

Uniformization with Representatives

Comprehensive Transient Analysis of Infinite-State QBDs

Anne Remke, Boudewijn R. Haverkort, Lucia Cloth
University of Twente
The Netherlands

[anne, brh, lucia]@cs.utwente.nl

ABSTRACT

A large variety of computer and communication systems can be modeled adequately as infinite-state continuous-time Markov chains (CTMCs). A highly structured class of such infinite-state CTMCs is the class of Quasi-Birth-Death processes (QBDs), on which we focus in this paper. We present an efficient variant of uniformization for computing the transient probability $\mathbf{V}_{i,j}(t)$ of being in each state j of the QBD for any possible initial state i at time t . Note that both the set of starting states and the set of goal states have infinite size. All the probabilities $\mathbf{V}_{i,j}(t)$ are needed in the context of probabilistic model checking. The key idea of our algorithm is to split the infinite set of starting states into a finite part and an infinite (repeating) part. The transient probabilities for the infinite part are then computed using the new notion of *representatives*. We present the required data structures and algorithm, as well as an application-dependent optimal stopping criterion. In a simple case study we show the feasibility of our approach, as well as the efficiency gain due to the optimal stopping criterion.

Categories and Subject Descriptors

G.3 [Probability and Statistics]: Stochastic processes, Markov processes; C.4 [Performance of Systems]: Reliability, availability, and serviceability

General Terms

Algorithms, Performance

1. INTRODUCTION

Quasi-Birth-Death processes (QBDs) [11] are often used in the context of performance models of systems with very large or, seen from a performance perspective, infinitely large buffers [12]. The transient evolution of such QBDs is of interest, e.g., in temporary situations of overload (for which a steady-state distribution might not exist), in situations where the time the system is to be observed is too

short to reach steady state, or in case one is explicitly interested in the transient behavior toward steady state (think of “slow start” in TCP).

The need to determine transient probabilities in QBDs is also described in [13], which presents an overall model-checking approach for QBDs, including a uniformization-based static approach to calculate the transient state probability of being in each state of the QBD for any possible initial state. In this paper, we enhance that static algorithm to a new iterative (dynamic) algorithm, called *uniformization with representatives*, that iteratively computes ever more accurate values in each step in a memory-efficient way. Our algorithm only uses as many iterations as necessary to match a given error bound or to decide whether the transient probabilities meet a given probability bound.

The use of our approach is twofold. First, for CSL model checking we need the transient probabilities for *any possible initial state*. Second, comprehensive transient analysis can be used to calculate transient probabilities for QBDs if the initial state probability vector possibly has positive entries for all states. This can be done by considering all states as starting states and combining the transient probabilities with the initial distribution afterwards. The algorithm presented in this paper is the first one that facilitates these needs.

The rest of this paper is further organized as follows. We introduce QBDs in Section 2. We then describe how to apply uniformization to QBDs in general in Section 3. Section 4 describes the repetitive structure of the infinite matrices needed for uniformization of QBDs and shows how they can be cut to a finite representation. The iterative algorithm for uniformization with representatives is then presented in Section 5. Two application-dependent stopping criteria are discussed in Section 6. We present tool support in Section 7. In Section 8 we illustrate by means of a case study that uniformization with representatives is practically feasible. We discuss related work in Section 9 and conclude in Section 10.

2. QUASI-BIRTH-DEATH-PROCESSES

A QBD is a CTMC with a structured infinite state space. The infinite state space of a QBD can be viewed as a two-dimensional strip, which is finite in one dimension and infinite in the other. The states in this strip can be grouped in so-called *levels*, according to their identity in the infinite dimension. Figure 1 gives a graphical representation of a QBD. Transitions, represented by positive entries in the generator matrix \mathbf{Q} , can only occur between states of the same

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SMCtools'06, October 10, 2006, Pisa, Italy

Copyright 2006 ACM 1-59593-506-1/06/10 ...\$5.00

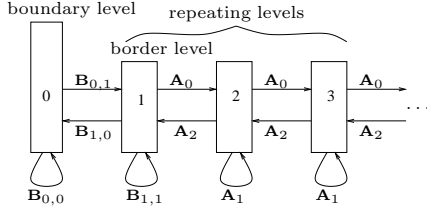


Figure 1: Quasi-Birth-Death Process

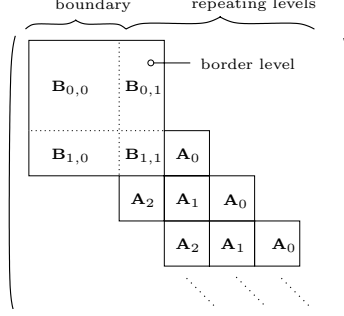


Figure 2: Block-tridiagonal structure of the generator matrix \mathbf{Q}

level or between states of neighboring levels. All repeating levels have the same inter-level and intra-level transition structure and contain corresponding states. The requirements for a QBD are summarized in the following definition.

Definition 1

A **labeled QBD** \mathcal{Q} of order (N_0, N) (with $N_0, N \in \mathbb{N}^+$) is a labeled infinite-state continuous-time Markov chain. The set of states can be partitioned into an infinite number of finite sets $S^j, j = \{0, 1, \dots\}$, each containing the states of one level, such that $S = \bigcup_{j=0}^{\infty} S^j = \{0, \dots, N_0 - 1\} \times \{0\} \cup \{0, \dots, N - 1\} \times \mathbb{N}^+$, where the first part represents the boundary level with N_0 states, and the second part the infinite number of repeating levels, each with N states. The block-tridiagonal generator matrix \mathbf{Q} consists of the following finite matrices describing the inter- and intra-level transitions:

- $\mathbf{B}_{0,0} \in \mathbb{R}^{N_0 \times N_0}$: intra-level transitions, boundary level,
- $\mathbf{B}_{0,1} \in \mathbb{R}^{N_0 \times N}$: inter-level transitions: boundary to border level,
- $\mathbf{B}_{1,0} \in \mathbb{R}^{N \times N_0}$: inter-level transitions: border to boundary level,
- $\mathbf{B}_{1,1} \in \mathbb{R}^{N \times N}$: intra-level transitions, border level,
- $\mathbf{A}_0 \in \mathbb{R}^{N \times N}$: inter-level transitions: repeating to next higher repeating level,
- $\mathbf{A}_1 \in \mathbb{R}^{N \times N}$: intra-level transitions, repeating levels,
- $\mathbf{A}_2 \in \mathbb{R}^{N \times N}$: inter-level transitions: repeating to next lower repeating level.

\mathcal{Q} , being an infinite-state CTMC, describes a stochastic process $\{X_t \in S \mid t \geq 0\}$. \square

Although QBDs are introduced here at the state level, high-level specification formalisms, e.g., based on stochastic Petri nets [12] or stochastic process algebra [3, 10], do exist.

The states of each level S^i are divided into three, not

necessarily disjoint, sets of states: $S^i = S_{in}^{i,\uparrow} \cup S_{center}^{i,\uparrow} \cup S_{out}^{i,\uparrow}$: $S_{in}^{i,\uparrow}$ comprises the states that can be reached from the next lower level in one step, $S_{center}^{i,\uparrow}$ comprises the states from which no other level can be reached in one step, and $S_{out}^{i,\uparrow}$ comprises the states from which the next higher level can be reached in one step. Similarly, the sets $S^i = S_{in}^{i,\downarrow}, S_{center}^{i,\downarrow}$ and $S_{out}^{i,\downarrow}$ are defined.

$g(s_1, s_2) = |\text{shortestpath}(s_1, s_2)|$ gives the minimum number of steps that has to be undertaken to reach s_2 from $s_1, s_1, s_2 \in S$.

Let $d^\uparrow \geq 1$ be the so-called *upward level diameter*, that is, the minimum number of state transitions that is needed to reach the next higher repeating level from a state in $S_{in}^{i,\uparrow}$:

$$d^\uparrow = \min\{g(s_1, s_2) \mid s_1 \in S_{in}^{i,\uparrow}, s_2 \in S_{in}^{i+1,\uparrow}\}$$

The *downward level diameter* d^\downarrow is defined along the same lines as

$$d^\downarrow = \min\{g(s_1, s_2) \mid s_1 \in S_{in}^{i,\downarrow}, s_2 \in S_{in}^{i-1,\downarrow}\}.$$

We define d , the *symmetric level diameter*, as the minimum of the upward and downward level diameter. As the repeating levels of a QBD all exhibit the same structure, they all have the same level diameter.

Notice, however, that the number of steps needed to cross l levels may be larger than $l \cdot d$, depending on the structure of the QBD levels. Thus, one can introduce a *cross-level diameter* d_i , giving the number of steps that is needed to cross i subsequent levels. However, we do not further use this in our algorithm and implementation so far.

3. UNIFORMIZATION ON QBDS

As motivated in the introduction, we want to compute the transient probability of being in each state of the infinite-state QBD for any possible initial state. Applying uniformization to QBDS, we have to deal with a system of differential equations of infinite size. An important reason why uniformization can be used on QBDS is that the matrix \mathbf{Q} has only a finite number of different diagonal entries (originating from the matrices $\mathbf{B}_{0,0}, \mathbf{B}_{1,1}$, and \mathbf{A}_1), so that the uniformization rate λ can be determined even though \mathbf{Q} has an infinite number of entries. \mathbf{P} then follows the same block-tridiagonal structure as \mathbf{Q} , where the submatrices are replaced by $\hat{\mathbf{B}}_{0,0}, \hat{\mathbf{B}}_{0,1}, \hat{\mathbf{B}}_{1,0}, \hat{\mathbf{B}}_{1,1}, \hat{\mathbf{A}}_0, \hat{\mathbf{A}}_1$ and $\hat{\mathbf{A}}_2$, respectively. Together, they form the probability matrix \mathbf{P} of the corresponding DTMC and are calculated as follows:

$$\hat{\mathbf{B}}_{i,j} = \begin{cases} \mathbf{I} + \frac{\mathbf{B}_{i,j}}{\lambda}, & i = j, \\ \frac{\mathbf{B}_{i,j}}{\lambda}, & i \neq j. \end{cases} \quad \hat{\mathbf{A}}_i = \begin{cases} \mathbf{I} + \frac{\mathbf{A}_i}{\lambda}, & i = 1, \\ \frac{\mathbf{A}_i}{\lambda}, & i \neq 1. \end{cases}$$

\mathbf{P} is a stochastic matrix because all entries are between 0 and 1 and the rows sum up to 1, and describes a DTMC.

Let $\mathbf{U}^{(k)}$ be the state probability distribution matrix after k epochs in the DTMC with transition matrix \mathbf{P} . It can be derived recursively as:

$$\mathbf{U}^{(0)} = \mathbf{I}, \quad \text{and} \quad \mathbf{U}^{(k)} = \mathbf{U}^{(k-1)} \mathbf{P}, \quad k \in \mathbb{N}^+. \quad (1)$$

Let $\mathbf{V}(t)$ be the matrix of transient state probabilities at time t , that can be calculated:

$$\mathbf{V}(t) = \sum_{k=0}^{\infty} \psi(\lambda t; k) \mathbf{P}^k = \sum_{k=0}^{\infty} \psi(\lambda t; k) \mathbf{U}^{(k)}, \quad (2)$$

where the Poisson probabilities $\psi(\lambda t; k)$ express the probability of k events occurring in the interval $[0, t]$ in a Poisson process with rate λ . The probability distribution in the DTMC after k steps is described by $\mathbf{V}(0)\mathbf{P}^k$ (note that $\mathbf{V}(0) = \mathbf{I}$). Hence, by the law of total probability the transient probability matrix $\mathbf{V}(t)$ is obtained as the weighted sum $\sum_{k=0}^{\infty} \psi(\lambda t; k)\mathbf{P}^k$ over all possible numbers of steps.

Recall that matrices $\mathbf{V}(t)$ and $\mathbf{U}^{(k)}$, $k \in \mathbb{N}$, have infinite size. However, we can give a finite representation as will be described in Section 4.

To avoid the infinite summation over the number of steps k , the sum (2) needs to be truncated. We denote the maximum error that possibly occurs in an entry of $\mathbf{V}(t)$ when the series is truncated after n steps as $\varepsilon_t^{(n)}$. We have

$$\varepsilon_t^{(n)} = \left\| \sum_{k=n+1}^{\infty} \psi(\lambda t; k)\mathbf{U}^{(k)} \right\| \leq 1 - \sum_{k=0}^n e^{-\lambda t} \frac{(\lambda t)^k}{k!}. \quad (3)$$

Note that n can be computed *a priori*, given $\varepsilon_t^{(n)}$, λ and t , as the smallest n that satisfies

$$\sum_{k=0}^n \frac{(\lambda t)^k}{k!} \geq \frac{1 - \varepsilon_t^{(n)}}{e^{-\lambda t}} = (1 - \varepsilon_t^{(n)})e^{\lambda t}.$$

We denote the approximation of $\mathbf{V}(t)$ that has been calculated with up to n terms with $\mathbf{V}^{(n)}(t)$:

$$\mathbf{V}^{(n)}(t) = \sum_{k=0}^n \psi(\lambda t; k)\mathbf{U}^{(k)}. \quad (4)$$

We can compute $\mathbf{V}^{(n+1)}(t)$ as:

$$\mathbf{V}^{(n+1)}(t) = \mathbf{V}^{(n)}(t) + \psi(\lambda t; n+1)\mathbf{U}^{(n+1)}. \quad (5)$$

The matrix $\mathbf{U}^{(n)}$ is multiplied by the probability of exactly n events occurring in the time interval $[0, t]$. Since $\mathbf{V}^{(n)}(t)$ remains zero wherever $\mathbf{U}^{(n)}$ has zero entries, $\mathbf{V}^{(n)}(t)$ clearly follows the structure of $\mathbf{U}^{(n)}$. Thus, we can cut the infinite matrix $\mathbf{V}^{(n)}(t)$ to a finite representation as well where the size depends on the highest iteration index n and the level diameter d . $\mathbf{V}^{(n)}(t)$ has the structure of $\mathbf{U}^{(n)}$.

4. REPETITIVE STRUCTURE

The matrices $\mathbf{V}^{(n)}(t)$ and $\mathbf{U}^{(n)}$ are of infinite size. However, exploiting the repetitive structure of QBDs and the truncation given by uniformization we can give a finite representation that depends on the number of considered steps n for a given error bound.

In Section 4.1 we present the basic idea of this finite representation. We explain this in detail for $\mathbf{U}^{(n)}$ in Section 4.2. In Section 4.3 we discuss how the level diameter can be used to keep the considered parts of the infinite matrices as small as possible.

4.1 Finite representation

From every single state, only a finite number of states is reachable in n steps. The transient probability to get to one of the non reachable states is zero. Hence, for a single starting state it is sufficient to consider the finite set of reachable states instead of the whole infinite state space. This idea has already been mentioned in [4, 9, 17]. However, when simultaneously considering every state of the infinite state space as starting state, one would have to consider an infinite number of finite parts of the QBD, which is not feasible.

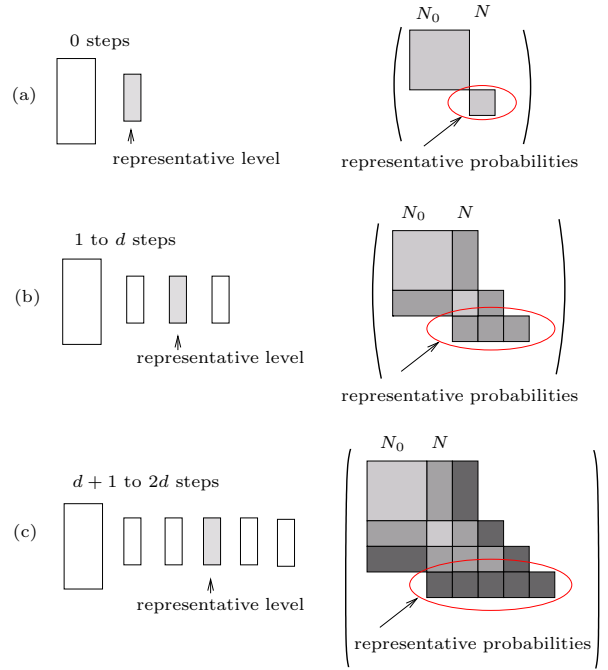


Figure 3: Considered part of the state space (left) and finite representation of $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$ (right), depending on the number of considered steps

However, given a finite number of steps n , there is a repeating level l from which onwards the boundary level cannot be reached anymore. Therefore the finite part of the QBD that has to be considered for starting states from repeating levels $\geq l$ does not contain states of the boundary level. The structure of all these finite parts is identical, only shifted appropriately. This implies that we obtain identical transient probabilities (shifting appropriately) for corresponding states in repeating levels $\geq l$, within the error bounds of uniformization within n steps. Therefore we can use the states of level l as *representatives* for all corresponding states of higher levels. In fact, we restrict the computation to a finite number of starting states and still perform a comprehensive transient analysis considering every state as starting state. This new approach is only possible because of the repetitive structure of QBDs.

For a finite representation of the matrices $\mathbf{V}^{(n)}(t)$ and $\mathbf{U}^{(n)}$, it is now sufficient to store all non-zero entries for starting states of levels $\leq l$. The entries of level l then account for corresponding states of the remaining infinite part of the QBD. The size of the finite representation depends on the considered number of steps n , hence, on the time, the uniformization rate, and the required accuracy.

4.2 Probability distribution after n epochs

We now address the growth of the matrices $\mathbf{U}^{(n)}$ in the course of the uniformization. Figure 3(a) shows that the dimension of the finite representation of $\mathbf{U}^{(0)}$ is $(N_0 + N) \times (N_0 + N)$. Since we consider zero steps, we cannot leave a level and the first repeating level is already representative for all (other) repeating levels.

Allowing for one step, we can reach the next higher or the

next lower level. Since the next lower level is the boundary level, the first repeating level cannot be used as representative. However, we can use the second repeating level as representative, as shown in Figure 3(b). As we consider only one step, it is possible to reach the next higher level as well; thus we have to consider starting in one of the first three levels (including the boundary level) and ending up in one of the first four levels. The dimension of $\mathbf{U}^{(1)}$ is therefore given as $(N_0 + 2N) \times (N_0 + 3N)$. With a symmetric level diameter d , we will need at least another $d - 1$ steps before possibly reaching the next higher repeating level. Thus, the size of all $\mathbf{U}^{(n)}$, for $n = 1, \dots, d$, will be the same as for $n = 1$.

Figure 3(c) shows the finite representation of matrices $\mathbf{U}^{(n)}$ for $n = d + 1, \dots, 2d$. From a given level, we can reach at most the next two higher or lower levels. Therefore, we have to pick a new representative: the third repeating level. Starting from this representative, we can reach the next two higher repeating levels. We have to attach another row (of blocks) for the new representative, and in every other row we have to attach one block to the left (the next lower) and one to the right (the next higher level), wherever possible. The dimension is then $(N_0 + 3N) \times (N_0 + 5N)$, for all $\mathbf{U}^{(n)}$, for $n = d + 1, \dots, 2d$.

In general for a given number of steps $n \geq 1$ and level diameter d , the number l of levels reachable from a representative level in one direction (up or down) is given by

$$l = ((n - 1) \operatorname{div} d) + 1 \quad (6)$$

The size of the matrix $\mathbf{U}^{(n)}$ is then determined by this value of l . Its dimension is $(N_0 + (l + 1)N) \times (N_0 + (2l + 1)N)$. In practice this means that whenever we have to pick a new representative, we attach one block subdiagonal to the left and one to the right, and a new block row is added to the lower end of the matrix. All blocks that are situated to the left or to the right of the multiple block diagonal only have entries equal to zero.

The last block row in the finite representation is always the representative. From the representative we can conclude the probability values for all block rows that are not explicitly included in the finite representation. They all consist of the same blocks as the representative, but the position of the blocks is shifted by one block to the right.

4.3 Exploitation of the level diameter

Recall that we defined an upward level diameter d^\uparrow and a downward level diameter d^\downarrow in Section 2. If the upward and the downward level diameters differ considerably from each other, it makes sense to exploit both level diameters instead of the symmetric one. For example, if the upward level diameter is much smaller than the downward level diameter, we can, with a growing number of steps, reach more repeating levels to the right than repeating levels to the left. When considering the symmetric level diameter, we would include repeating levels between the boundary level and the representative which actually could not be reached. Therefore, we propose to check in the n -th iteration step, whether $(n - 1) \bmod d^\uparrow = 0$ or $(n - 1) \bmod d^\downarrow = 0$. In the first case, an additional higher repeating level can be reached from the representative, and in the second case the boundary level becomes reachable from the representative. The considered portion of the QBD has to be adapted as shown in Figure 4. As can be seen in Figure 4(a), for 1 to d^\uparrow steps, we can only

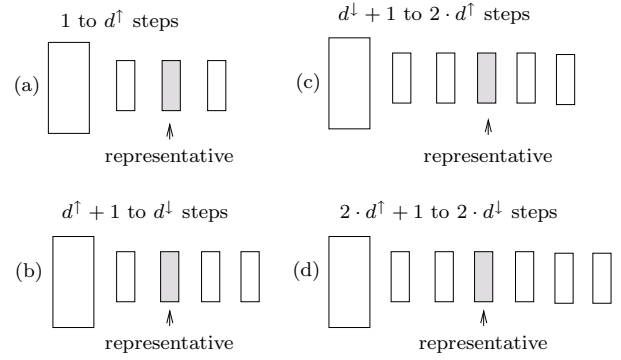


Figure 4: Considered part of the state space when exploiting the upward and downward level diameter under the assumption that $d^\uparrow < d^\downarrow$

reach the next lower or the next higher repeating level. In Figure 4(b), we depict the case where two higher repeating levels can be reached, but just one lower repeating level is reachable. Figures 4(c) and 4(d) show how this iteration evolves for an higher number of steps.

For ease of notation, we stick to considering the symmetric level diameter throughout the rest of this paper.

5. UNIFORMIZATION WITH REPRESENTATIVES

Having explained the structure and dimension of the finite representation, we will present the actual computation of $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$ in this section. The matrix $\mathbf{V}^{(n)}(t)$ is computed iteratively according to Equation (5); in the $(n + 1)$ th iteration step, one has to compute $\mathbf{U}^{(n+1)}$. Starting with $n = 0$, and thus with a small finite portion of the QBD, cf. Figure 3(a), we increase n in each iteration. With growing n and thus increasing accuracy, the considered finite representation of the QBD grows. However, in each iteration we always use the smallest possible representation.

In the following section we describe matrix data-structures for the iterative method to compute $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$. These matrices have a block structure, according to the levels of a QBD, as discussed in Section 4. We denote the blocks that give the probabilities from states in level i to states in level j as $\mathbf{U}_{i,j}^{(n)}$ and $\mathbf{V}_{i,j}(t)$, respectively. Row index i represents all levels that are considered as starting levels, and column index j denotes all levels that are considered as possible end levels.

5.1 Probability distribution after n epochs

In every step of (4), we need the state probability matrix $\mathbf{U}^{(n)}$, which is calculated recursively using (1). Starting with $\mathbf{U}^{(0)}$ (with dimension $(N_0 + N) \times (N_0 + N)$), the calculation of $\mathbf{U}^{(1)}$ is visualized in Figure 5: we multiply the finite representation of $\mathbf{U}^{(0)}$, where one row of blocks is added for the new representative repeating level, with a finite portion of \mathbf{P} that consists of three block rows, for the three considered starting levels, and of four block columns, for the four levels that can be reached.

In general, for $n \leq 1$, $\mathbf{U}^{(n)}$ is computed as $\mathbf{U}^{(n-1)} \cdot \mathbf{P}$, according to Equation (1). Due to the finite representations

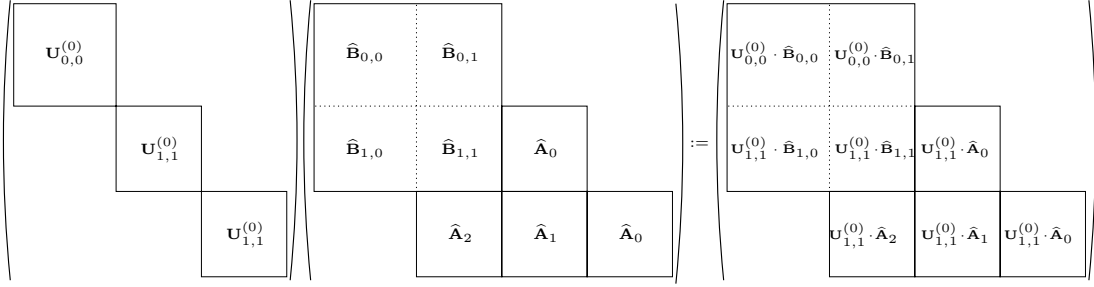


Figure 5: Computation of $\mathbf{U}^{(0)} \cdot \mathbf{P} = \mathbf{U}^{(1)}$

of $\mathbf{U}^{(n)}$ for $n = 1, \dots, d$, $\mathbf{U}^{(n)}$ is now computed as follows:

$$\begin{aligned}
 \mathbf{U}_{i,0}^{(n)} &= \mathbf{U}_{i,0}^{(n-1)} \cdot \widehat{\mathbf{B}}_{0,0} + \mathbf{U}_{i,1}^{(n-1)} \cdot \widehat{\mathbf{B}}_{1,0}, \\
 &\text{for } i = 0, \dots, l, \\
 \mathbf{U}_{i,1}^{(n)} &= \mathbf{U}_{i,0}^{(n-1)} \cdot \widehat{\mathbf{B}}_{0,1} + \mathbf{U}_{i,1}^{(n-1)} \cdot \widehat{\mathbf{B}}_{1,1} + \mathbf{U}_{i,2}^{(n-1)} \cdot \widehat{\mathbf{A}}_2, \\
 &\text{for } i = 0, \dots, l+1, \\
 \mathbf{U}_{i,j}^{(n)} &= \mathbf{U}_{i,j-1}^{(n-1)} \cdot \widehat{\mathbf{A}}_0 + \mathbf{U}_{i,j}^{(n-1)} \cdot \widehat{\mathbf{A}}_1 + \mathbf{U}_{i,j+1}^{(n-1)} \cdot \widehat{\mathbf{A}}_2, \\
 &\text{for } i = 0, \dots, l+1, \quad j = 2, \dots, i+l, \quad (7)
 \end{aligned}$$

where l , the number of levels reachable in one direction is computed as in (6). Note that for extending the finite representation no extra computations are required. All blocks that are not included by the indices i, j and are left or right of the included blocks are zero matrices. Note that [9] use equations that pretty much resemble (7), see Section 9 for a discussion.

5.2 Approximation $\mathbf{V}^{(n)}(t)$

Recall that $\mathbf{V}^{(n)}(t)$ is the approximation of $\mathbf{V}(t)$ that has been calculated for up to n steps and its finite representation has the same dimension as $\mathbf{U}^{(n)}$. Due to the block structure of $\mathbf{V}^{(n)}(t)$, we can rewrite (4) as:

$$\mathbf{V}_{i,j}^{(n)}(t) = \mathbf{V}_{i,j}^{(n-1)}(t) + \psi(\lambda t; n) \mathbf{U}_{i,j}^{(n)}, \quad (8)$$

for $i = 0, \dots, l+1$ and for $j = \max\{0, i-l\}, \dots, i+l$.

After d steps, the size of $\mathbf{V}^{(n)}(t)$ will change. Therefore, we will possibly include matrices $\mathbf{V}_{i,j}^{(n-1)}(t)$ in the calculation of (8) that have not been addressed before. These matrices are either located left or right of the block-structured diagonal. In that case, these matrices are zero. We also have to attach one more row of blocks. The entries of these blocks have the same entries as the blocks in the representative. However, the position of the blocks is shifted to the right by one block.

6. WHEN TO STOP?

We can consider transient probabilities in two different contexts. Either we are interested in transient probabilities as a performance measure on its own, or we can use transient probabilities for model checking QBDs. We consider stopping criteria for these two cases in the following, after which we discuss complexity.

6.1 A priori stopping criterion

Recall that uniformization requires a finite number of steps n to be taken into account in order to compute the transient

probabilities for an allowed numerical error $\varepsilon_t^{(n)}$. The number of levels l reachable from the representative level in one direction when allowing up to n steps is computed by (6). In order to get the state-to-state transient probabilities for all states of the QBD, it is sufficient to compute them for levels $0, \dots, l+1$ only. Hence, only a finite part of the infinite QBD is needed.

After having completed n steps with uniformization with representatives, we considered the finite portion of the QBD as shown in Figure 6. Level $l+1$ is the representative, as outgoing from this level the boundary level cannot be reached anymore with n steps. Moving up from the representative,

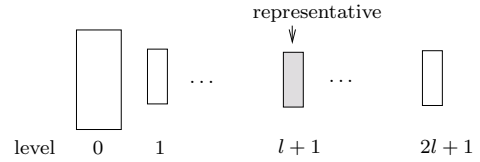


Figure 6: Finite fraction of the QBD needed for the transient solution within a given error bound.

we still have to consider l levels “to the right”. The highest level we have to consider is therefore $2l+1$ (cf. Figure 6). Thus, for a given accuracy, we reduced the task of computing transient probabilities for an infinite QBD to the computation of transient probabilities in a finite CTMC.

6.2 Dynamic stopping criterion

For model checking QBDs, we do not need the exact probabilities; it is sufficient to know whether the probability of reaching a given set of goal states meets a bound p .

A CSL *until* formula ϕ requires the calculation of transient probabilities [13, 1]. However, it is always wrapped into the so-called probability operator $\mathcal{P}_{\triangleright p}(\phi)$. Thus, we do not need the exact probabilities, but we want to decide whether the probability for ϕ meets the bound p . Applying the a priori stopping criterion when model checking QBDs introduces several problems. First, we might incur an unnecessary amount of computation, because we might be able to decide the outcome after fewer iterations. Second, there are several settings at which the accuracy is not sufficient to decide whether the probability meets the bound.

To overcome both problems, we propose a dynamic stopping criterion, which we claim to be optimal in the setting of model checking QBDs.

Whenever a model checking algorithm relies on the calculation of transient probabilities at time t , we have to com-

pare for each starting state the probability of ending in one of a given set of goal states $G \subseteq S$ with probability bound p . The probability to end up in a state in G , given that we started at a state s , is denoted as $\gamma_s(t)$. Hence, we have to decide for all $s \in S$ whether $\gamma_s(t)$ meets the bound p . Along the lines of uniformization with representatives we introduce an approximation of $\gamma_s(t)$ after n iterations as:

$$\gamma^{(n)}(t) = \mathbf{V}^{(n)}(t) \cdot \gamma(0).$$

Note that the vector $\gamma^{(n)}(t)$ follows the repetitive structure of the QBD, just as the matrices $\mathbf{V}^{(n)}(t)$ and $\mathbf{U}^{(n)}$. The vector can be split into several levels, where the first level constitutes the boundary level and the following levels constitute the appropriate repeating levels. In principle, $\gamma^{(n)}(t)$ is infinite, but we can cut it to a finite representation, as from a representative repeating level on, all levels contain the same values.

During the iteration over the number of considered steps n , the entries of $\gamma^{(n)}(t)$ increase in a monotonic way. Thus, comparing entries of the current probability vector $\gamma^{(n)}(t)$ with the bound p on a regular basis, we might be able to decide whether the probability meets the bound p after a smaller number of iterations than computed a priori for the static stopping criterion (cf. Section 6.1).

With uniformization with representatives, the computed approximation after n steps is always an underestimation of the actual probability. Fortunately, we are able to indicate the maximum possible error as follows. Recall that $\varepsilon_t^{(n)}$ is the maximum error after n iteration steps (cf. Equation (3)). As we consider a goal set G with $|G|$ states, the overall maximum error is given by $\varepsilon_t^{(n)} \cdot |G|$, such that

$$\gamma_s(t) \leq \gamma_s^{(n)}(t) + \varepsilon_t^{(n)} \cdot |G|.$$

From (3) follows clearly, that the value of $\varepsilon_t^{(n)}$ decreases as n increases. Exploiting the above inequality, we obtain the following stopping criteria:

- (a) $\gamma_s^{(n)}(t) \geq p \Rightarrow \gamma_s(t) \geq p$
- (b) $\gamma_s^{(n)}(t) \leq p - \varepsilon_t^{(n)} \cdot |G| \Rightarrow \gamma_s(t) \leq p$

These criteria can be exploited as follows. Starting with a small number of steps, we check whether for the current approximation one of the inequalities (a) or (b) holds for all starting states. If this is not the case we continue the computation, check again, etc. until one of the stopping criteria holds. Once uniformization has stopped, we can interpret the outcome as to whether a given starting state meets the bound p .

However, if for one of the starting states $s \in S$ we have $\gamma_s(t) = p$, the iteration never stops, as neither of the stopping criteria ever holds.

6.3 Complexity

The level index l_k of the representative increases with the number of considered steps k and decreases with the symmetric level diameter d . In the k th iteration, we actually consider the states of the boundary level and of $l_k + 1$ repeating levels as starting states and the states of the boundary level and of $2l_k + 1$ repeating levels as end states. The boundary level has N_0 states and each repeating level has N states, resulting in matrices with $(N_0 + (l_k + 1)N) \cdot (N_0 + (2l_k + 1)N)$ entries, so that the storage requirement grows with the level index of the representative.

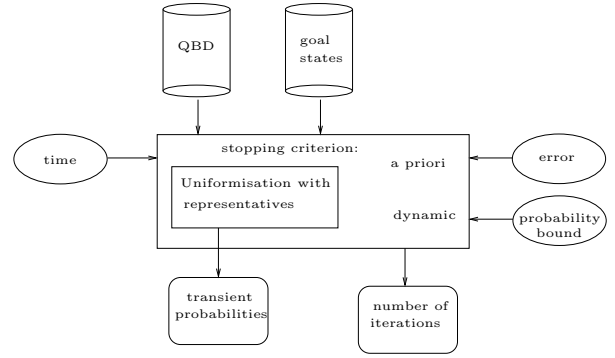


Figure 7: Sketch of the tool architecture

If n is the maximum number of steps considered, the overall storage complexity for the three probability matrices $\mathbf{U}^{(n-1)}$, $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ is $\mathcal{O}(3(N_0^2 + l_n N_0 N + l_n^2 N^2))$.

Let ν denote the average number of transitions originating from a single state in the QBD. Assuming a sparse representation, the discrete transition matrix \mathbf{P} has storage complexity $\mathcal{O}(\nu(N_0 + 2N))$.

In the k th iteration, the multiplication of matrix $\mathbf{U}^{(n-1)}$ with \mathbf{P} is carried out in $\mathcal{O}(\nu(N_0 + (l_k + 1)N))$. For n the maximum number of considered steps, the overall time complexity therefore equals $\mathcal{O}(n \cdot \nu(N_0 + (\frac{l_n}{2} + 1)N))$.

7. TOOL SUPPORT

We have developed a software tool to compute the transient probabilities of being in a, possibly infinite, set of goal states, for every possible initial state of a QBD. As input we require a QBD and a set of goal states. The QBD is specified through six files containing the transitions per block matrix (and by a file containing the number of states in the boundary level and in the repeating levels). These files are generated from a specification of an infinite-state stochastic Petri net.

The set of goal states can be finite (being just an enumeration of states), can be of infinite size (involving some regular pattern of states of the repeating levels), or can be a combination of both; these goal states are also specified in a file. Figure 7 shows a sketch of the tool architecture.

To perform the transient analysis, we need to specify the time point t and the allowed maximum error ϵ . Both can be specified via a graphical user interface. With this interface it is also possible to choose between the a priori and the dynamic stopping criterion. In case the dynamic stopping criterion is chosen, we have to specify the probability bound with which the computed probability is compared as additional parameter.

As output we obtain the transient probabilities for all starting states to reach the set of goal states and the number of steps needed for the computation, as well as the number of steps needed to calculate the transient probabilities depending on the chosen stopping criterion. The output is restricted to those states for which these transient probabilities are non-zero, given the preset error tolerance ϵ .

8. CASE STUDY: CONNECTION MANAGEMENT

We analyze the transient behavior of a connection management mechanism, known as “on-demand connection with delayed release” (OCDR) [7], thereby using the technique developed in this paper. In Section 8.1 we describe the OCDR mechanism in detail, in Section 8.2 we derive a QBD model for this mechanism. Finally, we derive two different measures in Section 8.3.

8.1 System Description

The transport protocol TCP offers a connection-oriented service in the Internet, which implies that a connection has to be established prior to any application data being exchanged [8]. For applications with a connectionless nature, such as e-mail or web browsing, prior to application data transfer, a connection needs to be established. Arriving application-level packets therefore potentially suffer a delay from the connection establishment at the TCP layer boundary, unless an existing connection can be (re)used. Once a connection has been established, all application-level packets can be transported and the connection can be released immediately afterwards, or after some delay (time-out). The latter is exactly the mechanism that is being used for HTTP 1.1; its predecessor, HTTP 1.0, did not allow for connection reuse.

We analyze the OCDR mechanism as sketched in Figure 8. Packets (often called ‘segments’ in TCP context) that have to be transported are generated by an abstract *packet generator* and submitted to the transfer queue. The *connection* can be in one of two modes: (i) it can be *active*, so that an arriving packet can be served immediately, at the cost of maintaining an possibly unused connection; (ii) the connection can be *released*, so that an arriving packet can only be transmitted after the connection is re-established, but there are no costs for maintaining an unused connection. Arriving packets at a released connection suffer an extra *connection-establishment delay*. Once active, all queued packets, as well as those arriving, can be transported. The connection is released after having been unused in active mode for some predefined time-out period.

In this specific application, packets are assumed to arrive in bursts, that is, the packet generator cycles through periods in which packets are generated with high intensity, followed by periods in which no packets are generated at all. The connection management decides and controls when and how to switch between modes, so as to find the right balance between good performance (low delays) and low costs. Having served the last packet of a burst, the connection will be held active for some time, waiting for a new burst of packets to arrive. If such a burst does not start within some time-out period, the OCDR mechanism decides to release the connection.

8.2 QBD Model

To keep the model simple and illustrative, we assume an exponentially distributed connection-establishment delay with rate c , as well as an exponentially distributed time-out for release with mean $1/r$. The model could be extended easily to more deterministic delays, e.g., by using Erlangian approximations [12]. Packets take an exponentially distributed amount of time to be transmitted, with rate μ . In a burst, packets are generated according to a Poisson pro-

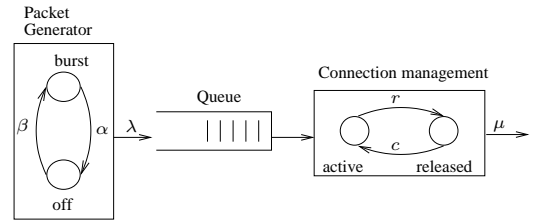


Figure 8: An abstract model of the OCDR mechanism for connection management

cess with rate λ . The generator switches between epochs of activity and idleness, both exponentially distributed, with rates α and β , respectively. Under these conditions, Figure 9 provides the corresponding QBD. In this model the state space is $S = \{(i, j, k) \mid i \in \mathbb{N}, j, k = 0, 1\}$, where i denotes the number of packets queued (and being transmitted), j denotes whether the connection is *active* ($j = 1$) or *released* ($j = 0$), and k denotes whether the packet arrival process is in a burst ($k = 1$) or not ($k = 0$). Clearly, each level i ($= i$ tasks present) consists of four states: $S^i = \{(i, 0, 0), (i, 0, 1), (i, 1, 0), (i, 1, 1)\}$. The symmetric level diameter is $d = 1$. Table 1 shows the numerical values of the parameters as presented in [7].

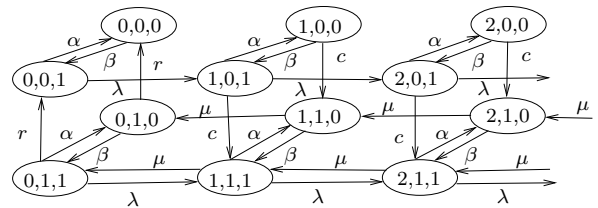


Figure 9: QBD of the system model

Table 1: Numerical values for the parameters of the model

parameter	sec^{-1}
λ	100
μ	125
α	1
β	0.04
c	10
r	10

8.3 Analysis

We consider two different transient measures of the OCDR mechanism. First, we evaluate the probability to reach one of the states in the boundary level (level 0) within a given period of time. For this measure, we make the four goal states $(0, j, k)$ in the boundary level absorbing, since we are interested in *reaching* them from any of the other states. The number of goal states is 4.

The second measure we want to compute is the probability of reaching a state, where no packets are generated ($k = 0$) and the connection is released ($j = 0$). These are states $(i, 0, 0)$ for all $i \geq 0$, which results in an infinite number of goal states. We first compute the above measures with the

a priori stopping criterion in Section 8.3.1, before we apply the dynamic stopping criterion in Section 8.3.2. For both measures we choose as time instance $t = 0.5$ seconds.

8.3.1 A priori stopping criterion

Table 2 shows the a priori computed number of steps n needed for a given accuracy $\varepsilon_t^{(n)}$. Figure 10 gives the probabilities of reaching a state in the boundary level within $t = 0.5$ seconds for $\varepsilon_t^{(n)} = 10^{-6}$, starting in any state of the considered part of the QBD. For the states of the boundary level itself, these probabilities are all 1. With growing distance from the boundary level, the probabilities drop and reach values very close to zero ($< 10^{-20}$) at level 150 already. For level 256 and all higher levels, the probabilities are 0 (for the given accuracy).

In Figure 11 we present the probabilities of reaching a state $(j, 0, 0)$ before $t = 0.5$. For these states themselves, the probability is always 1 and we omit the corresponding curve. For state $(0, 0, 1)$ (no packet present, connection released, packets are generated), the probability of reaching one of the goal states is slightly larger than 0.1. For the first repeating level the probability for $(i, 0, 1)$ jumps to almost 0.3 and then decreases and stagnates at about 0.09. The explanation for this is as follows: to reach one of the goal states $(j, 0, 0)$ from $(i, 0, 1)$ in a repeating level, one can either switch to one of the states $(i, 0, 0)$ without leaving *released* mode or one has to go “through” the boundary level 0 before. The first is done with constant probability 0.09, the second with a probability declining with growing distance from the boundary level. The curve for states $(i, 1, 1)$ (connection established, generate packets) also shows a jump at the beginning and then follows the one for reaching the boundary level, because one always has to go through a state in the boundary level in order to reach a goal state. Among the states $(i, 1, 0)$ (connection established, no packets), the one in the boundary level $(0, 1, 0)$ is the only one from which a goal state is directly reachable and it is highly probable that it actually switches to this state. For the following repeating levels, the probability declines with growing distance from the boundary level and settles down to zero.

The curves show that the number of iterations needed for a reasonable accuracy (e.g., $\varepsilon_t^{(n)} = 10^{-6}$ for 256 steps), according to Table 2, might actually be too large, as the values for the computed probabilities already reach a for much smaller level numbers. Note that, since, for $d = 1$ the number of steps n equals the level-index of the right-most reachable level. This behavior depends on the size and structure of the QBD. Hence, with the a priori stopping criterion, we compute too much. This is exactly where the dynamic stopping criterion turns out to be more efficient.

Table 2: Number of steps required for given error bound $\varepsilon_t^{(n)}$

$\varepsilon_t^{(n)}$	# steps n
10^{-1}	169
10^{-2}	193
10^{-3}	212
10^{-4}	226
10^{-5}	243
10^{-6}	256

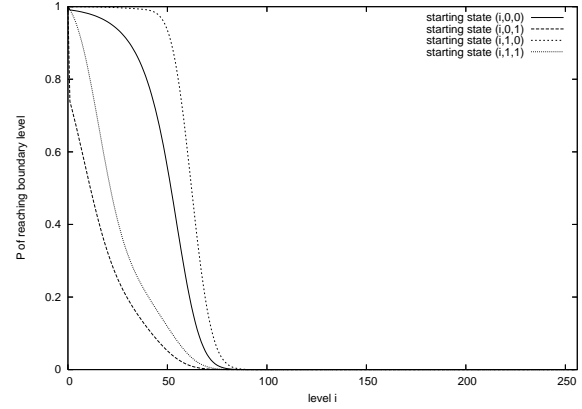


Figure 10: Transient probabilities of reaching a state in the boundary level from each of the states in level i within $t = 0.5$ sec, for $i = 0, \dots, 256$

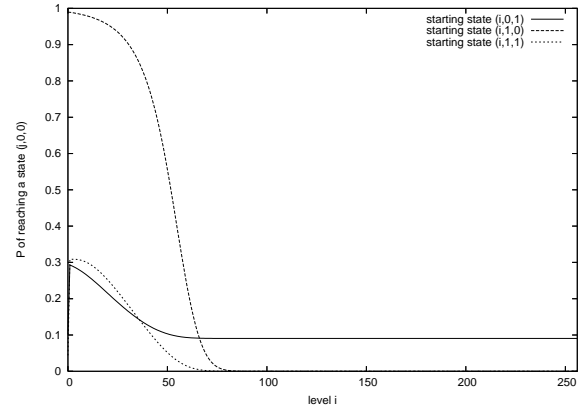


Figure 11: Transient probabilities of reaching one of the states $(j, 0, 0)$ for $j \geq 0$ from each of the states in level i within $t = 0.5$ sec, for $i = 0, \dots, 256$

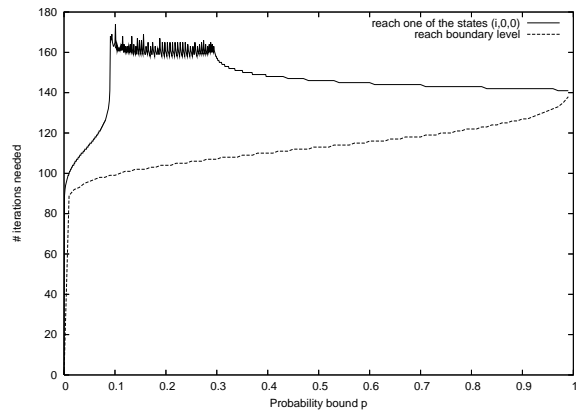


Figure 12: Number of iterations needed to compare the probabilities of reaching a state $(i, 0, 0)$ within $t = 0.5$ sec with a given probability bound

8.3.2 Dynamic stopping criterion

Using the dynamic stopping criterion, as presented in Section 6.2, Figure 12 shows the number of uniformization steps needed for the computation of the two measures, depending on the probability bound p . The stopping criterion is adjusted to work with $\geq p$. On the x -axis we record the probability bound p . The y -axis shows the actual number of iterations needed to decide for each state whether the transient probability of reaching one of the goals states is below or above the probability bound p .

After 0 steps the comparison can be evaluated for $p = 0$ for both measures: every probability is ≥ 0 .

Then both curves show a jump to around 90 iterations. This is because the Poisson probabilities are de facto equal to zero in the first few iterations and no decision can be made when comparing with $p > 0$. For the first measure (reaching the boundary level), the number of iterations needed grows slightly with p , but stays far below any of the a priori computed numbers of iterations as given in Table 2. The curve for the second measure (reaching a state $(i, 0, 0)$) is less regular. It shows several peaks at the beginning and then declines slightly. A peak occurs every time when one of the computed probabilities gets really close to the probability bound p we have to compare with. But even these peaks are far below the precomputed numbers of iterations from Table 2. Note that in case a probability that is computed actually would agree with the probability bound p , the dynamic algorithm would not stop. But this is unlikely for “smooth” probability bounds p , because the Poisson probabilities introduce “quite crude” numbers.

9. RELATED WORK

We are aware of a number of other approaches toward the computation of transient-state probabilities for infinite-state CTMCs. We briefly refer to these here, and indicate the difference to our approach.

In [20] the infinitely-sized Kolmogorov differential equations describing the transient state probabilities of the QBD are transformed into the Laplace domain. The solution to this differential equation system is then obtained (in the Laplace domain) via the solution of an infinite system of linear equations. Due to the QBD structure, a matrix-geometric solution can be found for this equation system. The “only” remaining problem is the required back - transformation to the time domain. How to do that efficiently, is not described in the paper. It is certainly of interest to investigate the applicability of new methods for Laplace transform inversions in the context of finite-state CTMCs [6] in our, infinite-state, context. In any case, this approach does only consider transient-state probabilities for a single starting state.

A variety of uniformization variants has been developed over the last 20 years, for increased efficiency for particular applications or parameter settings. For instance, Van Moorsel and Sanders proposed adaptive uniformization for stiff CTMCs [17, 18, 19], and Van Moorsel proposed orthogonal uniformization for acyclic CTMCs [17]. Van Moorsel also hints at an approach called dynamic uniformization [17], and so does Grassmann [4], as a technique to evaluate systems with infinite state spaces. In their well-known 1984 paper [5], Gross and Miller already refer to a possible use of uniformization for infinite state systems; how to do this, is

not discussed though. In the last three works, only hints toward evaluating transient-state probabilities in infinite-state systems are given; no true algorithms or data structures have been, nor is the issue of having an infinite number of possible starting states discussed.

With step-wise uniformization [2] it is possible to calculate transient-state probabilities in large or even infinite-size CTMCs and DTMCs. This is done by step-wisely extending the considered state space, i.e., on-the-fly while generating the state space, until enough steps, given the time point considered, are taken into account. Clearly, this requires a unique starting state, and results in transient-state probabilities just for that starting state.

Recent work by Van Houdt and Blondia [15, 16] also addresses the transient analysis of QBDs. First of all, this work has been set up for discrete-time (but can be extended for continuous time). Secondly, it has been tailored to very specific measures, being the state probability distribution at time n (being a discrete time identifier) or the waiting time of the n -th customer, all given a single fixed starting state, i.e., the empty queue. Furthermore, the analysis is approximate, in that the deterministic time-point of interest is brought into the model (and so enlarges the state space) using a negative binomially distributed random variable. By increasing the number of phases in this random variable, its variability decreases, hence, this random variable approximates the deterministic time value n ever-more precisely. The advantage of this approach is that algorithms for steady-state analysis can be used, albeit at the cost of being approximate. Clearly, this approach differs from our approach substantially: we address continuous time, and compute all transient state probabilities for all states as starting states in an exact way.

In [9], Le Ny and Sericola present a transient analysis of the continuous time BMAP/PH/1 queue, also using uniformization. The paper focuses on the computation of the queue length distribution at time t , i.e., $\Pr\{QL(t) \leq i\}$, as well as the busy time period distribution, starting from an initial probability distribution. Their recurrence relations for the queue length [9, Equ. 3, 4, 5] and the busy period [9, Equ. 10, 11] strongly resemble our recursive calculation of the probability distribution after k epochs (cf., Equ. 7) and the approximation matrix of the transient probabilities (cf., Equ. 5.2) This is due to the fact that both approaches exploit the QBD structure as well as the finite time horizon. However, their method is tailored towards two specific measures, and takes a single fixed starting state, whereas we compute all transient state probabilities for all starting states at the same time, by introducing the notion of *representatives*.

Finally, in [14] Rubino and Sericola propose a method to compute the interval availability in finite-state Markovian models, given an initial state probability distribution. Their method also works in case of infinite state spaces and a tri-diagonal structure of the Markov chain, but this is not described in detail (only mentioned in the conclusions). However, the algorithm presented in this paper does not compute what we want to compute, that is, the transient-state probabilities for all states in a QBD, for all starting states. In conclusion, the algorithms for transient analysis of QBDs as available in or hinted at in the literature, do not compute all transient-state probabilities for all starting states, as our algorithm does.

10. SUMMARY AND CONCLUSIONS

Uniformization with representatives, as presented in this paper, is an efficient adaptation of state-to-state uniformization, and has been developed specifically to compute transient probabilities of QBDs. Exploiting the highly repetitive structure of QBDs, it is possible to use finite representations of the involved infinite-sized matrices. Due to the special structure of QBDs, it is possible to compute these matrices iteratively without computational overhead. In contrast to the well-known a priori stopping criterion we present another stopping criterion that we claim to be optimal, if comparing a transient probability to a given bound p , as is the typical thing to do in CSL model checking. Finally, in a small case study, we have shown our approach to be feasible.

We are the first to actually compute transient probabilities on a wide class of infinite state CTMCs, considering the complete infinite state space as starting states and as end states. Uniformization with representatives is both computationally and memory efficient. The only drawback the approach suffers from is that the number of considered steps can become large, as n depends on the product λt ; but this is a general drawback of uniformization. Note that this is a proto-type implementation and that speed improvements might be achieved by using banded matrices.

We intend to apply the idea of uniformization with representatives to other highly structured classes of infinite-state CTMCs as well. We finally note, that the optimal stopping criterion can also be used for efficient model checking of finite CTMCs.

11. ACKNOWLEDGMENTS

We thank Benny van Houdt for valuable discussions on the topic. Anne Remke is supported through the MC=MC project (612.000.311) and Lucia Cloth through the MOQS project (642.000.505), both financed by the Netherlands Organization for Scientific Research (NWO).

12. REFERENCES

- [1] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, July 2003.
- [2] G. Ciardo. Discrete-time Markovian stochastic Petri nets. In Kluwer, editor, *Computations with Markov Chains*, pages 339–358. Raleigh, 1995.
- [3] A. El-Rayes, M. Kwiatkowska, and G. Norman. Solving infinite stochastic process algebra models through matrix-geometric methods. In *Proc. 7th Process Algebras and Performance Modelling Workshop (PAPM'99)*, pages 41–62. University of Zaragoza, 1999.
- [4] W. Grassmann. Finding transient solutions in Markovian event systems through randomization. In *Numerical solution of Markov chains*, pages 411–420. Dekker, 1991.
- [5] D. Gross and D. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.
- [6] P. Harrison and W. Knottenbelt. Passage Time Distributions in Large Markov Models. In *Proceedings ACM Sigmetrics 2002*, pages 77–85, 2002.
- [7] G. Heijenk and B. Haverkort. Designing and evaluation of a connection management mechanism for an ATM-based connectionless service. *Distributed System Engineering Journal*, 3(1):53–67, 1996.
- [8] J. Kurose and K. Ross. *Computer Networking*. Addison Wesley, 2005.
- [9] L. Le Ny and B. Sericola. Transient analysis of the BMAP/PH/1 queue. *I.J. of Simulation*, 3(3-4):4–15, 2003.
- [10] I. Mitrani, A. Ost, and M. Rettelbach. TIPP and the spectral expansion method. In F. Baccelli, A. Jean-Marie, and I. Mitrani, editors, *Quantitative Models in Parallel Systems*, pages 99–113. Springer, 1995.
- [11] M. Neuts. *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Johns Hopkins University Press, 1981.
- [12] A. Ost. *Performance of Communication Systems. A Model-Based Approach with Matrix-Geometric Methods*. PhD thesis, Dept. of Computer Science, RWTH Aachen, 2001.
- [13] A. Remke, B. Haverkort, and L. Cloth. Model checking infinite-state Markov chains. In *Proc. 11th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'05), LNCS 3440*, pages 237–252. Springer, 2005.
- [14] G. Rubino and B. Sericola. Interval Availability Analysis Using Denumerable Markov Processes: Application to Multiprocessor Subject to Breakdowns and Repair. *IEEE Transactions on Computers*, 44(2):286–291, 1995.
- [15] B. Van Houdt and C. Blondia. Approximated transient queue length and waiting time distributions via steady-state analysis. *Stochastic Models*, 21:725–744, 2005.
- [16] B. Van Houdt and C. Blondia. QBDs with marked time epochs: a framework for transient performance measures. In *Proc. 2nd International Conference on the Quantitative Evaluation of Systems*, pages 210–219. IEEE press, 2005.
- [17] A. van Moorsel. *Performability Evaluation Concepts and Techniques*. PhD thesis, Dept. of Computer Science, University Twente, 1993.
- [18] A. van Moorsel and W. Sanders. Adaptive Uniformization. *ORSA Communications in Statistics: Stochastic Models*, 10(3):619–648, 1994.
- [19] A. van Moorsel and W. Sanders. Transient Solution of Markov Models by Combining Adaptive & Standard Uniformization. *IEEE Transactions on Reliability*, 46(3):430–440, 1997.
- [20] J. Zhang and E. Coyle. Transient analysis of quasi-birth-death processes. *Stochastic Models*, 5(3):459–496, 1989.