

A uniformization-based algorithm for model checking the CSL until operator on labeled queueing networks ^{*}

Anne Remke and Boudewijn R. Haverkort

University of Twente
Design and Analysis of Communication Systems
Faculty for Electrical Engineering, Mathematics and Computer Science
[anne,brh]@cs.utwente.nl

Abstract. We present a model checking procedure for the CSL until operator on the CTMCs that underly Jackson queueing networks. The key issue lies in the fact that the underlying CTMC is infinite in as many dimensions as there are queues in the JQN. We need to compute the transient state probabilities for all goal states and for all possible starting states. However, for these transient probabilities no computational procedures are readily available. The contribution of this paper is the proposal of a new uniformization-based approach to compute the transient state probabilities. Furthermore, we show how the highly structured state space of JQNs allows us to compute the possible infinite satisfaction set for until formulas. A case study on an e-business site shows the feasibility of our approach.

1 Introduction

Jackson queueing networks (JQNs) [9] are widely used to model and analyze the performance of computer and communication systems. Recently, we have presented detailed CSL model checking algorithms for QBDs [13] and in [12] (at Formats 2007) we presented a general approach to model check JQNs against continuous-stochastic logic (CSL)[1], [2]; note, however, that [12] only presented the principles of such an approach. This paper proposes an efficient iterative algorithm for the computation of the transient probabilities for any possible initial state on JQNs, which allows for model checking the CSL until operator on JQNs; the proposed algorithm generalizes our previous work for QBDs [13]. We elaborate on the form and the growth of the data structures, as needed for JQNs, that make our algorithm memory efficient. Furthermore, we explain that our algorithm is computationally efficient, as it uses just as many iterations as necessary to decide whether the transient probabilities meet a given probability bound.

^{*} The work presented in this paper has been performed in the context of the MC=MC project (612.000.311), financed by the Netherlands Organization for Scientific Research (NWO). The authors thank Lucia Cloth for fruitful discussions on the topic.

Related work on transient analysis on queueing networks is mostly restricted to finite state spaces. For instance, Harrison [8] presents an iterative method to solve the time-dependent Kolmogorov equations of finite queueing networks. In [4] Buchholz applies uniformization to hierarchical queueing networks that have a finite structured state-space. We are not aware about any approach that tackles the problem we are solving with our new algorithm.

This paper is organized as follows: In Section 2 we introduce Jackson queueing networks and the partitioning of the underlying infinite state space. The general approach for model checking the time-bounded until operator on JQNs is discussed in Section 3, before we present the details of the uniformization-based algorithm on JQNs in Section 4. Section 5 shows how the presented algorithm can be used to facilitate model checking the CSL until operator. As a case study, we model an e-business site as JQN and analyze its scalability with the newly developed model checking techniques in Section 6 before we conclude in Section 7.

2 Jackson queueing networks

In the following we recapitulate some of the foundations needed for CSL model checking of Jackson queueing networks [9], as presented in [12]. A labeled JQN consists of a number of interconnected queues and is defined as follows:

Definition 1 (Labeled Jackson queueing network)

A labeled Jackson queueing network **JQN J** of order M (with $M \in \mathbb{N}^+$) is a tuple $(\lambda, \underline{\mu}, \mathbf{R}, L)$ with arrival rate λ , a vector of size M of service rates $\underline{\mu}$, a routing matrix $\mathbf{R} \in \mathbb{R}^{(M+1) \times (M+1)}$ and a labeling function L that assigns a set of valid atomic propositions from a fixed and finite set AP of atomic propositions to each state $\mathbf{s} = (s_1, s_2, \dots, s_M)$. \square

The underlying state space of a JQN **J** of order M is a highly-structured *labeled infinite state continuous-time Markov chain*, \mathcal{J} , with state space $\mathbf{S} = \mathbb{N}^M$, that is infinite in M dimensions. Every state $\mathbf{s} \in \mathbf{S}$ is represented as an M -tuple $\mathbf{s} = (s_1, s_2, \dots, s_M)$ and denoted the number of customers per queue. To deal with the underlying infinite state space we presented a partition into an infinite number of disjoint finite sets [12], such that with one step only the next higher or the next lower partition of the state space can be reached. A given corner point¹ \bar{v} partitions the state space into rectangular shaped fronts, that are pairwise disjoint and situated like shells around each other. In an M dimensional JQN the front $F(\bar{v} + \bar{\mathbf{1}})$ is a finite set of states defined by

$$F(\bar{v} + \bar{\mathbf{1}}) = \{\mathbf{s} \in \mathbf{S} \mid \exists m (s_m = v_m + i) \wedge (\forall n \neq m (s_n \leq v_n + i))\}, \quad (1)$$

¹ denoted splitting vector in [12]

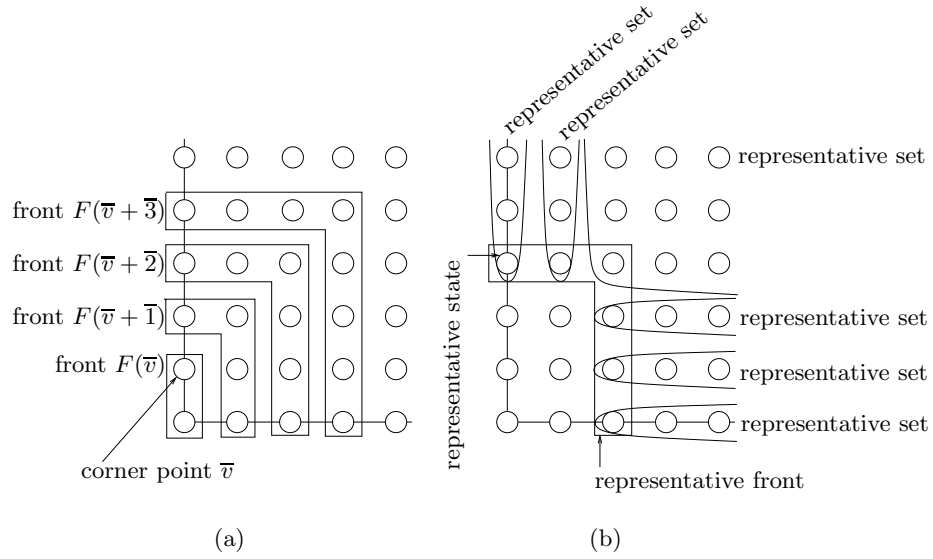


Fig. 1. (a) Partitioning of the underlying state space (b) Representative front, state and set according to Definition 2.

with $m \in \{1, \dots, M\}$. The number of states per front in a partitioning with corner point \bar{v} equals:

$$|F(\bar{v})| = \prod_{m=1}^M (v_m + 1) - \prod_{m=1}^M v_m. \quad (2)$$

In the following we introduce the concept of representative fronts, states and sets for JQNs, as visualized in Figure 1. A CSL state formula Φ is *independent* as of \bar{g} if the validity of Φ remains the same in corresponding states in fronts $F(\bar{g} + \bar{i})$ for $i \geq 0$.

Definition 2 (Representative front, state and set)

For a JQN and a CSL formula that is independent as of \bar{g} the notion of representatives is defined as follows: The front $F(\bar{g})$ is called *representative front* and denoted as $R(\bar{g})$. The states in the representative front are called *representative states* $r \in R(\bar{g})$. Each representative state r represents a distinct infinite set of states, denoted S_r , such that for all $r \in R(\bar{g})$ and for all $s \in S_r$ it holds that $r \models \Phi \Leftrightarrow s \models \Phi$. In general, in an M -dimensional JQN, there are M types of representative sets that account for 1 up to M infinite dimensions. A representative set S_r is called infinite in dimension m if and only if $r_m = g_m$, and restricted in dimension m otherwise. In case a representative state r equals \bar{g} in k dimensions it represents a k -dimensional set S_r , such that

$$s \in S_r \Leftrightarrow \begin{cases} s_m \geq r_m, & \text{iff } r_m = g_m, \\ s_m = r_m, & \text{otherwise.} \end{cases} \quad \square$$

Hence, a state \mathbf{s} belongs to S_r when it takes the same value as \mathbf{r} in the restricted dimensions and any value at least r_i in the infinite dimensions.

3 Time-bounded until operator

Recall that the CSL path formula $\Phi \mathcal{U}^I \Psi$ is valid if a Ψ -state is reached on a path during the time interval I via only Φ -states. As shown in [2], for model checking the time-bounded until operator with a time interval of the form $I = [0, t]$, the future behavior of the JQN is irrelevant for the validity of φ , as soon as a Ψ -state is reached. Thus all Ψ -states can be made absorbing without affecting the satisfaction set of formula φ . On the other hand, as soon as a $(\neg\Phi \wedge \neg\Psi)$ -state is reached, φ will be invalid, regardless of the future evolution. As a result of the above consideration, we may switch from checking the underlying CTMC \mathcal{J} to checking a new, derived, Markov chain denoted as $\mathcal{J}[\Psi][\neg\Phi \wedge \neg\Psi] = \mathcal{J}[\neg\Phi \vee \Psi]$, where all states in the underlying Markov chain that satisfy the formula in square brackets are made absorbing. Model checking a formula involving the until operator then reduces to calculating the transient probabilities $\pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\mathbf{s}, \mathbf{s}', t)$ for all Ψ -states \mathbf{s}' . Exploiting the partitioning of the underlying state space yields:

$$\begin{aligned} \mathbf{s} \models \mathcal{P}_{\bowtie p}(\Phi \mathcal{U}^{[0,t]} \Psi) &\Leftrightarrow \text{Prob}^{\mathcal{J}}(\mathbf{s}, \Phi \mathcal{U}^{[0,t]} \Psi) \bowtie p \\ &\Leftrightarrow \left(\sum_{i=0}^{\infty} \sum_{\mathbf{s}' \in \text{Sat}^{F(i)}(\Psi)} \pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\mathbf{s}, \mathbf{s}', t) \right) \bowtie p. \end{aligned} \quad (3)$$

The transient probabilities are accumulated for the Ψ states in fronts $F(i)$ for $i \in \mathbb{N}$. The transient probability of being in each state of the infinite-state JQN for any possible initial state can be calculated with a new iterative uniformization-based method, which we present in the Section 4.

4 Uniformization with Representatives

We recapitulate the basic idea of uniformization with representatives as introduced in [13], before we present the details of applying uniformization with representatives to JQNs.

4.1 Uniformization

Uniformization is a well-known technique to compute the transient probabilities $\mathbf{V}(t)$ in a CTMC [7]. As standard property of uniformization, the finite time bound t is transformed to a finite number of steps n . The probability matrix $\mathbf{P}(\mathbf{s}, \mathbf{s}')$ for the uniformized DTMC is defined as

$$\mathbf{P}(\mathbf{s}, \mathbf{s}') = \frac{\mathbf{G}(\mathbf{s}, \mathbf{s}')}{\nu} \text{ for } \mathbf{s} \neq \mathbf{s}', \text{ and } \mathbf{P}(\mathbf{s}, \mathbf{s}) = \frac{\mathbf{G}(\mathbf{s}, \mathbf{s})}{\nu} + 1, \text{ for all } \mathbf{s}, \mathbf{s}'.$$

The uniformization constant ν must be at least equal to the maximum of absolute values of $\mathbf{G}(\mathbf{s}, \mathbf{s})$; for JQNs, the value $\nu = \lambda + \sum_{m=1}^M \mu_m$ suffices. Let $\mathbf{U}^{(k)}$ be the state probability distribution matrix after k epochs in the DTMC with transition matrix \mathbf{P} . That is, entry (i, j) of $\mathbf{U}^{(k)}$ is the probability that j is reached from i in k steps. $\mathbf{U}^{(k)}$ can be derived recursively as:

$$\mathbf{U}^{(0)} = \mathbf{I}, \quad \text{and} \quad \mathbf{U}^{(k)} = \mathbf{U}^{(k-1)}\mathbf{P}, \quad k \in \mathbb{N}^+. \quad (4)$$

Then, the matrix of transient state probabilities for the original CTMC at time t , can be calculated as:

$$\mathbf{V}(t) = \sum_{k=0}^{\infty} \psi(\nu t; k) \mathbf{P}^k = \sum_{k=0}^{\infty} \psi(\nu t; k) \mathbf{U}^{(k)}, \quad (5)$$

where $\psi(\nu t; k)$ is the probability of k events occurring in the interval $[0, t)$ in a Poisson process with rate ν . The probability distribution in the DTMC after k steps is described by $\mathbf{V}(0) \cdot \mathbf{P}^k$ (note that $\mathbf{V}(0) = \mathbf{I}$).

Note that matrices $\mathbf{V}(t)$ and $\mathbf{U}^{(k)}$, $k \in \mathbb{N}$, have infinite size. To avoid the infinite summation over the number of steps k , the sum (5) needs to be truncated. We denote the approximation of $\mathbf{V}(t)$ that has been calculated with up to $n+1$ terms of the summation with $\mathbf{V}^{(n+1)}(t)$:

$$\mathbf{V}^{(n+1)}(t) = \sum_{k=0}^{n+1} \psi(\nu t; k) \mathbf{U}^{(k)} = \mathbf{V}^{(n)}(t) + \psi(\nu t; n+1) \mathbf{U}^{(n+1)}. \quad (6)$$

Note that $\mathbf{V}^{(n)}(t)$ follows the structure of the previous $\mathbf{U}^{(m)}$ ($m \leq n$) in terms of zeroes and non-zeroes because any non-zero entry in $\mathbf{V}^{(n)}$ corresponds to a non-zero in $\mathbf{U}^{(m)}$ ($m \leq n$). We denote a maximum bound on the error that possibly occurs in an entry of $\mathbf{V}(t)$ when the series is truncated after n steps as $\varepsilon_{t,\nu}^{(n)}$. For a given number of steps n , $\varepsilon_{t,\nu}^{(n)}$ increases linearly with $\nu \cdot t$ and decreases linearly with n :

$$\left\| \sum_{k=n+1}^{\infty} \psi(\nu t; k) \mathbf{U}^{(k)} \right\| \leq 1 - \sum_{k=0}^n e^{-\nu t} \frac{(\nu t)^k}{k!} = \varepsilon_{t,\nu}^{(n)}. \quad (7)$$

4.2 Finite representation

In the following we will use uniformization to compute the transient probabilities to reach all possible goal states from all (starting) states in a JQN. The homogeneous probability matrix \mathbf{P} contains the probability to reach a state \mathbf{s}' from a state \mathbf{s} within one step for all $\mathbf{s}, \mathbf{s}' \in \mathbf{S}$. From every possible starting state, only n fronts can be reached with n steps. Hence, for a given number of steps n all states that are $n+1$ steps away from the origin $\hat{\mathbf{s}}$ seem to be identical in the JQN. We only need to consider a finite part of the JQN, depending on the number of steps n . As we will see, the homogenous structure of the JQN and

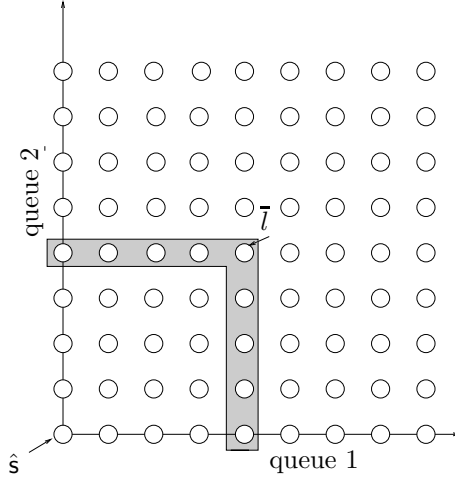
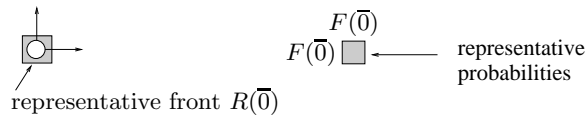


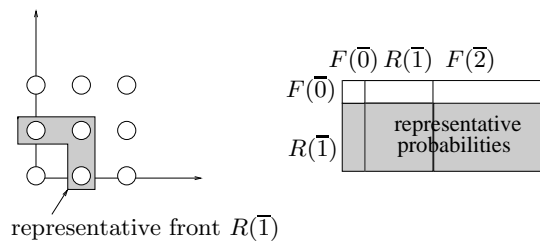
Fig. 2. Finite state space that needs to be considered for a given $\bar{l} = (4, 4)$

of the probability matrix implies that we obtain identical transient probabilities for states $s, s' \in S_r$ with $r \in R(\bar{l})$, within the error bounds of uniformization given n steps. In fact, we restrict the computation to a finite number of starting states and still perform a comprehensive transient analysis for every possible state as starting state. As shown in Figure 2, starting from every representative state $r \in R(\bar{n})$, still n steps can be undertaken in every direction without reaching beyond the origin \hat{s} . In a two-dimensional JQN the total amount of starting states we have to consider equals $(n+1)^2$ and the total amount of goal states equals $(2n+1)^2$. In an M dimensional setting $(l+1)^M$ starting states and $(2l+1)^M$ goal states have to be considered out of which $(l+1)^M - l^M$ states are representative. The matrix $\mathbf{U}^{(n)}$ is the state probability matrix after n discrete epochs and $\mathbf{V}^{(n)}(t)$ holds the approximated transient probabilities after n steps. Note that these matrices remain two-dimensional for JQNs, as they represent all possible combinations of starting states and goal states. It is now sufficient to consider only starting states that belong to fronts $F(\bar{i})$ for $i \leq n$ for a finite representation of $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$. 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. We now address the growth of the matrices $\mathbf{U}^{(n)}$ in the course of the computation. Figure 3(a) shows that the dimension of the finite representation of $\mathbf{U}^{(0)}$ is: $\dim(\mathbf{U}^{(0)}) = (|F(\bar{0})|)^2 = (1^M - 0^M)^2 = 1$. Since $n = 0$, we cannot leave a state and the first front $R(\bar{0})$ is already a representative front. Figure 3(b) shows the dimension of the finite representation of $\mathbf{U}^{(1)}$. Since $n = 1$, we can reach the next higher or the next lower fronts. Thus, front $F(\bar{0})$ cannot be used as representative front, but we can use the next higher front $R(\bar{1})$ as representative front, as shown in Figure 3(b). Since $n = 1$, it is possible to reach the front $F(\bar{2})$ as well; thus we have to consider starting in one of the first two fronts $F(\bar{i})$ for $i = \{0, 1\}$ and ending up in one of the first three fronts

(a) 0 steps



(b) 1 steps



(c) 2 steps

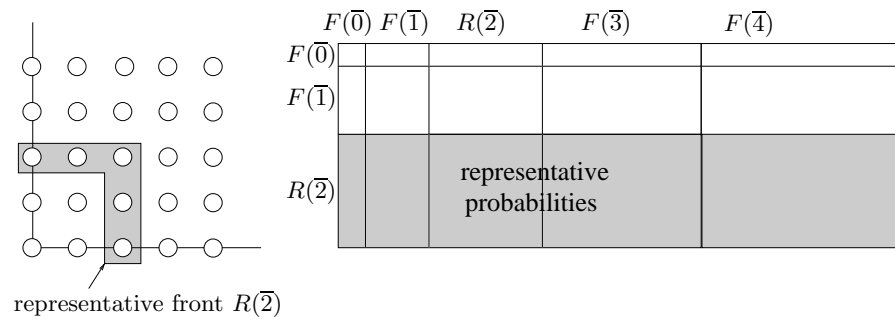


Fig. 3. Considered part of the state space (left) and finite representation of $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$ (right), depending on the number of considered steps

$F(\vec{j})$ for $j = \{0, 1, 2\}$. The dimension of the finite representation of $\mathbf{U}^{(1)}$ depends on the fronts that contain the starting states and on the fronts containing the goal states. The number of states of a given front can be calculated according to Equation (2). The dimension of $\mathbf{U}^{(1)}$ is given by:

$$\begin{aligned} \dim(\mathbf{U}^{(1)}) &= (|F(\vec{0})| + |R(\vec{1})|) \times (|F(\vec{0})| + |R(\vec{1})| + |F(\vec{2})|) \\ &= (1^M - 0^M + 2^M - 1^M) \times (1^M - 0^M + 2^M - 1^M + 3^M - 2^M) \\ &= 2^M \times 3^M. \end{aligned}$$

Figure 3(c) shows the finite representation of the matrix $\mathbf{U}^{(2)}$. From a given front, we can reach at most two more fronts in both directions. Picking the second front as new representative, ensures that we cannot reach beyond the origin \hat{s} . We have to attach another row of states to represent starting from the new representative front. Furthermore, we attach two more columns to account for the fronts $F(\vec{3})$ and $F(\vec{4})$ that can now be reached from the new representative front. In a general JQN, for a given number of steps n , the size of the matrix $\mathbf{U}^{(n)}$ is then

$$\dim(\mathbf{U}^{(n)}) = \left(\sum_{i=0}^n |F(\vec{i})| \right) \times \left(\sum_{j=0}^{2 \cdot n} |F(\vec{j})| \right) = (n+1)^M \times (2 \cdot n + 1)^M. \quad (8)$$

Note that, even though the left side of Figure 3 only shows the two dimensional case, ($M = 2$) the right side is also correctly depicted for an M dimensional setting. As before, the finite representation of the matrix $\mathbf{V}^{(n)}(t)$ has the same dimension as $\mathbf{U}^{(n)}$.

4.3 Uniformization with Representatives

We now proceed with the actual computation of the state probability matrix $\mathbf{U}^{(n)}$ and the approximated transient probability matrix $\mathbf{V}^{(n)}(t)$ according to (6). Starting with $n = 0$, and thus with the smallest finite portion of the JQN, cf. Figure 3(a), we increase n step by step, thus increasing accuracy and size of the considered finite representation of the JQN. However, in each iteration we always use the smallest possible representation. Considering n steps, the probability of starting in a state in the representative front $r \in R(\vec{n})$ and ending in a state s' in one of the fronts $F(\vec{i})$, for $i \in \{0, \dots, 2 \cdot n\}$, represents the probability of starting in a state $s \in S_r$ and ending in the corresponding state s'' . In order to increase the number of steps from $n - 1$ to n we first adapt the size of the data structure before computing the values for n steps. Moving from step $n - 1$ to n we have to add the front $F(\vec{n})$ that is going to be representative for n steps, to the set of starting states and the fronts $F(2\vec{n} - \vec{1})$ and $F(2\vec{n})$ to the set of goal states. First, the two new sets of columns of goal states are initialized with zero, as it is impossible to reach these states with $n - 1$ steps. Second, the new row of starting states $F(\vec{n})$ is initialized with the probabilities of the corresponding entries from front $R(\vec{n} - \vec{1})$ that is representative for $n - 1$

steps. Note that this holds for $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$. An entry (s, s') in the new row of starting states $F(\bar{n})$ constitutes moving from a starting state s to a goal state s' with $s \in F(\bar{n})$ and $s' \in F(\bar{i})$ for $i = 0, \dots, 2n$. We first need to find the corresponding starting state $r \in R(\bar{n} - \bar{1})$ such that $s \in S_r$. The corresponding goal state then is the state s'' that is, in every dimension, exactly as far away from r than s' is from s , ($r - s'' = s - s'$). Given a tuple of starting and goal state (s, s') with $s \in F(\bar{n})$, the corresponding tuple (r, s'') with $r \in R(\overline{n-1})$ is given by $r = s - \bar{h}(s)$ and $s'' = s' - s + r$, with

$$\bar{h}(s) = \begin{cases} h_i = 1, & s_i = n, \\ h_i = 0, & s_i \neq n, \end{cases} \quad (9)$$

The matrices $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}(t)$ have a block structure, according to the fronts of a JQN; we denote the blocks that give the probabilities from states in front $F(\bar{i})$ to states in front $F(\bar{j})$ as $\mathbf{U}_{\bar{i}, \bar{j}}^{(n)}$ and $\mathbf{V}_{\bar{i}, \bar{j}}^{(n)}(t)$. Note that \mathbf{P} can also be organized according to this block structure. In iteration step n , we then need to multiply the enlarged representation of $\mathbf{U}^{(n-1)}$ with the square part of \mathbf{P} that accounts for the one-step probabilities for all states in the first $2 \cdot n$ fronts. In general, for $n \geq 1$, $\mathbf{U}^{(n)}$ is computed as $\mathbf{U}^{(n-1)} \cdot \mathbf{P}_{\bar{i}}$ cf. (4), as follows:

$$\mathbf{U}_{\bar{i}, \bar{j}}^{(n)} = \sum_{k=0}^{2n+1} \mathbf{U}_{\bar{i}, \bar{k}}^{(n-1)} \cdot \mathbf{P}_{\bar{k}, \bar{j}}, \quad (10)$$

for $\bar{i} = \bar{0}, \dots, \bar{n}$ and $\bar{j} = \bar{0}, \dots, 2 \cdot \bar{n}$. Due to the block structure of $\mathbf{V}^{(n)}(t)$, we can rewrite (6) as:

$$\mathbf{V}_{\bar{i}, \bar{j}}^{(n)}(t) = \mathbf{V}_{\bar{i}, \bar{j}}^{(n-1)}(t) + \psi(\nu t; n) \cdot \mathbf{U}_{\bar{i}, \bar{j}}^{(n)}, \quad (11)$$

again for $\bar{i} = \bar{0}, \dots, \bar{n}$ and $\bar{j} = \bar{0}, \dots, 2 \cdot \bar{n} + \bar{1}$.

4.4 Complexity issues

In the k -th iteration, we actually consider the states of the first k fronts as starting states and the states of the first $2 \cdot k$ fronts as goal states, resulting in matrices with $(k+1)^M \times (2 \cdot k + 1)^M$ entries, as given by (8). If n is the maximum number of steps considered, the overall storage complexity for the three probability matrices $\mathbf{U}^{(n-1)}$, $\mathbf{U}^{(n)}$, $\mathbf{V}^{(n)}$ and the discrete transition matrix \mathbf{P} is $\mathcal{O}(4n^{2 \cdot M})$. The k -th multiplication of matrix $\mathbf{U}^{(n-1)}$ with \mathbf{P} is carried out in $\mathcal{O}(k^{6 \cdot M})$. For n the maximum number of considered steps, the overall time complexity therefore equals $\mathcal{O}(n^{6 \cdot M + 1})$. Note that the iteration costs per step increase. However, when full probability matrices of the size $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are used throughout the complete computation, the iteration costs are much higher.

5 How to stop?

For model checking an until-formula $\mathcal{P}_{\triangleright p}(\Phi \mathcal{U}^{[t_1, t_2]} \Psi)$ we have to compare for each starting state the probability to follow a $(\Phi \mathcal{U}^{[t_1, t_2]} \Psi)$ -path with the probability bound p . In the transformed JQN $\mathcal{J}[-\Phi \vee \Psi]$ the set of goal states consists

of all Ψ -states. We denote the probability to end up in a Ψ -state before time t , given starting state \mathbf{s} , as $\gamma_{\mathbf{s}}(t)$. For the time interval $I = [0, t]$, we have:

$$\gamma_{\mathbf{s}}(t) = \sum_{i=0}^{\infty} \sum_{\mathbf{s}' \in \text{Sat}^{F(\tau)}(\Psi)} \pi^{\mathcal{J}[\neg\Phi \vee \Psi]}(\mathbf{s}, \mathbf{s}', t).$$

Note that the vector $\gamma(t)$ consists of sub-vectors corresponding to the fronts of the JQN. The approximation of $\gamma_{\mathbf{s}}(t)$ after n iterations is denoted $\gamma^{(n)}(t) = \mathbf{V}^{(n)}(t) \cdot \gamma(0)$, with

$$\gamma_{\mathbf{s}}(0) = \begin{cases} 1, & \mathbf{s} \models \Psi, \\ 0, & \text{otherwise.} \end{cases}$$

In principle, $\gamma^{(n)}(t)$ is of infinite size, but we can cut it to a finite representation, as from a representative front on, corresponding states have the same probability values. For all states $\mathbf{s} \in \mathcal{S}$, we add the computed transient probabilities to reach any Ψ -state and check whether the accumulated probability meets the bound p on a regular basis. The accumulated probability is always an underestimation of the actual probability. Recall that $\varepsilon_{t,\nu}^{(n)}$ is the maximum error of uniformization after n iteration steps (cf. (7)), such that $\gamma_{\mathbf{s}}(t) \leq \gamma_{\mathbf{s}}^{(n)}(t) + \varepsilon_{t,\nu}^{(n)}$ for time interval $I = [0, t]$. From (7) it follows that the value of $\varepsilon_{t,\nu}^{(n)}$ decreases as n increases. Exploiting the above inequality, we obtain the following stopping criteria:

- (a) $\gamma_{\mathbf{s}}^{(n)}(t) \geq p \Rightarrow \gamma_{\mathbf{s}}(t) \geq p$,
- (b) $\gamma_{\mathbf{s}}^{(n)}(t) < p - \varepsilon_{t,\nu}^{(n)} \Rightarrow \gamma_{\mathbf{s}}(t) < p$.

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, check again, etc., until either of the stopping criteria holds. However, if for one of the starting states $\mathbf{s} \in \mathcal{S}$ we have $\gamma_{\mathbf{s}}(t) = p$, the iteration never stops, as neither of the stopping criteria ever holds. However, this is highly unlikely to occur in practice. In case $(\neg\Phi \vee \Psi)$ is independent as of \bar{g} and either (a) or (b) holds for all considered starting states with n steps, front $R(\bar{g} + \bar{n})$ is representative and the transient probabilities for all $\mathbf{s} \in S_r$ computed with n steps will be the same. $\mathcal{P}_{\times p}(\Phi \mathcal{U}^{[0,t]}\Psi)$ then is independent as of $\bar{g} + \bar{n}$. In that case, we check for all states $\mathbf{s} \leq \bar{g} + \bar{n}$ whether the accumulated transient probability of reaching a Ψ -state meets the bound p . The representative states that satisfy $\mathcal{P}_{\times p}(\Phi \mathcal{U}^{[0,t]}\Psi)$ form the representative satisfaction set $\text{Sat}^{R(\bar{g}+\bar{t})}(\mathcal{P}_{\times p}(\Phi \mathcal{U}^{[0,t]}\Psi))$.

6 Case study: An e-business site

Modeling an e-business site as Jackson queueing network facilitates analyzing its scalability. This is extremely important as customers become dissatisfied easily in case such a site is overloaded. We are able to model an e-business site in as

much detail as shown in [10], however, we use a model with one queue per server instead of two, to keep the model concise. On the other hand, where [10] only analyzes average response times, we are able to analyze a wide range of more advanced measures, given by the logic CSL and the new analysis algorithm.

6.1 System description and model

Consider an online retail shop, where requests arrive from a potentially infinite customer base. The site itself consists of three servers: a web server, an application server and a database server. The requests are first dealt with by the web server that manages all the web pages and handles the direct interactions with the customer. The application server implements the core logic of the site and the database server stores persistent information about registered customers, prices and article descriptions. Arriving requests first visit the web server, after which they are either forwarded to the application server, routed back to the web server itself or leave the system, when they have been completed. Jobs that visit the application server are either forwarded to the database server or routed back to either the web server or the application server. From the database server, jobs are routed to either the application server or back to the database server itself. Note that requests can only leave the system via the web server. As illustrated in Figure 4, the associated JQN then consists of three unbounded queues modeling the buffer of the web server, the buffer of the application server and the buffer of the database server, respectively. Requests from the infinite population arrive according to a Poisson process with rate λ and are then routed as shown in Figure 4. The arrival rates per queue that follow from solving the traffic equation and the service rates per queue are given in Table 1. To analyze the scalability of the e-business site, we define the CSL formula `overflow` to indicate that all queues are filled above a certain threshold as

$$\text{overflow} = (s_1 \geq \text{full}) \vee (s_2 \geq \text{full}) \vee (s_3 \geq \text{full}),$$

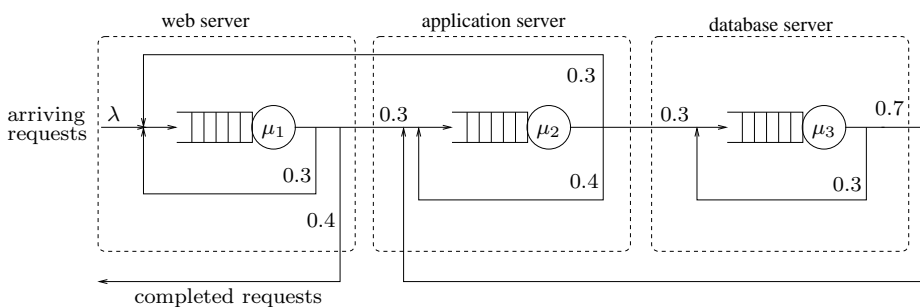


Fig. 4. Queueing network model for an online auction site

parameter	λ_1	λ_2	λ_3	μ_1	μ_2	μ_3
sec^{-1}	$\frac{5}{2} \cdot \lambda$	$\frac{5}{2} \cdot \lambda$	$\frac{15}{14} \cdot \lambda$	5	5	3

Table 1. Numerical values for the parameters of the model

for different possible values of full. The atomic proposition

$$\text{no_overflow} = \neg \text{overflow} = (s_1 < \text{full}) \wedge (s_2 < \text{full}) \wedge (s_3 < \text{full})$$

indicates that all queues contain less than full requests.

6.2 Model checking time-bounded until

Figure 5 shows the number of uniformization steps needed for model checking

$$\text{Sat}(\mathcal{P}_{\geq p}(\text{overflow } \mathcal{U}^{[0,t]} \text{ no_overflow})) \text{ for } t = \{5; 10; 5\},$$

depending on the probability bound p . We show the number of iterations with the dynamic stopping criterion, as well as the a priori computed number of steps required for an error $\varepsilon_{t,\nu}^{(n)} = 10^{-7}$. Clearly, the a priori number of steps is independent of the probability bound p and increases with time bound t . After 0 steps the comparison can be evaluated for $p = 0$ for all time bounds when using the dynamic stopping criterion. Then the number of iterations first increases steeply and the maximum number of iterations is reached for a probability bound at most 0.2 for all four time bounds. In general, the number of iteration steps using the dynamic stopping criterion decreases for larger p . Note that the step size of p , as shown in Figure 5 was taken to be 0.01. The number of iterations in Figure 5 clearly varies over time. A peak occurs whenever the computed probability for some state is really close to the probability bound p we have to compare with. The maximum number of iterations with the dynamic stopping criterion approximates the a priori computed number of steps for $\varepsilon_{t,\nu}^{(n)} = 1 \cdot 10^{-7}$. For larger time bounds t , the difference between the number of iterations for the dynamic and the a priori stopping criterion increases, showing the efficiency gain using the dynamic stopping criterion. In Table 2 the first group of rows show the minimum and maximum number of iterations, depending on the probability bound p , per time bound with the dynamic stopping criterion and the a priori computed number of iterations per time bound. The second group of rows then show the finite number of states that is considered of the underlying infinite Markov chain \mathcal{J} , depending on the number of iterations and again depending on the time bound t . The corresponding number of states in the absorbing Markov chain $\mathcal{J}[\neg\Phi \vee \Psi]$ is shown in third row. In the last group of rows, the numerical error $\varepsilon_{t,\nu}^{(n)}$ for the corresponding number of iterations is given. Using the dynamic stopping criterion, the number of iterations that is necessary to decide whether $s \models \mathcal{P}_{\geq p}(\text{overflow } \mathcal{U}^{[0,t]} \text{ no_overflow})$ for all $s \in \mathcal{S}$ grows with increasing time bound t . This is due to the fact, that with a larger time bound

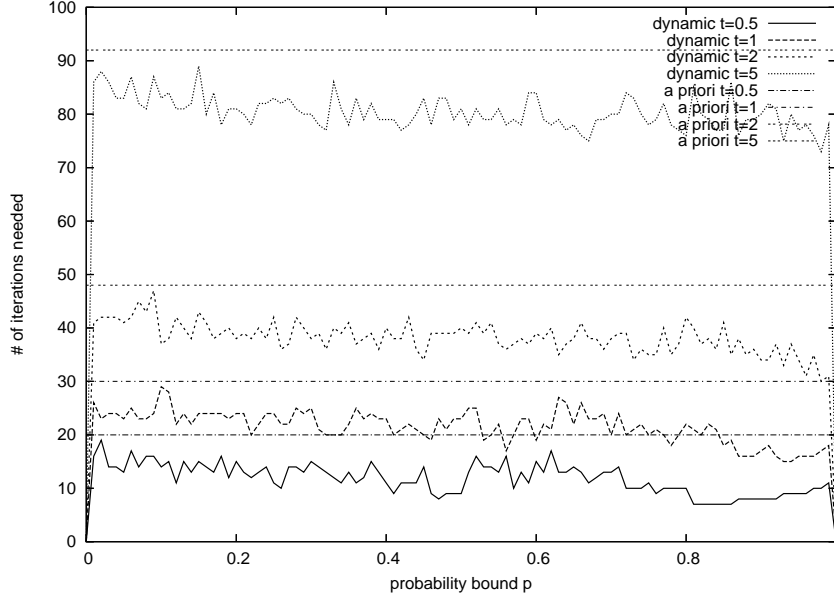


Fig. 5. Number of iterations needed for model checking $s \models \mathcal{P}_{\geq p}(\text{overflow } \mathcal{U}^{[0,t]} \text{ no_overflow})$ with the dynamic stopping criterion and with a priori error

more steps can be taken. With an increasing number of iterations also the considered finite part of the underlying infinite Markov chain grows. In contrast, with more iterations, the introduced numerical error $\varepsilon_{t,\nu}^{(n)}$ decreases. Therefore, the error bound in column *dynamic min* is larger than the error bound in column *dynamic max*. For time bound $t = 5$, $\varepsilon_{t,\nu}^{(n)} = 1.1 \cdot 10^{-7}$ is enough to decide that $s \models \mathcal{P}_{\geq p}(\text{overflow } \mathcal{U}^{[0,t]} \text{ no_overflow})$ for probability bound $p = 0.98$. Whereas $\varepsilon_{t,\nu}^{(n)} = 3.45 \cdot 10^{-7}$ is enough to decide this for all probability bounds $p \in \{0.0, 0.01, 0.02, \dots, 0.99, 1.0\}$. However, this small error is only necessary to decide the validity of the CSL formula for probability bound $p = 0.15$. Note that the given number of iterations and the given error might not be enough to decide for every other probability bound. The last column of Table 2 shows the number of iterations that has to be undertaken to keep $\varepsilon_{t,\nu}^{(n)} \leq 1 \cdot 10^{-7}$. Figure 5 shows that an error of $1 \cdot 10^{-7}$ is always enough to decide whether $s \models \mathcal{P}_{\geq p}(\text{overflow } \mathcal{U}^{[0,t]} \text{ no_overflow})$ for all $s \in \mathcal{S}$. The number of states of the infinite Markov chain that has to be considered is always slightly larger than for the maximum in case the dynamic stopping criterion is used.

6.3 Tool usage

To model check the time bounded until operator the JQN has been transformed manually into a stochastic Petri net [5]. To model the possible infinite population

	t	dynamic		a priori
		min	max	$\varepsilon_{t,\nu}^{(n)} = 1 \cdot 10^{-6}$
number of iterations	0.5	7	19	20
	1	15	29	30
	2	30	47	48
	5	73	89	92
number of states in \mathcal{J}	0.5	7770	260130	273819
	1	23426	877975	1004731
	2	91881	2081156	2362041
	5	782246	1333300	1456935
number of states in $\mathcal{J}[\neg\Phi \vee \Psi]$	0.5	121768	252132	265821
	1	532276	869977	996733
	2	1406912	2073158	2354043
	5	782246	1325302	1448937
uniformization error $\varepsilon_{t,\nu}^{(n)}$	0.5	$1.3 \cdot 10^{-1}$	$4.0 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
	1	$2.4 \cdot 10^{-2}$	$3.1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
	2	$1.5 \cdot 10^{-2}$	$1.0 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
	5	$1.1 \cdot 10^{-3}$	$3.45 \cdot 10^{-7}$	$1 \cdot 10^{-7}$

Table 2. Numerical values for the parameters of the model

of the JQN, an additional place si added to the SPN from which all arrivals take place and to which all departures are routed. In case the inner formula $\neg\Phi \vee \Psi$ is independent of g and given the number of iterations is n , the place *finite* is initialized with $g + 2 \cdot n$ tokens. Then the CSPL implementation by Bell [3] is used to generate the underlying Markov chain and an implementation of the uniformization method for finite state Markov chains by Cloth [6] is used to compute the transient probabilities. A script then emulated the dynamic behavior of the algorithm. The time to compute the transient probabilities ranges from 0.4 seconds to 22 seconds for the different time bounds, when using the dynamic stopping criterion, and between 1.3 seconds and 26 seconds, when using the a priori stopping criterion.

7 Conclusion

In this paper we presented a model checking algorithm for efficiently checking the time-bounded until operator labeled Jackson queueing networks. Note that with this algorithm the until operator with time interval $[t_1, t_2]$ and with time interval $[t, t]$ can be model checked on JQNs along the same lines as presented for QBDs in [13]; even more details on this can be found in [11]. Hence, we have shown that it is possible to carry the idea of uniformization with representatives to other highly structured classes of infinite-state CTMCs as well.

Storage complexity and the computational complexity for doing uniformization with representatives on JQNs is much higher than for uniformization with representatives on QBDs. This is due to the fact that the state space of a QBD

grows without bound in just one direction, whereas the state space of a JQNs grows without bound in as many directions as the JQN has queues.

Note that we used an approximate algorithm to construct a decision procedure for model checking the until operator. The approximate algorithm, uniformization, computes the transient probabilities for a given error bound. Comparing the computed probability with the given probability bound and the uniformization error allows us to decide for a given starting state whether or not a CSL formula that contains the until operator is valid.

References

1. A. Aziz, K. Sanwal, and R. Brayton. Model checking continuous-time Markov chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.
2. C. Baier, B.R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, 2003.
3. A. Bell. *Distributed evaluation of stochastic Petri nets*. PhD thesis, Dept. of Computer Science, RWTH Aachen, 2004.
4. P. Buchholz. A class of hierarchical queueing systems and their analysis. *Queueing Systems*, 15:59–80, 1994.
5. G. Ciardo. Discrete-time Markovian stochastic Petri nets. In *Computations with Markov Chains*, pages 339–358. Raleigh, 1995.
6. L. Cloth, J.-P. Katoen, M. Khattri, and R. Pulungan. Model checking Markov reward models with impulse rewards. In IEEE Press, editor, *Int. Conf. on Dependable Systems and Networks (DSN2005)*, 2005.
7. D. Gross and D.R. Miller. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Operations Research*, 32(2):343–361, 1984.
8. P.G. Harrison. Transient behaviour of queueing networks. *Journal of Applied Probability*, 18(2):482–490, 1981.
9. J.R. Jackson. Networks of waiting lines. *Operations Research*, 5(4):518–521, 1957.
10. D.A. Menasce, V.A.F. Almeida, and L.W. Dowdy. *Performance by Design*. Prentice Hall, 2004.
11. A. Remke. *Model checking structured infinite Markov chains*. PhD thesis, Dept. of Computer Science, University Twente, 2008.
12. A. Remke and B.R. Haverkort. CSL model checking algorithms for infinite-state structured Markov chains. In *Formats '07*, number 4763 in LNCS, pages 336–351. Springer, 2007.
13. A. Remke, B.R. Haverkort, and L. Cloth. CSL model checking algorithms for QBDs. *Theoretical Computer Science*, 2007.