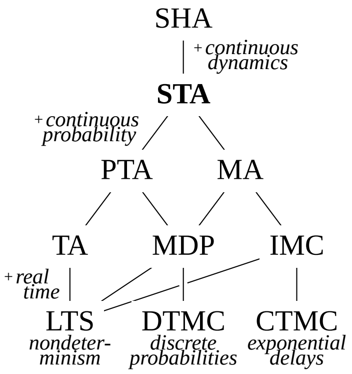
Keywords: stochastic timed automata, probabilistic reachability, expected rewards

1 Introduction

Proper consideration of quantitative aspects is crucial to formally model and analyse many complex safety-critical or economically vital systems. Key quantities are *time*, to represent e.g. timeouts and delays, and *probabilities*, to model the quantified uncertainty that appears, for example, in randomised algorithms, as disturbances like random failures, and as randomised delays. Additionally, nondeterminism is a key feature for verification that enables abstraction, concurrency, and the specification of unquantified uncertainty. We need to analyse properties like the probability of (un)desired behaviour, the expected time to success, or the probability of an error within a given amount of time.

A suitable model for these kinds of systems are stochastic timed automata (STA). They allow nondeterministic decisions, real time aspects, continuous and discrete probabilistic choices, and any combination thereof. STA had been introduced as the original formal semantics of the high-level compositional modelling language MODEST [BDHK06]. They are at the heart of a large spectrum of compositional models, summarised in Figure 1, rooted in labelled transition

[†] This work is supported by the EU 7th Framework Programme under grant agreements 295261 (MEALS) and 318490 (SENSATION), the DFG Transregional Collaborative Research Centre SFB/TR 14 AVACS, the CAS/SAFEA International Partnership Program for Creative Research Teams, the Chinese Academy of Sciences Fellowship (grant no. 2013Y1GB0006), and the research fund for International Young Scientists (grant no. 61350110518).



Key:	SHA	stochastic hybrid automata
	STA	stochastic timed automata
	PTA	probabilistic timed automata
	MA	Markov automata
	TA	timed automata
	MDP	Markov decision processes (or probabilistic automata)
	IMC	interactive Markov chains
	LTS	labelled transition systems
	DTMC	discrete-time Markov chains
	CTMC	continuous-time Markov chains

Figure 1: Stochastic timed automata and related models

systems and Markov chains. Modest has since been extended with support for continuous dynamics [HHHK13] based on the model of stochastic hybrid automata [FHH$^+$11]. The compositionality properties of STA in turn rest on results established by Strulo, Bravetti and especially D'Argenio [BD04, BG02, DK05, HS00]. STA can also be viewed as generalised semi-Markov processes (GSMP) extended with discrete and continuous nondeterminism.

The Modest Toolset, which is available at www.modestchecker.net, provides analysis tools for a variety of these models [HH14]. However, so far it did not support the genuine analysis of full STA models with nondeterministic decisions, and that is what this paper is about: We present an algorithm to compute upper/lower bounds on maximum/minimum reachability probabilities and expected cumulative reward values in a given STA. It uses abstraction to convert the STA into a PTA, which can then be analysed using existing PTA model checking techniques [NPS13]. We show the correctness of the abstraction for the considered properties. The underlying theory was originally developed for stochastic hybrid systems [FHH$^+$11, Hah13]; we explain how we take advantage of the specialisation to timed systems to improve scalability, usability and applicability. We implemented the new approach in the Modest Toolset, which allows us to investigate its effectiveness and efficiency using four different example models.

Related work. Kwiatkowska et al. [KNSS00] have pioneered the foundational basis of STA model checking with their work on timed automata with generally distributed clocks, verified against properties in probabilistic timed CTL. They use a semantics based on the region graph where regions are further partitioned to cater for the stochastic behaviour. The main differences to what we present in this paper are that our approach can handle distributions with unbounded support (e.g. the exponential and normal distributions), supports expected rewards, and that we avoid the region construction. We also show a working implementation, which instead currently uses a digital clocks semantics, but this can be interchanged with other approaches. In case an STA only uses bounded-support distributions (e.g. the continuous uniform one), our approach provides the same error bounds. However, we do not provide error bounds for the general case.

Other related approaches that we find are based on statistical model checking [DLL$^+$11], numerical discretisation [LHK01], discrete event simulation [HS00], or state classes [BBH$^+$13] (on a different model also called STA). However, all of these either implicitly or explicitly exclude the presence of nondeterminism, and thus work in the GSMP realm instead. As an example, consider the "STA" model of [BBJM12] (which is closely related to the one of [BBH$^+$13]): There, a single distribution is sampled on every edge, the result being the exact sojourn time in the following location. In comparison, our model of STA also supports continuous and discrete nondeterminism as well as multiple samplings per edge and multiple sampled variables that can memorise their values over several edges/locations.

In particular, the method we present in this paper is geared towards correctly handling the general combination of stochastics and nondeterminism. Dedicated approaches for deterministic models provide better precision or performance for that special case. We return to this tradeoff in our evaluation in Section 6, where we look at two deterministic models for comparison, and two nondeterministic case studies that can only be handled correctly with our new approach.

2 Preliminaries

We use $\mathbb{R}_0^+$ to denote the set of nonnegative real numbers and $\mathbb{N}^+$ for the positive natural numbers. For a set S, $\mathscr{P}(S)$ denotes its powerset. We assume familiarity with general notions and constructions from probability theory. Due to space constraints, we do not consider possible measurability issues (see e.g. [Hah13, Chapter 5] for discussions concerning a more general model). For all probability distributions, we assume an according (Borel) space to be given. By $\mathrm{Prob}(\Omega)$ we denote the set of all probability measures on the sample space Ω. The *Dirac distribution* $\mathscr{D}(x) \in \mathrm{Prob}(\Omega)$ is s.t. we have $\mathscr{D}(x)(A) = 1$ if $x \in A$ and $\mathscr{D}(x)(A) = 0$ otherwise. By $[\forall i\colon x_i \mapsto p_i]$ or $[x_1 \mapsto p_1, \ldots, x_n \mapsto p_n]$ we denote the distribution $\sum_i p_i \mathscr{D}(x_i)$.

Given a set *Var* of *variables* where each variable x has an associated domain (or type) $\mathrm{Dom}(x)$, we let *Val* denote the set of variable *valuations*, i.e. of functions $\mathit{Var} \to \bigcup_{x \in \mathit{Var}} \mathrm{Dom}(x)$ where $v \in \mathit{Val} \Rightarrow \forall x \in \mathit{Var}\colon v(x) \in \mathrm{Dom}(x)$. $\mathbf{0} \in \mathit{Val}$ assigns zero to every variable. By *Exp* we denote the set of *expressions* over the variables in *Var*. We write $e(v)$ for the *evaluation* of expression e in valuation v. We consider three restricted classes of expressions: Boolean expressions *Bxp*, arithmetic expressions *Axp* and sampling expressions $\mathit{Sxp} \supsetneq \mathit{Axp}$ that may include probability distributions. The set of *assignments* is $\mathit{Asgn} = \mathit{Var} \times \mathit{Sxp}$ such that $\langle x, e \rangle \in \mathit{Asgn} \Rightarrow \forall v \in \mathit{Val}\colon e(v) \in \mathrm{Dom}(x)$. The modification of $v \in \mathit{Val}$ according to $u \in \mathit{Asgn}$ is written as as $[\![u]\!](v)$. A set of assignments is called an update, and notation for assignments can be lifted to updates. A *symbolic probability distribution* for a set S is a function $f \in S \to \mathit{Axp}$ that maps elements of S to weights s.t. the support $\{\, s \in S \mid f(s) \neq 0 \,\}$ is countable. Given a valuation for the variables appearing in these weights, a symbolic distribution can be turned into the concrete probability distribution given by the ratios of individual weights over the sum of all weights in the support. We only consider *proper* symbolic distributions: those where all weights evaluate to positive numbers and the sum of all weights is finite (i.e. convergent) and nonzero, for all relevant valuations.

3 Stochastic Timed Automata

As a generalisation of timed automata, stochastic timed automata deal with time through *clock variables* (or *clocks*). Clocks take values in $\mathbb{R}_0^+$ and advance synchronously over time with rate 1. If $v \in \mathit{Val}$ and $t \in \mathbb{R}_0^+$, then $v + t$ denotes the valuation where all clocks have been incremented

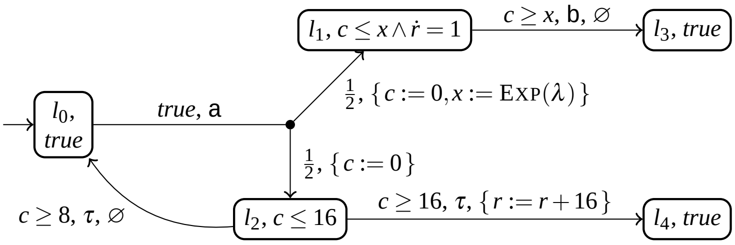

l_1, $c \leq x \wedge \dot{r} = 1$
$c \geq x$, b, $\varnothing$
l_3, *true*
$\frac{1}{2}$, $\{ c := 0, x := \textsc{Exp}(\lambda) \}$
l_0, *true*
true, a
$\frac{1}{2}$, $\{ c := 0 \}$
$c \geq 8$, τ, $\varnothing$
l_2, $c \leq 16$
$c \geq 16$, τ, $\{ r := r + 16 \}$
l_4, *true*

Figure 2: An example stochastic timed automaton

by t. *Clock constraints* are expressions in *Bxp* of the form

$$\mathscr{CC} ::= b \mid \mathscr{CC} \wedge \mathscr{CC} \mid \mathscr{CC} \vee \mathscr{CC} \mid c \sim e \mid c_1 - c_2 \sim e$$

where $\sim \in \{ >, \geq, <, \leq, =, \neq \}$, c, c_1, c_2 are clocks and $b \in Bxp$, $e \in Axp$ are clock-free expressions. If all e are of integer type, we have an *integer* clock constraint. A clock constraint that does not contain the last case (where two clocks are compared) is *diagonal-free*. If all comparison operators $\sim$ used in a clock constraint are in $\{ \geq, \leq, = \}$, it is *closed*.

Definition 1 A *stochastic timed automaton* (STA) is a 6-tuple $\langle Loc, Var, A, E, l_{init}, Inv \rangle$ where Loc is a countable set of locations, $Var \supseteq \mathscr{C}$ is a finite set of variables with a subset of clocks $\mathscr{C}$, A is the automaton's finite alphabet, $E \in Loc \to \mathscr{P}(\mathscr{CC} \times A \times Wxp)$ is the edge function, $l_{init} \in Loc$ is the initial location, and $Inv \in Loc \to \mathscr{CC}$ is the invariant function. An edge consist of a guard that determines when the edge is enabled, an action label, and a symbolic probability distribution over updates and target locations in $Wxp = \mathscr{P}(Asgn) \times Loc \to Axp$. We also write $l \xrightarrow{g,a} \mathscr{W}$ for $\langle g, a, \mathscr{W} \rangle \in E(l)$. The invariant function maps each location to an expression that allows time to pass as long as it evaluates to *true*.

We can equip STA with *rewards*, which can be seen as real-valued variables available to external observers only (i.e. they can be used during verification, but not be read in guards etc.). They advance at a certain rate in locations and can be increased when taking an edge:

Definition 2 A *reward* $r = \langle Rew_{Loc}, Rew_E \rangle \in (Loc \to Axp) \times (E \to Axp)$ for an STA as above assigns *rate rewards* to its locations and *edge rewards* to edges.

Example 1 *The graphical representation of an example STA with reward r is shown in Figure 2. Locations contain their name, invariant and rate reward (when not zero). Edges are shown either as simple arrows labelled with guard, action and update if they lead to a single update/location pair with probability* 1*, or as split arrows with an intermediate node otherwise. Edge rewards are included in updates. The example automaton contains a probabilistic choice on the edge labelled* a*. Out of* l_2*, the edge to* l_4 *can only be taken after a deterministic delay of* 16 *time units, while the one back to* l_0 *can be taken after any delay nondeterministically chosen out of* $[8, 16]$*. After* 16 *time units, the choice of edge in* l_2 *thus becomes nondeterministic. The delay incurred in* l_1*, on the other hand, is stochastic:* $x := \textsc{Exp}(\lambda)$ *assigns to* x *a value sampled from the exponential distribution with rate* λ*, thus the delay is exponentially distributed with rate* λ*. The reward* r *keeps track of the time spent in* l_1*, and is increased by* 16 *upon entering* l_4*.*

The semantics of STA is given in terms of timed probabilistic transition systems [BDHK06]:

Definition 3 A *timed probabilistic transition system* (TPTS for short) is a 4-tuple $\langle S, A, T, s_{init} \rangle$ where S is an (uncountable) set of states, $A = \mathbb{R}_0^+ \uplus A'$ is the system's (uncountable) alphabet that can be partitioned into delays in $\mathbb{R}_0^+$ and discrete actions in A', $T \in S \to \mathscr{P}(A \times \text{Prob}(S))$ is the transition function, and $s_{init} \in S$ is the initial state. We also write $s \xrightarrow{a} \mu$ for $\langle a, \mu \rangle \in T(s)$. For every delay-labelled transition $\langle x, \mu \rangle \in T(s)$, $x \in \mathbb{R}_0^+$, we require that $\exists s' \in S \colon \mu = \mathscr{D}(s')$, $\langle x, \mu' \rangle \in T(s) \Rightarrow \mu = \mu'$ *(time determinism)*, and $\langle x + x', \mathscr{D}(s') \rangle \in T(s) \Leftrightarrow \exists s'' \in S \colon \langle x, \mathscr{D}(s'') \rangle \in T(s) \wedge \langle x', \mathscr{D}(s') \rangle \in T(s'')$ for $x' \in \mathbb{R}_0^+$ *(time additivity)*.

Definition 4 A *reward structure* for a TPTS is a function $\text{rew} \in T \to \mathbb{R}_0^+$ assigning a nonnegative reward to each of its transitions.

Definition 5 The semantics of an STA $M = \langle Loc, Var, A, E, l_{init}, Inv \rangle$ is defined as the TPTS $[\![M]\!] = \langle Loc \times Val, \mathbb{R}_0^+ \uplus A, T_M, \langle l_{init}, \mathbf{0} \rangle \rangle$ where T_M is the smallest function that satisfies

$$\frac{l \xrightarrow{g,a}_E \mathscr{W} \quad g(v)}{\langle l, v \rangle \xrightarrow{a}_{T_M} \mu_{\mathscr{W}}^v} \; (jump) \qquad \frac{t \in \mathbb{R}^+ \quad \forall t' \leq t \colon (Inv(l))(v + t')}{\langle l, v \rangle \xrightarrow{t}_{T_M} \mathscr{D}(\langle l, v + t \rangle)} \; (delay)$$

where for $l' \in Loc$ and measurable $V' \subseteq Val$ we have

$$\mu_{\mathscr{W}}^v(\langle l', V' \rangle) \stackrel{\text{def}}{=} \textstyle\sum_{l \in Loc, U \in \mathscr{P}(Asgn)} \pi_{\mathscr{W}}^v(\langle U, l \rangle) \cdot \mu_U^v(V')$$

where $\pi_{\mathscr{W}}^v$ is the discrete probability distribution for the symbolic distribution $\mathscr{W}$ in valuation v and $\mu_U^v(V')$ returns the probability of V' corresponding to the sampling expressions in update U.

The *jump* inference rule creates action-labelled transitions for the discrete jumps corresponding to taking an edge in the STA. These transitions therefore go from a state into a continuous distribution over target states according to the sampling expressions in the assignments. Inference rule *delay* creates real-labelled transitions that represent the passage of time whenever this is allowed by the invariants. They always lead into Dirac distributions, i.e. a single target state.

Definition 6 The semantics of a reward r for an STA M is a reward structure $[\![r]\!] \colon T_M \to \mathbb{R}_0^+$ for the TPTS semantics $[\![M]\!]$. For transitions labelled with time actions $t \in \mathbb{R}_0^+$, it assigns a reward of t times the location reward rate according to Rew_{Loc}. For A-labelled transitions, the reward value is as defined by Rew_E for the STA edge inducing the TPTS transition.

3.1 Reachability Probabilities and Expected Rewards

For a given STA, we want to answer questions of the form "what is the probability of reaching a certain set of states from the initial state" and "what is the expected accumulated reward when a certain set of states is reached for the first time". These *properties* ask for the computation of reachability probabilities and expected rewards. Since STA may be nondeterministic, we quantify over the resolutions of nondeterminism by asking for *maximum* or *minimum* values. For a given TPTS $M = \langle S, A, T, s_{init} \rangle$, we now define paths and schedulers:

Definition 7 The set of *finite paths* is $\text{Paths}_M^{\text{fin}} \stackrel{\text{def}}{=} S \times (A \times \text{Prob}(S) \times S)^*$. The last state of the finite path $\beta = s_0 a_0 \mu_0 s_1 a_1 \mu_1 \ldots s_n$ is $\text{last}(\beta) \stackrel{\text{def}}{=} s_n$. A *scheduler* is a function $\sigma \in \text{Paths}_M^{\text{fin}} \to$

$\text{Prob}(A \times \text{Prob}(S))$ so that for each $\beta \in \text{Paths}_M^{\text{fin}}$ we have $\sigma(\beta)(A \times \text{Prob}(S) \setminus T(\text{last}(\beta))) = 0$. A scheduler σ induces the stochastic processes $X_M^\sigma(\cdot)$ of the current state of M and $Y_M^\sigma(\cdot)$ of the transition chosen by σ in the current state. It is *time-divergent* if $\text{Prob}(\sum_{i=0}^{\infty} f(Y_M^\sigma(i)) = \infty) = 1$ for $f(s \xrightarrow{a} \mu) = a$ if $a \in \mathbb{R}_0^+$ and $f(s \xrightarrow{a} \mu) = 0$ otherwise. We denote the set of all time-divergent schedulers of M by $\mathfrak{S}_M$.

A scheduler assigns probabilities to sets of enabled action-distribution pairs depending on the history seen so far. It resolves the nondeterminism in a TPTS so as to obtain probability measures, allowing to derive according stochastic processes. The semantics of the two kinds of properties we consider for STA can then be defined on the TPTS semantics in the usual way using measurable sets of paths and the cylinder construction. Given a set of states B, we are interested in minimal/maximal values, that is infima/suprema over all $\sigma \in \mathfrak{S}_M$. The *reachability probability* induced by σ is defined as $\text{Prob}(\exists i \geq 0\colon X_M^\sigma(i) \in B)$, i.e. the measure of paths with a state in B. The *expected accumulated reward* is $\mathbf{E}(\sum_{i=0}^{X_M^\sigma(i) \in B} [\![r]\!](Y_M^\sigma(i)))$ if $\text{Prob}(\exists i \geq 0\colon X_M^\sigma(i) \in B) = 1$ and ∞ otherwise. It is thus the expected reward accumulated along paths provided B is reached eventually; otherwise the expected value is infinity. As the values of clocks are explicit in TPTS, timed properties can be specified by referring to these values directly in the characterisation of B, e.g. referring to an extra clock that is never reset to specify time bounds.

Example 2 *We are interested in the probability of reaching* l_3 *or* l_4 *within at most* t *time units in the STA of the previous example. The minimum probability is* 0 *because the invariant of* l_0 *allows us to stay there forever. If* $t < 8$*, we can only reach* l_3 *and thus compute the maximum probability using the cdf of the exponential distribution: it is* $p = \frac{1}{2} \cdot (1 - \mathrm{e}^{-\lambda t})$*. If* $t \geq 16$*, we can also reach* l_4 *and the result is* $p + \frac{1}{2}$*. For* $t \in [8, 16)$*, we get* $p' = \frac{1}{2} \cdot (1 - \mathrm{e}^{-\lambda t}) + \frac{1}{4} \cdot (1 - \mathrm{e}^{-\lambda(t-8)})$ *by going back to* l_0 *from* l_2 *as soon as possible. Observe that* $p = p'$ *for* $t = 8$*, but for* $t = 16$*,* $p' \neq \frac{1}{2} + p$*: here, the nondeterministic choice available in* l_2 *makes an important difference.*

Now, let us look at the (time-unbounded) minimum and maximum expected reward r *when we reach* l_3 *or* l_4*. By definition, since there is a scheduler that reaches those locations with probability less than* 1 *(by staying in* l_0 *forever), the maximum value is* ∞*. If* $\lambda \geq \frac{1}{16}$*, the minimum value that we can achieve is* $\frac{1}{\lambda}$ *by always returning to* l_0 *from* l_2*; otherwise, it is* $\frac{1}{2} \cdot (16 + \frac{1}{\lambda})$*.*

3.2 Model Context

STA are related to many other automata models (cf. Figure 1). Of particular interest for this paper are *stochastic hybrid automata* (SHA) and *probabilistic timed automata* (PTA): The analysis technique we present is based on an existing one for SHA, and it involves the transformation of STA into PTA that are subsequently model checked using the digital clocks approach for PTA.

SHA [FHH+11] add continuous variables to STA. These can change over time according to differential (in)equations specified by the invariants. In contrast to clocks, they can also appear on the right-hand side of assignments, in particular in sampling expressions. SHA thus combine hybrid system behaviour (as in hybrid automata) with stochastic sampling and delays (as in STA).

PTA are the special case of STA where all clock constraints are integer and no continuous probability distributions are used. All delays and choices are thus based on discrete (usually finite-support) distributions, or nondeterministic. A number of techniques to model check PTA exist [NPS13]. In this paper, we use the *digital clocks* approach because it supports both reachability probabilities and expected rewards: Clocks are replaced by (bounded) integer variables, and self-loop edges are added to increment them synchronously as long as the location invariant is satisfied. This turns the PTA into a (finite) *Markov decision process* (MDP) where reachability probabilities and expected rewards can be computed using standard techniques. The results are correct for the original PTA whenever all clock constraints are closed and diagonal-free.

4 Checking Reachability and Rewards

We use a combination of abstraction and probabilistic model checking to compute bounds on reachability probabilities and expected reward values for STA. This works as follows: First, the continuous distributions that occur in the STA are abstracted by a combination of discrete probabilistic choices and continuous nondeterminism. The result is a PTA. The digital clocks approach is used to convert that into a finite MDP. Standard techniques like value iteration can now be used to derive maximum/minimum reachability probabilities and expected rewards. The results are upper/lower bounds on the corresponding values in the original STA. This approach is a special case of a technique developed for SHA safety verification [FHH^{+}11] and reward-based analysis [Hah13], which was (partly) implemented in the prohver tool [HHHK13]. By specialising for STA, we gain scalability, improve usability by requiring less user input and improving automation, and are able to compute useful lower bounds on minimum probabilities.

4.1 Abstracting Continuous Distributions

In the first step, the support of a continuous distribution is divided into a number of intervals and the probability of each interval is computed. The continuous sampling is then replaced by a probabilistic choice over the intervals with the computed probabilities, followed by a nondeterministic choice of which concrete value to pick from the chosen interval. When using prohver, the probabilities for the intervals had to be concrete real values due to the PHAVER backend used. In our new approach, we can map to PTA with probabilities that depend on state variables (but not on clocks or variables that were previously sampled). Since PTA allow only integer clock constraints, the choice of intervals is limited to those with integer bounds. We always overapproximate continuous distributions with intervals of unit width 1 aligned on integer bounds in the current implementation; all integer time points are anyway enumerated in the resulting MDP's state space. For distributions with unbounded support, such as the exponential or normal distribution, we generate as many unit width intervals as needed to cover a probability mass of $1-\rho$ and then add half-open intervals for the residual of the support. Instead of a set of intervals as with prohver, the only parameter of our approach therefore is this *residual probability* ρ. We use a default of $\rho = 0.05$ unless stated otherwise.

Example 3 *For the STA of Example 1, we show the PTA overapproximation for the case that a single unit-width interval is sufficient to cover* $1-\rho$ *probability in Figure 3. With* $\rho = 0.05$*, this is ensured provided* $\lambda \geq 3$*. We use* $\geq_\exists$ *and* $\leq_\exists$ *to denote interval comparisons. They are satisfied whenever there exists some value in the interval such that the concrete comparison is satisfied. This amounts to a comparison with the upper bound for* $\leq_\exists$ *and with the lower bound for* $\geq_\exists$

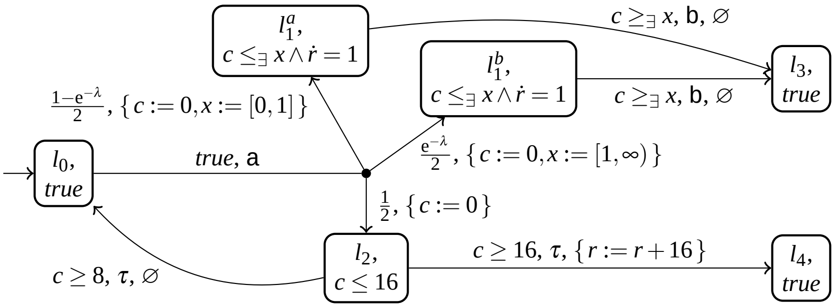

l_1^a,
$c \leq_\exists x \wedge \dot{r} = 1$
l_1^b,
$c \leq_\exists x \wedge \dot{r} = 1$
$c \geq_\exists x$, b, $\varnothing$
$c \geq_\exists x$, b, $\varnothing$
l_3,
true
$\frac{1-\mathrm{e}^{-\lambda}}{2}$, $\{c := 0, x := [0,1]\}$
l_0,
true
true, a
$\frac{\mathrm{e}^{-\lambda}}{2}$, $\{c := 0, x := [1,\infty)\}$
$\frac{1}{2}$, $\{c := 0\}$
l_2,
$c \leq 16$
$c \geq 16$, τ, $\{r := r + 16\}$
l_4,
true
$c \geq 8$, τ, $\varnothing$

Figure 3: A PTA abstraction of the example STA

when the interval operand is on the right-hand side.

4.2 Correctness

We now show that, in the PTA that is constructed as described above, the maximum/minimum reachability probabilities and expected reward values are indeed upper/lower bounds for the corresponding values in the original STA. We first define the effect of abstraction more formally:

Definition 8 Consider an STA $M = \langle Loc, Var, A, E, l_{init}, Inv\rangle$ and a (potentially infinite) family of sets $\mathscr{A} = \langle B_i \rangle_{i\in I}$. Each abstract state $B_i \subseteq Loc \times Val$ subsumes certain concrete states of $[\![M]\!]$, we have $\bigcup_i B_i = Loc \times Val$ so that all states are covered. We require that an abstract state only subsume concrete states of the same location. Assume $B_{\text{init}} \ni \langle l_{init}, \mathbf{0}\rangle$, and B_i, B_j with $i \neq j$ disjoint. The *abstraction TPTS* is defined as $\text{abs}(M, \mathscr{A}) \overset{\text{def}}{=} \langle \mathscr{A}, A \uplus \mathbb{R}_0^+, T_M^{\text{abs}}, B_{\text{init}}\rangle$ where T_M^{abs} is defined similar to Definition 5 with the *jump* rule being

$$\frac{l \xrightarrow{g,a}_E \mathscr{W} \quad \langle l, v\rangle \in B_i \quad g(v)}{B_i \xrightarrow{a}_{T_M^{\text{abs}}} [\forall j : B_j \mapsto \mu_{\mathscr{W}}^v(B_j)]}$$

where $\mu_{\mathscr{W}}^v$ is as in Definition 5. We require $\mathscr{A}$ to be defined s.t. all induced $[\forall j : A_j \mapsto \mu_{\mathscr{W}}^v(A_j)]$ have finite support. Timed transitions are defined accordingly. We assign rewards to abstract states according to the rate for its location and the rewards of the edges originating from there.

In the context of this paper, $\mathscr{A}$ is obtained by splitting the possible values sampling variables can take into unit width or half-open intervals. This construction ensures the finite-support requirement. For instance, for a single sampling variable x, all concrete states where x is sampled to take values between 1 and 2 are subsumed by a single abstract state. For multiple sampling variables, abstract states are built from the cross product of intervals.

Lemma 1 *For an STA M with abstraction set $\mathscr{A}$ and some set of states B the maximal (minimal) probability/reward value to reach B in* $\text{abs}(M, \mathscr{A})$ *is not lower (not higher) than the maximal (minimal) probability/reward value in* $[\![M]\!]$.

Proof. We only consider disjoint abstract states. Non-disjoint ones (from overlapping intervals) would however not affect correctness, yet induce imprecision due to additional transitions in the abstraction. Let $M = \langle Loc, Var, A, E, l_{init}, Inv\rangle$ and $\mathscr{A} = \langle B_i \rangle_{i\in I}$. We define the *intermediate* *abstraction* $M' \overset{\text{def}}{=} \langle Loc \times Val, A \uplus \mathbb{R}_0^+, T'_M, \langle l_{init}, \mathbf{0} \rangle \rangle$ by replacing *jump* of Definition 5 by

$$\frac{l \xrightarrow{g,a}_E \mathscr{W} \quad g(v) \quad \langle s'_j \rangle_{j \in I} \text{ s.t. } \forall j \in I\colon s'_j \in B_j}{\langle l, v \rangle \xrightarrow{a}_{T'_M} [\forall j \in I\colon s'_j \mapsto \mu_{\mathscr{W}}^v(B_j)]} \quad .$$

Let f map paths from the intermediate abstraction to the semantics $[\![M]\!]$, so for a path $\beta = s_0 a_0 [\forall j\colon s'_j \mapsto \mu_{\mathscr{W}}^v(B_j)] s_1 a_1 \ldots$ we have $f(\beta) \overset{\text{def}}{=} s_0 a_0 \mu_{\mathscr{W}}^{v_0} s_1 a_1 \ldots$.

For $\sigma \in \mathfrak{S}_{[M]}$ we construct $\sigma' \in \mathfrak{S}_{M'}$. Consider path β with $\mathrm{last}(\beta) = \langle l, v \rangle$. W.l.o.g. consider a subset $A = \{a\} \times A_{\mathrm{dist}} \subseteq A \times \mathrm{Prob}(S)$ of the possible successors when choosing edge $e = l \xrightarrow{g,a} \mathscr{W} \in E$ with $\langle l, v \rangle \xrightarrow{a}_{T_M} \mu_{\mathscr{W}}^v$. Let $\langle S_i \rangle_i \subseteq \mathscr{A}_i$ be the finite set of abstract states for which $\mu_{\mathscr{W}}^v(S_i) > 0$. Define $\mu_i \in \mathrm{Prob}(S_i)$ as $\mu_i(A_i) \overset{\text{def}}{=} \mu_{\mathscr{W}}^v(A_i) / \mu_{\mathscr{W}}^v(S_i)$ for measurable $A_i \subseteq S_i$ and denote their product measure by $\mu_{\mathrm{prod}} \in \mathrm{Prob}(\bigtimes_i S_i)$. Define $U \overset{\text{def}}{=} \{[\forall i\colon s'_i \mapsto \mu_{\mathscr{W}}^v(S_i)] \mid \forall i\colon s'_i \in S_i\}$, function $g([s'_1 \mapsto p_1, \ldots, s'_n \mapsto p_n]) \overset{\text{def}}{=} (s'_1, \ldots, s'_n)$, and $\mu(B) \overset{\text{def}}{=} \mu_{\mathrm{prod}}(g(B))$. Then we set $\sigma'(\beta)(A) \overset{\text{def}}{=} \mu(A_{\mathrm{dist}} \cap U) \sigma(f(\beta))(\{\text{edge } e \text{ chosen}\})$. This way σ' for M' simulates the continuous distributions in $[\![M]\!]$ s.t. measures on paths with σ and σ' agree [Hah13, Theorem 4.22]. This implies that reachability probabilities and reward values when using equivalent reward structures agree.

Because distributions in M' and $\mathrm{abs}(M, \mathscr{A})$ have finite support, one can define a finite automata simulation relation [SL95] such that $\langle l, v \rangle \preceq B_i$ if $\langle l, v \rangle \in B_i$ from which one concludes that $\mathrm{abs}(M, \mathscr{A})$ also bounds reachability probabilities of M'. Using extensions of simulation relations similar to e.g. [Hah13, Definition 7.26] one can also bound reward values in this way. □

4.3 Digital Clocks and Scaling Time

We model-check the resulting PTA using the existing digital clocks approach [NPS13]. Let us illustrate this approach on our running example:

Example 4 *The digital clocks MDP for the PTA from the previous example is shown in Figure 4. The clock-incrementing self-loops are labelled* `tick`*. We have excluded the non-stochastic part (locations* l_2 *and* l_4*) and merged the interval-valued variable x into the locations to show the concrete comparisons on the edges of* l_1^a *and* l_1^b*. We have also included the concrete probabilities for* $\lambda \approx 3$*. The maximum probability of reaching* l_3 *or* l_4 *in this MDP in at most* $t \in \mathbb{N}$ *time units is* 0.475 *for* $t = 0$ *and* 0.5 *for* $1 \leq t \leq 7$*. We know from Example 2 that the actual probability in the STA is* $\frac{1}{2} \cdot (1 - \mathrm{e}^{-\lambda t}) < 0.5$*. In our case of* $\lambda \approx 3$*, this is* 0 *for* $t = 0$*, approx.* 0.475 *for* $t = 1$ *and very close to* 0.5 *for* $t = 7$*. The error is thus between* 0.475 *and almost* 0 *depending on* t*.*

For reward r, the maximum value is ∞ *even if we remove the* `tick`*-edge from* l_0*: We can stay in* l_1^b *forever due to the right-open interval created for the unbounded exponential distribution. The minimum value computed in this MDP is* $0.475 \cdot 0 + 0.025 \cdot 1 = 0.025$*, whereas the actual value for* $\lambda \approx 3$ *is* $\approx \frac{1}{3}$*.*

The example shows that the error introduced by the abstraction of the continuous distributions depends on the variance of the distributions in relation to the interval width of at least 1 required to use PTA. In models where the dependence between time and property values is similarly direct as in this example, we can get more accurate results at the cost of larger MDP state spaces by *scaling time*: Both the results of the sampling and the non-interval values that clocks are compared to (including those in properties) are multiplied by some factor $d \in \mathbb{N}^+$. (For the

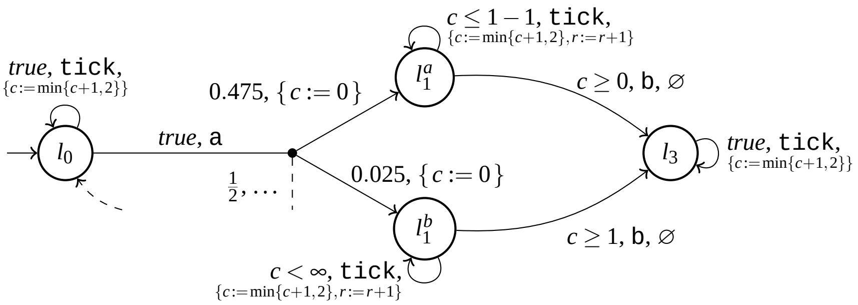


Figure 4: Digital clocks MDP of the PTA abstraction (explicit intervals)

exponential distribution, for example, the former can be achieved by dividing the rate by d.)

Example 5 *By scaling time by a factor of* $d = 2$ *in our running example STA, two unit width intervals are used for* $r = 0.05$ *and* $\lambda \approx 3$*, with probabilities* 0.388 *and* 0.087*. The upper bound for the reachability probability drops to* 0.388 *for* $t = 0$ *and* 0.475 *for* $t = 1$*; the lower bound for the minimum expected reward rises to* 0.137*.*

Although scaling time *can* lead to tighter bounds, there is another, independent cause of overapproximation error, which is due to the digital clocks requirement of closed clock constraints: All adjacent intervals have a singleton overlap, and we can only refer to exactly these overlapping values in clock constraints and properties. They have probability 0 in the STA, but not in the PTA, which leads to e.g. the upper bounds for time-bounded reachability probabilities being “one step ahead”: In Example 5, the upper bound computed for $t = 0$ is the actual probability for $t = 1$, the bound for $t = 1$ is the probability for $t = 2$, and so on.

5 Implementation

We have implemented our STA analysis approach in the new mcsta tool within the MODEST TOOLSET [HH14]. It relies neither on mcpta [HH09] nor on PRISM for PTA model checking. It currently supports the continuous uniform, exponential and normal distributions as follows, where x is a variable of type `real` and sampling expressions may reference other state variables:

– $x := \textsc{Uni}(\mathit{lower}, \mathit{upper})$ for the uniform distribution, where *lower* resp. *upper* are expressions of type `real` for which a concrete lower bound *lb* resp. a concrete upper bound $\mathit{ub} \in \mathbb{R}$ can be determined with $\mathit{lb} \leq \mathit{ub}$. The intervals are then $[\lfloor \mathit{lb} \rfloor, \lfloor \mathit{lb} \rfloor + 1], \ldots, [\lceil \mathit{ub} \rceil - 1, \lceil \mathit{ub} \rceil]$ with probability expressions constructed according to $\mathit{cdf}_{\mathrm{UNI}}(x) = (x - \mathit{lower})/(\mathit{upper} - \mathit{lower})$.

– $x := \mathit{offset} + \textsc{Exp}(\mathit{rate})$ for the exponential distribution, where *offset* is an expression of type `int` and *rate* is an expression of type `real` for which a concrete lower bound $\lambda \in \mathbb{R}^+$ can be determined. The intervals are then $[\mathit{offset}, \mathit{offset}+1], \ldots, [\mathit{offset}+n-1, \mathit{offset}+n]$ and $[\mathit{offset}+n, \infty)$ where $n = \lceil \frac{-\ln \rho}{\lambda} \rceil$ (using the quantile function of the exponential distribution). The probability expressions of the intervals are constructed according to $\mathit{cdf}_{\mathrm{EXP}}(x) = 1 - \mathrm{e}^{-\mathit{rate} \cdot x}$.

– $x := \textsc{Norm}(m, \sigma)$ for the normal distribution, where the mean m is an expression of type `int` and the standard deviation σ is a concrete value in $\mathbb{R}^+$. The intervals are $(-\infty, m-n], \ldots, [m-1, m]$, $[m, m+1], \ldots, [m+n, \infty)$. Since neither the quantile function nor the cdf of the normal

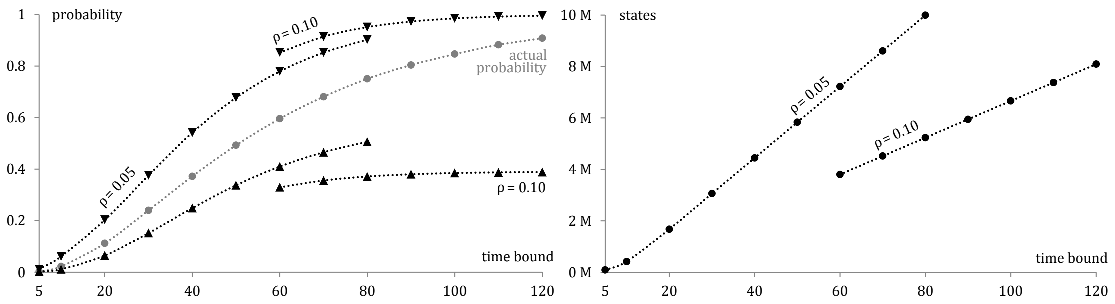


Figure 5: Reachability results and state space sizes for the M/G/1 example

distribution have a closed-form solution, we require σ to be a concrete value to precompute n and the actual interval probabilities based on σ and ρ close to `double` precision.

These examples show a general recipe to support other continuous distributions using their quantile function and cdf. In case a distribution is parameterised by an expression that contains state variables, we may generate more intervals than necessary for some valuations, which then have zero probability. For example, we generate two intervals for $x := \text{UNI}(0,2i)$ when i has domain $\{0,1\}$ since the upper bound of expression $2i$ is 2. However, since the probabilities are preserved as expressions, the probability of $[1,2]$ will evaluate to 0 for all states where $i \neq 2$.

6 Evaluation

We have applied mcsta to four different examples. We are interested in how close the computed bounds are to the actual values (effectiveness), and how large the state spaces of the underlying MDP become[1] (efficiency). All measurements were performed on the same 1.7 GHz Intel Core i5-3317U system with 4 GB of RAM running 64-bit Windows 8.1. The first two models we present are deterministic. As mentioned, our method is not targeted for this special case, so we expect correct and useful, but not very tight, computed bounds. Specialised methods will perform better or be more precise in these cases. The last two models, however, contain continuous and discrete nondeterminism, so our technique is currently the only one available for verification.

6.1 M/G/1 Queueing System with Normal Distribution

Our first example models an *M/G/1/6 queueing system* as STA where the service time is normally distributed with mean 10 and standard deviation 2. Since clocks cannot be negative, it is implicitly truncated to values ≥ 0 when we compare the result to a clock. The time between customer arrivals is exponentially distributed with rate $\frac{1}{6}$. The queue has length 5, not counting the customer being served, and is initially empty. We are interested in the following values:

- the probability p that the queue is full and $\leq t_p$ time units have elapsed,
- the expected time t until the queue is full for the first time, and
- the expected number c of customers served before the queue becomes full.

Since nondeterminism is absent by construction, we can use statistical model checking with the modes simulator from the MODEST TOOLSET to obtain good approximations of p, t and c.

[1] Memory was the limiting factor in all examples; runtime was always below 3 minutes.

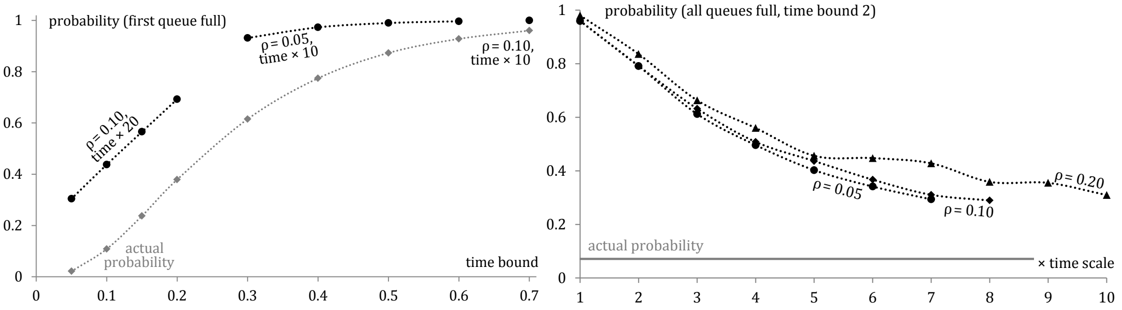


Figure 6: Reachability results and time scale effects for the tandem queues

The results of computing upper and lower bounds on p using mcsta are shown in Figure 5. On the left, we show the computed bounds for different values of t_p as black triangles. We see that there is a noticeable approximation error, but the general evolution of the probability over time is preserved. After $t_p \approx 80$, the lower bound shows no significant improvements. For $t_p \geq 90$, we ran out of memory, so we increased the residual probability parameter ρ to 0.1. The number of concrete states in the MDP of the digital clocks semantics is shown on the right of Figure 5. We see that it increases linearly with t_p and can be reduced significantly by increasing ρ, i.e. by lowering the number of intervals for the abstraction of the exponential and normal distributions.

Asking for *minimum* expected rewards, we compute bounds $t \geq 43.4$ and $c \geq 3.52$ for the other two values. As we do not need a global clock to check a time bound like t_p here, the underlying MDP has just 136 767 states. State-space exploration and computation of both bounds takes only 2.3 s in total. If we ask for *maximum* expected rewards, we get bounds ∞ due to the right-open intervals created by the abstraction of the unbounded distributions (cf. Example 4). Simulation with modes tells us that $t \approx 61$ and $c \approx 6.2$ for this deterministic model.

6.2 Tandem Queueing Network

We next look at a model from the PRISM benchmark suite [KNP12]: the *tandem queuing network* of an M/Cox$_2$/1/4 followed by an M/M/1/4 queue [HMS99]. It is a CTMC and we can thus model it as an STA without nondeterminism. We experiment with scaling time as described in Section 4.3. We compute the maximum probability p_{ff} of the first queue being full in time t, trying to use a value of $\rho \geq 0.05$ as low as possible and a time scaling factor as high as possible without running out of memory. The result is shown on the left of Figure 6.

The second property we look at is the maximum probability p_{af} of both queues becoming full within time t. This happens at a vastly different time scale: p_{af} only starts to approach 0.5 when t is on the order of 50. We thus focus on the effect of scaling time on the approximation error for fixed time bound $t = 2$. The results are shown on the right of Figure 6. We see that the error can be significantly reduced by scaling up time.

Finally, we compute bounds on the expected times t_{ff} until the first queue becomes full and t_{af} until both are full. As we increase the time scaling, we go from lower bounds $t_{ff} \geq 0.000012$ and $t_{af} \geq 0.56$ for time scale $d = 1$ with 9 557 MDP states, computed in 0.1 s, to $t_{ff} \geq 0.108710$ and $t_{af} \geq 5.87$ for $d = 10$ with 3 662 958 states, computed in 108 s. Again, upper bounds (i.e. maximum expected rewards) are all ∞. From simulation, we get $t_{ff} \approx 0.29$ and $t_{af} \approx 17.9$.

Table 1: Results and comparison for the WLAN example

model	type	P_{max}	$[E^{\wedge}_{min}, E^{\wedge}_{max}]$	$[E^{\vee}_{min}, E^{\vee}_{max}]$	$[E^{1}_{min}, E^{1}_{max}]$	states	time
wlan	PTA	0.18359	$[1325, 6280]$ µs	$[450, 4206]$ µs	$[450, 5586]$ µs	104 804	8 s
wlan-uni	STA	0.13659	$[2325, 4607]$ µs	$[950, 3018]$ µs	$[950, 3880]$ µs	264 240	15 s

6.3 Wireless LAN with Uniform Transmission Time

Departing from queueing systems, we now look at the model of a communication protocol: the carrier-sense multiple-access with collision avoidance (CSMA/CA) part of IEEE 802.11 *WLAN*. We take the MODEST PTA model [HH09] and replace the nondeterministic choice of transmission delay out of $[200, 1250]$ µs (with a unit of time representing 50 µs) by a uniformly distributed choice over the same interval. The result is still nondeterministic, and an STA instead of a PTA.

Model-checking results for the original PTA ("wlan") and the new STA ("wlan-uni") are shown in Table 1. We see that the state space of the underlying MDP is larger when the uniform distribution is used. This is because the states not only contain explicit values for all clocks as in the original PTA, but additionally 21 different concrete intervals that overapproximate the result of sampling from $\text{UNI}(4, 25)$. The blowup thus stays far below the worst-case factor of 21.

We analyse six time-unbounded properties: P_{max}, the maximum probability that either of the two modelled senders' backoff counters reaches the upper bound of 2, as well as $E^{\wedge}_{min}/E^{\wedge}_{max}$, $E^{\vee}_{min}/E^{\vee}_{max}$ and E^{1}_{min}/E^{1}_{max}, the minimum/maximum expected times until both senders, either of them, or the one with id 1 correctly deliver their packets. Due to the nondeterminism, we cannot use simulation or any other technology to compute the actual values. However, the computed bounds are plausible if we assume that in the PTA, the longest/shortest transmission delay maximises/minimises the values. The STA is thus indeed expected to show less extremal behaviour.

6.4 File Server

As a final example, we analyse another model that combines all essential features of STA and cannot be model checked with any other approach we know of (except prohver). It represents a single-threaded *file server* with slow archival storage:

- Requests arrive to a single queue of length $C = 5$ with interarrival times following $\text{EXP}(\frac{1}{8})$.
- File sizes are uniformly distributed over some range such that sending the file back to a client takes time uniformly distributed over $[1, 3]$.
- 2 % of all files are in slow archival storage. Retrieving a file is instantaneous for normal storage, but takes between 30 and 40 time units nondeterministically for archival storage.

We thus have continuous stochastic delays, a probabilistic choice and nondeterministic delays. Additionally, we model the initial queue length as uniformly distributed in $\{0, \ldots, \lfloor \frac{C}{2} \rfloor\}$. The model is part of the MODEST TOOLSET download.

We are interested in the probability p that the request queue becomes full within time t_p, and the minimum (i.e. worst-case) expected time t until this happens. For t, we obtain a lower bound of 462 time units from an MDP with 107 742 states in 6 s. For p, the results are shown in Figure 7. On the right, we see that the number of MDP states again grows linearly with the time bound.

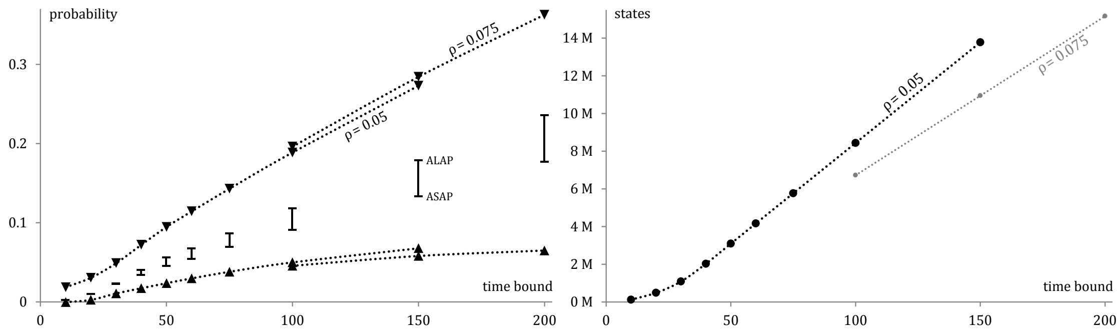


Figure 7: Model checking results and state space sizes for the file server example

On the left, we have plotted the computed upper/lower bounds using small triangles.

Due to the nondeterministic delay, we cannot use simulation. However, we can instruct modes to resolve that delay by scheduling events either as soon or as late as possible (ASAP/ALAP). Simulating these deterministic variants of the model gives us $t \approx 1012$ for ASAP and $t \approx 721$ for ALAP. For p, the simulation results are included on the right of Figure 7. The results that we get via our new approach are clearly useful: They are safe bounds whereas we do not know anything about the relationship between simulation results and the actual values.

7 Conclusion

We presented the first fully-automated model checking approach for STA with general, unbounded distributions and support for nondeterminism. It provides upper bounds for maximum and lower bounds for minimum reachability probabilities and expected rewards. We investigated causes of approximation error and showed that scaling time can effectively reduce the error. In experiments performed with our implementation, mcsta, we saw that the approach works well in practice, but state-space explosion is a significant problem for time-bounded properties.

References

[BBH+13] P. Ballarini, N. Bertrand, A. Horváth, M. Paolieri, E. Vicario. Transient Analysis of Networks of Stochastic Timed Automata Using Stochastic State Classes. In *QEST*. LNCS 8054, pp. 355–371. Springer, 2013.

[BBJM12] P. Bouyer, T. Brihaye, M. Jurdzinski, Q. Menet. Almost-Sure Model-Checking of Reactive Timed Automata. In *QEST*. Pp. 138–147. IEEE Computer Society, 2012.

[BD04] M. Bravetti, P. R. D'Argenio. Tutte le Algebre Insieme: Concepts, Discussions and Relations of Stochastic Process Algebras with General Distributions. In *Validation of Stochastic Systems*. LNCS 2925, pp. 44–88. Springer, 2004.

[BDHK06] H. C. Bohnenkamp, P. R. D'Argenio, H. Hermanns, J.-P. Katoen. MoDeST: A Compositional Modeling Formalism for Hard and Softly Timed Systems. *IEEE Trans. Software Eng.* 32(10):812–830, 2006.

[BG02] M. Bravetti, R. Gorrieri. The theory of interactive generalized semi-Markov processes. *Theor. Comput. Sci.* 282(1):5–32, 2002.

[DK05] P. R. D'Argenio, J.-P. Katoen. A theory of stochastic systems part I: Stochastic automata. *Inf. Comput.* 203(1):1–38, 2005.

[DLL$^+$11] A. David, K. G. Larsen, A. Legay, M. Mikucionis, Z. Wang. Time for Statistical Model Checking of Real-Time Systems. In *CAV*. LNCS 6806, pp. 349–355. Springer, 2011.

[FHH$^+$11] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, L. Zhang. Measurability and safety verification for stochastic hybrid systems. In *HSCC*. Pp. 43–52. ACM, 2011.

[Hah13] E. M. Hahn. *Model checking stochastic hybrid systems*. PhD thesis, Universität des Saarlandes, 2013.

[HH09] A. Hartmanns, H. Hermanns. A Modest Approach to Checking Probabilistic Timed Automata. In *QEST*. Pp. 187–196. IEEE Computer Society, 2009.

[HH14] A. Hartmanns, H. Hermanns. The Modest Toolset: An Integrated Environment for Quantitative Modelling and Verification. In *TACAS*. LNCS 8413, pp. 593–598. Springer, 2014.

[HHHK13] E. M. Hahn, A. Hartmanns, H. Hermanns, J.-P. Katoen. A compositional modelling and analysis framework for stochastic hybrid systems. *Formal Methods in System Design* 43(2):191–232, 2013.

[HMS99] H. Hermanns, J. Meyer-Kayser, M. Siegle. Multi Terminal Binary Decision Diagrams to Represent and Analyse Continuous Time Markov Chains. In *NSMC*. Pp. 188–207. Prensas Universitarias de Zaragoza, 1999.

[HS00] P. G. Harrison, B. Strulo. SPADES - a process algebra for discrete event simulation. *J. Log. Comput.* 10(1):3–42, 2000.

[KNP12] M. Z. Kwiatkowska, G. Norman, D. Parker. The PRISM Benchmark Suite. In *QEST*. Pp. 203–204. IEEE Computer Society, 2012.

[KNSS00] M. Z. Kwiatkowska, G. Norman, R. Segala, J. Sproston. Verifying Quantitative Properties of Continuous Probabilistic Timed Automata. In *CONCUR*. LNCS 1877, pp. 123–137. Springer, 2000.

[LHK01] G. G. I. López, H. Hermanns, J.-P. Katoen. Beyond Memoryless Distributions: Model Checking Semi-Markov Chains. In *PAPM-PROBMIV*. LNCS 2165, pp. 57–70. Springer, 2001.

[NPS13] G. Norman, D. Parker, J. Sproston. Model checking for probabilistic timed automata. *Formal Methods in System Design* 43(2):164–190, 2013.

[SL95] R. Segala, N. A. Lynch. Probabilistic Simulations for Probabilistic Processes. *Nord. J. Comput.* 2(2):250–273, 1995.