

# Approximation of discrete-time polling systems via structured Markov chains

P. Beekhuizen  
Philips Research  
Eindhoven, The Netherlands  
and  
EURANDOM  
Eindhoven University of Technology  
beekhuizen@eurandom.tue.nl

J.A.C. Resing  
Eindhoven University of Technology  
Department of Mathematics and Computer  
Science  
Eindhoven, The Netherlands  
resing@win.tue.nl

## ABSTRACT

We devise an approximation of the marginal queue length distribution in discrete-time polling systems with batch arrivals and fixed packet sizes. The polling server uses the Bernoulli service discipline and Markovian routing. The 1-limited and exhaustive service disciplines are special cases of the Bernoulli service discipline, and traditional cyclic routing is a special case of Markovian routing. The key step of our approximation is the translation of the polling system to a structured Markov chain, while truncating all but one queue. Numerical experiments show that the approximation is very accurate in general. Our study is motivated by networks on chips with multiple masters (e.g., processors) sharing a single slave (e.g., memory).

## Categories and Subject Descriptors

G.3 [Probability and Statistics]: Queueing Theory

## General Terms

Theory, Performance

## Keywords

Polling, Bernoulli scheduling, Markovian routing, Networks on chips

## 1. INTRODUCTION

In this paper, we devise an approximation of the queue length distribution of a discrete-time polling system with batch arrivals, fixed packet sizes, Bernoulli service, and Markovian routing. Bernoulli service means that after service of a packet from queue  $i$ , the server serves queue  $i$  again with probability  $q(i)$  and moves to another queue with probability  $1 - q(i)$ . Markovian routing means that if the server moves to

another queue, it moves to queue  $j$  with probability  $P(i, j)$  for  $j \neq i$ , independently of everything else.

Our study is primarily motivated by networks on chips. Networks on chips are an emerging paradigm for the connection of on-chip modules like processors and memories. Such modules are traditionally connected via single buses, but because these buses cannot be used by multiple modules simultaneously, communication difficulties arise as the number of modules increases. Networks on chips have been proposed as a solution (see [12]). In networks on chips, routers are used to transmit packets to their destination, so that multiple links can be used at the same time and communication becomes more efficient.

We are in particular motivated by networks on chips where all traffic has the same destination. Such networks occur for instance if multiple masters (e.g., processors) share a single slave (e.g., memory). In this case, multiple queues in a router share a single destination link, so the routers can be seen as polling systems.

Routers in networks on chips often use a round robin scheduler, which, in polling terminology, corresponds to cyclic routing and 1-limited service. We are therefore mostly interested in the cyclic 1-limited model, a special case of our model, but our approximation is aimed at the more general polling system with Bernoulli service and Markovian routing. This also implies that, although we are primarily motivated by networks on chips, the range of applications of our approximation extends beyond them.

Recently, it was shown that a network of polling systems can be reduced to a single station while preserving queue lengths [1], provided the service and routing discipline are *HoL-based*, which is a class that contains Bernoulli service combined with Markovian routing. Most importantly, this implies that such networks can be analysed by means of single-station results [2], as provided in this paper.

The essential part of our approximation is the translation of the polling system to a Structured Markov Chain (SMC) of the  $M/G/1$  type (see [23]). An SMC of  $M/G/1$  type is a Markov chain of which the states can be described as tuples  $(l, \phi)$ , where  $l$ , called the level, is an element from  $\{0, 1, \dots\}$  and  $\phi$ , called the phase, an element from some finite set (the phase space). The Markov chain has a transition probability matrix of the following block-partitioned form (hence the

name structured):

$$\begin{pmatrix} B_0 & B_1 & B_2 & B_3 & \dots \\ C_0 & A_1 & A_2 & A_3 & \dots \\ 0 & A_0 & A_1 & A_2 & \dots \\ 0 & 0 & A_0 & A_1 & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \quad (1)$$

The matrix  $A_0$  describes transitions where the level decreases by one,  $A_1$  describes transitions within a level, and so on. The behaviour of the Markov chain at the boundary  $l = 0$  may be different from that in the interior  $l > 0$ , which is reflected by the matrices  $B_k$ ,  $k = 0, 1, \dots$ , and  $C_0$ .

The idea behind our approximation is simple yet very effective: Instead of analysing the queue contents of *all* queues at the same time, we focus on the precise contents of *one* queue. For the other queues we only keep track of whether there are  $0, 1, \dots, B-1$ , or ' $B$  or more' packets in the queues. By also keeping track of the index of the queue that is being served, we obtain an SMC of the  $M/G/1$  type.

The truncation of queue lengths implies that we have to introduce additional parameters representing the probability that the contents of a queue go from ' $B$  or more' to  $B-1$ . The values of these unknown parameters are determined iteratively. In each iteration the equilibrium distribution of the SMCs is used to update the values of these parameters, until the values have converged. As  $B \rightarrow \infty$ , our approximation becomes exact. However, setting  $B = 2$  already leads to an accurate approximation.

This paper is organised as follows: In Section 2 we describe our model in more detail and in Section 3 we give an overview of the relevant literature. We describe our approximation in Section 4 and its accuracy is studied in Sections 5 and 6: In Section 5 we perform a detailed analysis of a single case, and in Section 6 a global analysis of multiple cases. We present our conclusions in Section 7.

In the implementation of our approximation, we used Kronecker products to determine the transition probability matrix. However, for the sake of readability, we describe our approximation without Kronecker products in Section 4. In Appendix A, we show how the transition probability matrix can be obtained using Kronecker products, which makes the computations significantly faster.

## 2. MODEL DESCRIPTION

We assume that all packets have the same size, which we assume to be 1. Packets arrive to the queues according to independent Bernoulli batch arrival processes (i.e., the numbers of arrivals in each time slot are i.i.d.).

Because networks on chips operate in the discrete-time domain, our polling model does so too. The model for which we derive our approximation is thus a discrete-time polling model with  $N$  infinite queues and the following characteristics:

1. Batch Bernoulli arrival processes.
2. Deterministic service times, equal to 1.
3. Bernoulli service discipline, i.e., after service of queue  $i$  the server serves queue  $i$  again with probability  $q(i)$  and moves to another queue with probability  $1 - q(i)$ .

4. Markovian routing, i.e., if the server moves, it moves from queue  $i$  to queue  $j \neq i$  with probability  $P(i, j)$ . We assume  $P(i, i) = 0$ .

5. Zero switch-over times.

The Bernoulli service discipline and Markovian routing dictate that, after a service completion at queue  $i$ , the server moves to queue  $j \neq i$  with probability  $(1 - q(i))P(i, j)$  and stays at queue  $i$  with probability  $q(i)$  (in which case we say  $j = i$ ). If queue  $j$  is empty, but not *all* queues are empty, the server immediately moves to another queue according to the routing matrix  $P$ , and again if this queue is empty too, and so on, until it finds a non-empty queue. If *all* queues are empty the server remains at queue  $j$ . When new packets arrive to any of the queues, the server again moves according to the routing matrix  $P$ , until it moves to one of the non-empty queues. All of these movements happen instantaneously.

**REMARK 2.1.** *The model described above is not equivalent to a model where the server always moves according to a matrix  $R$ , with  $R(i, i) = q(i)$  and  $R(i, j) = (1 - q(i))P(i, j)$ , until it finds a non-empty queue. With the exhaustive service discipline,  $R(i, i) = 1$ , which means that the server stays at queue  $i$  indefinitely, even after it has become empty. Such behaviour cannot occur in our model because  $P(i, i) = 0$ .*

To prevent a situation where the server cannot reach some non-empty queues, we assume  $P$  is irreducible. Furthermore, we assume that arrivals and service completions happen at the end of time slots, and that packets arriving to an empty queue at the end of time slot  $[t - 1, t)$  may be served in time slot  $[t, t + 1)$ . Finally, we assume the polling system is stable. Because the switch-over times are zero, the polling system is work-conserving and therefore stable if the total load is less than 1. We thus assume  $\sum_{k,l} l x_k(l) < 1$ , where  $x_k(l)$  denotes the probability that  $l$  arrivals occur to queue  $k$ .

## 3. RELEVANT LITERATURE

In this section, we review the literature relevant to our study of polling systems with Bernoulli service and Markovian routing. The Bernoulli service discipline was introduced by Keilson and Servi [17] in a *single-queue* vacation system where after each service completion the server takes a vacation with probability  $p$ . Tedijanto [28] later analysed a *multi-queue* polling system with the Bernoulli service discipline.

Markovian routing was analysed by Boxma and Weststrate in [10]. They derive the pseudo-conservation law for Markovian routing combined with the traditional service disciplines (exhaustive, gated, and 1-limited), see also Weststrate [30]. Independently of Boxma and Weststrate, Srinivasan [27] considers a similar system, but in a slightly more general setting.

More important to our work, however, are results on queue lengths. Resing [24] established that polling systems satisfying a certain 'branching property' can be viewed as branching processes and exact results can be obtained. For polling systems that do not satisfy the branching property, such as systems with the Bernoulli service discipline, even *mean* queue lengths are unknown, except for special cases such as 2-queue and symmetric systems. The 2-queue system

with Bernoulli service was analysed by Feng et al. [13]. The symmetric case with 1-limited service and a special case of Markovian routing, namely where  $P(i, j)$  is independent of  $i$ , was analysed by Kleinrock and Levy [18].

Since exact results are known only for special cases, and their derivation gives little hope for extensions to more general cases, we focus on approximations instead. In research of polling systems, the continuous-time domain receives far more attention than the discrete-time domain. One of the few examples of discrete-time approximations is the approximation by Frigui and Alfa [14], where a polling model with time-limited service is studied.

In the continuous-time domain, the queue length distribution in a polling system with the Bernoulli service discipline and cyclic routing was analysed using the power series approximation of Blanc [5–8]. For the cyclic 1-limited system, the most important special case of our model, other approximations of the queue length distributions exist: First, Van Vuuren and Winands [29] use structured Markov chains to approximate queue lengths in a  $k_i$ -limited polling system with cyclic routing. Their approach, however, is very different from ours since they use structured Markov chains to approximate visit and intervisit periods. Second, Leung obtains an approximation for a polling system with the probabilistically limited service discipline (which includes 1-limited) using discrete Fourier transforms [21]. Third, Lee and Sengupta approximate queue length distributions in a polling model with a reservation mechanism using an iterative approximation of visit and intervisit periods [20].

Most of these approximations can probably be extended to the discrete-time domain. However, doing so often involves subtleties and requires careful checking of each step. Furthermore, the approximations mentioned above, except Blanc's, are aimed at models with positive switch-over times. It is unclear whether good approximations can be obtained for the model without switch-over times by taking a limit where switch-over times tend to 0. Especially if the queue length approximation makes use of an approximation of visit and intervisit periods, such an approach seems problematic as both are 0 with probability 1 in the limit.

Other authors, such as Boxma and Meister [9], Fuhrmann and Wang [15], Levy and Groenendijk [16], and Srinivasan [26], only approximated *mean* waiting times in cyclic 1-limited polling systems (and by Little's law, mean queue lengths). Of those, we found that the Boxma-Meister [9] and Levy-Groenendijk [16] approximations could be extended to the discrete-time domain without much additional effort. We will compare our approximation with these two approximations in Section 5.

## 4. THE ALGORITHM

In this section we describe our algorithm in more detail. First, we introduce the phase spaces and derive the transition probability matrices. The iterative determination of the probabilities that the contents of the truncated queues go from ' $B$  or more' to  $B - 1$  is discussed at the end of this section.

For every queue, we construct an SMC such that the exact contents of that queue are stored in the level. The truncated contents of the other queues, as well as the index of the queue in service (called the service index) are stored in the phase.

The SMC where the contents of queue  $i$  are stored in the level, is called the 'SMC of queue  $i$ '. All SMCs describe the state of the system at integral times  $t$ , so immediately before the start of the service of a packet and immediately after arrivals, departures and server movements.

Every phase of the SMC of queue  $i$  is described by a vector  $(j, n_1, \dots, n_{i-1}, n_{i+1}, \dots, n_N)$ , where  $j$  is the service index,  $n_k = 0, \dots, B - 1$  means there are  $n_k$  packets in queue  $k$ , and  $n_k = B$  means there are  $B$  or more packets in queue  $k$ ,  $k \neq i$ . The phase space of level  $n_i > 0$  consists of all such combinations, except that the service index cannot be  $j$  if queue  $j$  is empty. We thus obtain

$$\Phi_i = \{1, \dots, N\} \times \{0, 1, \dots, B\}^{N-1} \setminus \{(j, n_1, \dots, n_{i-1}, n_{i+1}, \dots, n_N) : n_j = 0 \text{ (for } j \neq i)\}$$

as the phase space for level  $n_i > 0$ .

The phase space of level  $n_i = 0$ , is different: First, queue  $i$  cannot be served because it is empty. Second, if all queues are empty, the server waits at queue  $j$  until new packets arrive to any of the queues. Hence, the phase space of level  $n_i = 0$  is

$$\tilde{\Phi}_i = \{1, \dots, N\} \times \{0, 1, \dots, B\}^{N-1} \setminus \{(j, n_1, \dots, n_{i-1}, n_{i+1}, \dots, n_N) : j = i \text{ or } n_j = 0\} \cup \{(j, 0, \dots, 0) : j = 1, \dots, N\},$$

where  $(j, 0, \dots, 0)$  means that the server is waiting at queue  $j$  until new packets arrive.

**REMARK 4.1.** *The meaning of phase  $(i, 0, \dots, 0)$  depends on whether it is combined with level  $n_i > 0$  or  $n_i = 0$ . If  $n_i > 0$  phase  $(i, 0, \dots, 0)$  means that a packet from queue  $i$  will be served in the next time slot and that all other queues are empty. If  $n_i = 0$  it means that the entire system, including queue  $i$ , is empty and the server is waiting at queue  $i$ .*

In order to describe the transition probability matrix of the SMC of queue  $i$ , we divide the movement of the server from one queue to another into two parts: First, the server chooses a queue it *would like* to serve, regardless of whether this queue is empty or not, i.e., the server stays at queue  $j$  with probability  $q(j)$  and moves to queue  $k$  with probability  $(1 - q(j))P(j, k)$ . Second, the server keeps moving according to matrix  $P$  until it finds a non-empty queue (provided the server was not already at a non-empty queue, and not all queues are empty).

The transition probability matrix of the SMC of queue  $i$  is now given by

$$\begin{pmatrix} B_{i,0}\tilde{\Psi}_i & B_{i,1}\Psi_i & B_{i,2}\Psi_i & B_{i,3}\Psi_i & \dots \\ A_{i,0}\tilde{\Psi}_i & A_{i,1}\Psi_i & A_{i,2}\Psi_i & A_{i,3}\Psi_i & \dots \\ 0 & A_{i,0}\Psi_i & A_{i,1}\Psi_i & A_{i,2}\Psi_i & \dots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \quad (2)$$

Here, the matrices  $A_{i,l}$  and  $B_{i,l}$  describe arrivals, departures, and the first part of the server movements. The matrices  $\Psi_i$  and  $\tilde{\Psi}_i$  describe the second part of the server movements. Because queue  $i$  is empty if and only if the process is in level  $n_i = 0$ , there are different matrices for level  $n_i = 0$  and levels  $n_i > 0$ , denoted by  $\tilde{\Psi}_i$  and  $\Psi_i$  respectively. These matrices are specified in more detail below. The elements of matrices  $A_{i,l}$  are denoted by  $A_{i,l}(\cdot, \cdot)$ , and likewise for  $B_{i,l}$ .

REMARK 4.2. The matrices  $A_{i,l}$  give probabilities of transitions from phases in the phase space  $\Phi_i$  to all possible vectors  $(j, n_1, \dots, n_{i-1}, n_{i+1}, \dots, n_N)$  with  $j = 1, \dots, N$  and  $n_k \in \{0, 1, \dots, B\}$ , especially including those where the server is positioned at an empty queue. The matrix  $\Psi_i$  describes transitions from such vectors to phases in  $\Phi_i$ , where the server is not allowed to be positioned at empty queues. The products of these matrices thus indeed give transition probabilities on the phase space  $\Phi_i$ .

### Determination of $\Psi_i$ and $\tilde{\Psi}_i$

The matrices  $\Psi_i$  and  $\tilde{\Psi}_i$  can be determined as follows: Suppose that the server is positioned at an empty queue, but not all queues are empty. Define  $I$  and  $J$  as the subsets of empty and non-empty queues, respectively. The matrix  $P$  constitutes a Markov chain on  $\{1, \dots, N\}$ . The probability that the server moves from  $i \in I$  to  $j \in J$  is equal to the probability that the first visit of that Markov chain to set  $J$  occurs at state  $j$ , given that the Markov chain starts in state  $i$ . The matrices  $\Psi_i$  and  $\tilde{\Psi}_i$  follow from computing that probability for all phases (see e.g. [25, Sec. 2.11] for details).

### Determination of $A_{i,l}$ and $B_{i,l}$

In order to identify the contents of  $A_{i,l}$  and  $B_{i,l}$ ,  $l = 0, 1, \dots$ , we introduce matrices  $R$  describing changes in the service index,  $X_k$  describing changes in the contents of queue  $k \neq j$ , where  $j$  is the queue in service, and  $Y_{i,j}$  and  $\tilde{Y}_{i,j}$  describing changes in the contents of queue  $j$  for level  $n_i > 0$  and  $n_i = 0$  respectively.

We define  $R(j, j')$  as the probability that, after service of queue  $j$ , the server moves to queue  $j'$ :

$$R(j, j') = \begin{cases} (1 - q(j))P(j, j'), & \text{if } j \neq j', \\ q(j), & \text{if } j = j'. \end{cases}$$

The matrix  $X_k$  is such that  $X_k(n_k, n'_k)$  is the probability that the contents of queue  $k$  go from  $n_k$  to  $n'_k$ , with  $n_k, n'_k \in \{0, \dots, B\}$  given queue  $k$  is not in service. We have:

$$X_k = \begin{pmatrix} x_k(0) & x_k(1) & \dots & x_k(B-1) & 1 - \sum_{l < B} x_k(l) \\ 0 & x_k(0) & \dots & x_k(B-2) & 1 - \sum_{l < B-1} x_k(l) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & x_k(0) & 1 - x_k(0) \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix},$$

where  $x_k(l)$  is the probability that  $l$  packets arrive to queue  $k$ .

We define the matrix  $Y_{i,j}$ , such that  $Y_{i,j}(n_j, n'_j)$  is the probability that, in level  $n_i > 0$ , queue  $j$  goes from  $n_j$  to  $n'_j$ , with  $n_j \in \{1, \dots, B\}$  and  $n'_j \in \{0, \dots, B\}$ . Here,  $n_j \geq 1$  because service of queue  $j$  implies that queue  $j$  is non-empty.

For queue  $j$ , we make the approximation assumption that its contents go from 'B or more' to  $B-1$  with a fixed probability denoted by  $\tilde{\zeta}_{i,j}$  and  $\zeta_{i,j}$  for level  $n_i = 0$  and  $n_i > 0$  respectively. The rationale behind this level dependence is that, due to correlation between queue lengths, if queue  $i$  is empty it is more likely that the contents of queue  $j$  are small, and hence it is also more likely that queue  $j$  goes from 'B or more' to  $B-1$ . The parameters  $\zeta_{i,j}$  and  $\tilde{\zeta}_{i,j}$ ,  $j \neq i$ , are called the truncation parameters of queue  $i$ . The values of these parameters are determined iteratively, as will be described in more detail at the end of this section. For now, we simply assume that they have a certain value.

It follows that  $Y_{i,j}$  is given by

$$Y_{i,j} = \begin{pmatrix} x_j(0) & \dots & x_j(B-2) & x_j(B-1) & 1 - \sum_l x_j(l) \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & x_j(0) & x_j(1) & 1 - \sum_l x_j(l) \\ 0 & \dots & 0 & \zeta_{i,j}x_j(0) & 1 - \zeta_{i,j}x_j(0) \end{pmatrix},$$

where the last element of each row is such that the row sums to 1.

Likewise,  $\tilde{Y}_{i,j}(n_j, n'_j)$  is the probability that the contents of queue  $j$  go from  $n_j$  to  $n'_j$ , for level  $n_i = 0$ . The matrix  $\tilde{Y}_{i,j}$  is identical to  $Y_{i,j}$ , except that  $\tilde{\zeta}_{i,j}$  is substituted for  $\zeta_{i,j}$ .

The matrix  $A_{i,l}$  describes changes where queue  $i$  goes up by  $l-1$  levels. This happens if there are either  $l-1$  arrivals and no service completion, or  $l$  arrivals and a service completion. It follows that the probability of going from phase  $\omega = (j, n_1, \dots, n_{i-1}, n_{i+1}, \dots, n_N)$  to phase  $\omega' = (j', n'_1, \dots, n'_{i-1}, n'_{i+1}, \dots, n'_N)$  and going up by  $l-1$  levels is given by

$$A_{i,l}(\omega, \omega') = x_i(l-1)R(j, j')Y_{i,j}(n_j, n'_j) \prod_{\substack{k \neq j \\ k \neq i}} X_k(n_k, n'_k),$$

for  $j \neq i$ , with  $x_i(-1) := 0$ , and

$$A_{i,l}(\omega, \omega') = x_i(l)R(j, j') \prod_{k \neq j} X_k(n_k, n'_k)$$

for  $j = i$ . Here, the probability that the service index changes from  $j$  to  $j'$  is given by  $R(j, j')$ . The probability that the contents of the queues that are not in service change from  $n_k$  to  $n'_k$  are given by  $X_k(n_k, n'_k)$ . Finally, the probability that the contents of the queue in service change from  $n_j$  to  $n'_j$  is  $Y_{i,j}(n_j, n'_j)$ .

The matrices  $B_{i,l}$  are slightly different. First, if all queues are empty, the server remains at the same queue. Second, there is never a service completion at queue  $i$ , because queue  $i$  is empty. Third, the transitions of the queue in service, queue  $j$ , are not given by  $Y_{i,j}$  but by  $\tilde{Y}_{i,j}$ . We obtain:

$$B_{i,l}(\omega, \omega') = x_i(l) \prod_{k \neq i} X_k(0, n'_k),$$

if  $\omega = (j, 0, \dots, 0)$ , and

$$B_{i,l}(\omega, \omega') = x_i(l)R(j, j')\tilde{Y}_{i,j}(n_j, n'_j) \prod_{\substack{k \neq j \\ k \neq i}} X_k(n_k, n'_k),$$

otherwise.

We have specified all the matrices needed to determine the equilibrium distribution of the SMC of queue  $i$ . To compute the equilibrium distribution, we use the software tools of Bini et al. [3, 4]. For more details on how to compute the equilibrium distribution, the reader is referred to [23].

### Determination of $\zeta_{i,j}$ and $\tilde{\zeta}_{i,j}$

In the description of the SMCs, we introduced the truncation parameters of queue  $i$ ,  $\zeta_{i,j}$  and  $\tilde{\zeta}_{i,j}$ . We use an iterative procedure to compute the values of these parameters and we denote their value in step  $m$  of the iteration by  $\zeta_{i,j}^{(m)}$  and  $\tilde{\zeta}_{i,j}^{(m)}$ .

In each step of the iterative procedure, we determine new values for the truncation parameters of one queue by means of the most recently computed equilibrium distributions of

the other queues: We first determine the values of the truncation parameters of queue 1 and compute the equilibrium distribution of queue 1 with these values. We determine the truncation parameters of queue 2 using the newly computed equilibrium distribution of queue 1, as well as the previous equilibrium distributions of queues 3, 4, ...,  $N$ . We then compute the new equilibrium distribution of queue 2, and, with that, new values for the truncation parameters of queue 3, and so on, until the values of all truncation parameters have converged.

The truncation parameters of queue  $i$  describe the probability that, without arrivals to queue  $j$ , the contents of queue  $j \neq i$  go from 'B or more' to  $B - 1$ , given that a service completion occurs at queue  $j$ . Without arrivals and with a service completion, a transition from 'B or more' to  $B - 1$  occurs if the contents of queue  $j$  are in fact equal to  $B$ . If we denote the length of queue  $j$  by  $Q_j$ , and the service index by  $S$ , the probability of such a transition is thus given by

$$\mathbb{P}(Q_j = B | Q_j \geq B, Q_i > 0, S = j), \quad j \neq i, \quad (3a)$$

for level  $n_i > 0$  and

$$\mathbb{P}(Q_j = B | Q_j \geq B, Q_i = 0, S = j), \quad j \neq i, \quad (3b)$$

for level  $n_i = 0$ .

We denote the equilibrium distribution of the SMC of queue  $i$  in step  $m$  of the iteration by  $\pi_i^{(m)}(j, n_1, \dots, n_N)$ . By evaluating the conditional probabilities in (3a) and (3b) and substituting the corresponding most recently computed equilibrium probabilities of the SMC of queue  $j$ , we obtain

$$\zeta_{i,j}^{(m)} = \begin{cases} \frac{\sum_{n_j=B, n_i \geq 1} \pi_j^{(m)}(j, n_1, \dots, n_N)}{\sum_{n_j \geq B, n_i \geq 1} \pi_j^{(m)}(j, n_1, \dots, n_N)}, & \text{for } j < i, \\ \frac{\sum_{n_j=B, n_i \geq 1} \pi_j^{(m-1)}(j, n_1, \dots, n_N)}{\sum_{n_j \geq B, n_i \geq 1} \pi_j^{(m-1)}(j, n_1, \dots, n_N)}, & \text{for } j > i, \end{cases}, \quad (4a)$$

and

$$\tilde{\zeta}_{i,j}^{(m)} = \begin{cases} \frac{\sum_{n_j=B, n_i=0} \pi_j^{(m)}(j, n_1, \dots, n_N)}{\sum_{n_j \geq B, n_i=0} \pi_j^{(m)}(j, n_1, \dots, n_N)}, & \text{for } j < i, \\ \frac{\sum_{n_j=B, n_i=0} \pi_j^{(m-1)}(j, n_1, \dots, n_N)}{\sum_{n_j \geq B, n_i=0} \pi_j^{(m-1)}(j, n_1, \dots, n_N)}, & \text{for } j > i, \end{cases} \quad (4b)$$

as the values for the truncation parameters of queue  $i$  in step  $m \geq 1$ . All sums are taken over  $0 \leq n_k \leq B$ , for  $k \neq i, j$ .

For  $i < j$ , Equations (4a) and (4b) express  $\zeta_{i,j}^{(1)}$  and  $\tilde{\zeta}_{i,j}^{(1)}$  in terms of  $\pi_j^{(0)}(\cdot)$ , which is not defined. In this case, we choose

$$\zeta_{i,j}^{(1)} = \tilde{\zeta}_{i,j}^{(1)} = 1, \quad \text{for } j > i. \quad (4c)$$

Numerical experiments indicate that setting the initial values of the truncation parameters to 1 is important. Small values of the truncation parameters indicate that there are many packets in the other queues requiring service. If the initial truncation parameters are (much) smaller than 1, the SMC of queue  $i$  might even be transient, though in reality the polling system is recurrent. If the SMC of queue  $i$  is transient, its equilibrium distribution cannot be determined and new values for the truncation parameters cannot be found.

If the initial values are chosen equal to 1, the value of the truncation parameters converged in all our examples, though we cannot formally prove convergence. Choosing the initial parameters less than 1 and closer to their limiting values might speed up convergence, at the risk that no convergence occurs at all if the initial values are too small.

Our algorithm is summarised below:

ALGORITHM 4.3.

0. Fix  $\varepsilon$  (for instance  $\varepsilon = 10^{-8}$ ) and set  $m = 1$ .

1. Determine  $\Psi_i$  and  $\tilde{\Psi}_i$  for  $i = 1, \dots, N$ .

2. For  $i = 1, \dots, N$ :

(a) Use Equation (4a), (4b), or (4c) to determine  $\zeta_{i,j}^{(m)}$  and  $\tilde{\zeta}_{i,j}^{(m)}$ .

(b) Determine  $A_{i,l}$  and  $B_{i,l}$  with  $\zeta_{i,j} = \zeta_{i,j}^{(m)}$  and  $\tilde{\zeta}_{i,j} = \tilde{\zeta}_{i,j}^{(m)}$ .

(c) Determine the equilibrium distribution of queue  $i$ ,  $\pi_i^{(m)}(\cdot)$ .

3. Stop if  $m \geq 2$  and  $\max_{i,j} \{|\zeta_{i,j}^{(m)} - \zeta_{i,j}^{(m-1)}|, |\tilde{\zeta}_{i,j}^{(m)} - \tilde{\zeta}_{i,j}^{(m-1)}|\} < \varepsilon$ . Otherwise, set  $m = m + 1$  and repeat step 2.

Finally, after convergence of the truncation parameters, the approximation of the marginal distribution of the length of queue  $i$  follows from the equilibrium distribution of the SMC of queue  $i$ . Namely,  $\mathbb{P}(Q_i = l)$  is approximated by the sum of the equilibrium probabilities of all phases at level  $n_i = l$ .

## 5. NUMERICAL RESULTS

In this section, we study the accuracy of our approximation for a single case with 4 queues, 1-limited service and cyclic routing. The batch sizes are governed by a Poisson distribution with parameter  $\rho_i$ , where  $(\rho_1, \dots, \rho_4) = (0.1, 0.2, 0.3, 0.4)\rho$ . The approximated queue length distributions are compared with simulation outcomes in Table 1 for  $\rho = 0.5$ ,  $\rho = 0.7$ , and  $\rho = 0.9$ .

For loads 0.5 and 0.7 the approximation is very accurate even with  $B$  as small as 2. The differences in individual probabilities occur only in the 4th decimal or later. For  $\rho = 0.9$  and  $B = 2$ , the approximation is less accurate, but still quite good. If  $B = 3$ , the approximation again becomes more accurate.

The probabilities obtained from simulation are the average probabilities of 10 simulation runs of  $25 \cdot 10^6$  time slots each. Furthermore, each probability in the table is rounded according to the value of the standard deviation  $\sigma$  in the simulation outcomes of that probability. If the first four digits of  $\sigma/\sqrt{10}$  are zero, but the fifth is nonzero, then 4 digits are shown, etc.

It will be convenient to express the error of the approximation as a single number for each queue. To this end, we use the total variation distance between the approximated and simulated queue length distribution, which, for queue  $i$ , is defined as

$$d_i = \sum_{k=0}^{\infty} |q_{i,k} - \hat{q}_{i,k}|, \quad (5)$$

**Table 1: Queue length distributions**

$\rho = 0.5$		$\mathbb{P}(Q_i = 0)$	$\mathbb{P}(Q_i = 1)$	$\mathbb{P}(Q_i = 2)$	$\mathbb{P}(Q_i = 3)$	$\mathbb{P}(Q_i = 4)$	$\mathbb{P}(Q_i = 5)$	$\mathbb{P}(Q_i = 6)$
$i = 1$	$B = 2$	0.9362	0.0612	0.00249	0.000092	0.000003	0.000000	0.000000
	Sim	0.9362	0.0612	0.00249	0.000093	0.000004	0.000000	0.000000
$i = 2$	$B = 2$	0.8709	0.1179	0.01029	0.00084	0.000071	0.000006	0.000001
	Sim	0.8709	0.1179	0.01028	0.00084	0.000071	0.000006	0.000001
$i = 3$	$B = 2$	0.8055	0.1681	0.0230	0.00295	0.00040	0.00006	0.000008
	Sim	0.8054	0.1681	0.0230	0.00297	0.00040	0.00006	0.000008
$i = 4$	$B = 2$	0.7412	0.2109	0.0395	0.0069	0.00126	0.00024	0.000047
	Sim	0.7411	0.2109	0.0395	0.0069	0.00127	0.00024	0.000049

$\rho = 0.7$		$\mathbb{P}(Q_i = 0)$	$\mathbb{P}(Q_i = 1)$	$\mathbb{P}(Q_i = 2)$	$\mathbb{P}(Q_i = 3)$	$\mathbb{P}(Q_i = 4)$	$\mathbb{P}(Q_i = 5)$	$\mathbb{P}(Q_i = 6)$
$i = 1$	$B = 2$	0.8953	0.0969	0.00722	0.00052	0.000039	0.000003	0.000000
	Sim	0.8952	0.0970	0.00725	0.00052	0.000039	0.000003	0.000000
$i = 2$	$B = 2$	0.7840	0.1798	0.0300	0.00504	0.00090	0.00017	0.000032
	Sim	0.7838	0.1800	0.0301	0.00509	0.00091	0.00017	0.000034
$i = 3$	$B = 2$	0.6724	0.2401	0.0636	0.0171	0.00484	0.00142	0.00043
	Sim	0.6720	0.2400	0.0637	0.0172	0.00493	0.00148	0.00046
$i = 4$	$B = 2$	0.5661	0.2756	0.0994	0.0361	0.0137	0.0054	0.00218
	Sim	0.5655	0.2754	0.0994	0.0362	0.0139	0.0056	0.00228

$\rho = 0.9$		$\mathbb{P}(Q_i = 0)$	$\mathbb{P}(Q_i = 1)$	$\mathbb{P}(Q_i = 2)$	$\mathbb{P}(Q_i = 3)$	$\mathbb{P}(Q_i = 4)$	$\mathbb{P}(Q_i = 5)$	$\mathbb{P}(Q_i = 6)$
$i = 1$	$B = 2$	0.8302	0.1474	0.0195	0.00253	0.00034	0.000047	0.000007
	$B = 3$	0.8298	0.1477	0.0196	0.00255	0.00034	0.000048	0.000007
	Sim	0.8297	0.1477	0.0196	0.00256	0.00035	0.000049	0.000007
$i = 2$	$B = 2$	0.6362	0.2479	0.0779	0.0251	0.0084	0.0029	0.00100
	$B = 3$	0.6350	0.2480	0.0783	0.0254	0.0086	0.0030	0.00105
	Sim	0.6345	0.2481	0.0784	0.0256	0.0087	0.0031	0.00109
$i = 3$	$B = 2$	0.439	0.2669	0.1350	0.0710	0.0387	0.0215	0.0121
	$B = 3$	0.437	0.2654	0.1346	0.0713	0.0393	0.0222	0.0127
	Sim	0.435	0.2644	0.1343	0.0713	0.0397	0.0226	0.0131
$i = 4$	$B = 2$	0.267	0.2159	0.1438	0.0993	0.0712	0.0520	0.0384
	$B = 3$	0.265	0.2129	0.1413	0.0974	0.0701	0.0517	0.0387
	Sim	0.263	0.2109	0.1394	0.0959	0.0690	0.0510	0.0383

where  $q_{i,k}$  and  $\hat{q}_{i,k}$  denote simulation and approximation values for  $\mathbb{P}(Q_i = k)$ . The values of  $d_i$  can be found in Table 2.

**Table 2: Total variation distances**

	Queue 1	Queue 2	Queue 3	Queue 4
$\rho = 0.5$	0.0000	0.0001	0.0001	0.0002
$\rho = 0.7$	0.0003	0.0005	0.0009	0.0015
$\rho = 0.9 (B = 2)$	0.0011	0.0035	0.0138	0.0410
$\rho = 0.9 (B = 3)$	0.0002	0.0011	0.0055	0.0191

Because we can approximate the mean waiting times using Little’s law, we can also compare our approximation with the existing mean waiting time approximations of Boxma and Meister [9] and Groenendijk and Levy [16]. We do so in Table 3. Again, our approximation is very accurate. In all cases, except queue 4 and  $\rho = 0.9$ , our approximation is more accurate than the Boxma-Meister and Levy-Groenendijk approximations. Both Boxma and Meister [9, Rem. 5.2] and Groenendijk and Levy [16, Sec. IV] give suggestions to improve their approximations for high loads. These suggestions were taken into account in Table 3.

Because the state spaces of the SMCs are exponential in  $N$ , it is clear that our approximation can only be applied to polling systems with few queues. For our application, networks on chips, however, this does not pose a problem since the switches there typically have only few queues, usually 4 or 5. If  $N = 4$  and  $B = 2$ , the running time of our approximation for one value of  $\rho$  and all four queues is only about 2 or 3 seconds. If  $B = 3$ , the running time increases to roughly 50 seconds. For comparison, the ten simulation runs that give the level of accuracy presented in this section require about 15 to 20 minutes in total, per value of  $\rho$ .

**Table 3: Mean waiting times**

$\rho = 0.5$	Queue 1	Queue 2	Queue 3	Queue 4
$B = 2$	0.329	0.413	0.499	0.586
Sim.	0.329	0.413	0.500	0.587
BM	0.346	0.423	0.500	0.577
LG	0.299	0.396	0.498	0.604

$\rho = 0.7$	Queue 1	Queue 2	Queue 3	Queue 4
$B = 2$	0.615	0.854	1.138	1.462
Sim.	0.618	0.858	1.145	1.475
BM	0.709	0.938	1.167	1.395
LG	0.539	0.830	1.152	1.503

$\rho = 0.9$	Queue 1	Queue 2	Queue 3	Queue 4
$B = 2$	1.172	1.98	3.50	6.46
$B = 3$	1.179	2.01	3.59	6.84
Sim.	1.181	2.02	3.66	7.21
BM	1.590	2.71	4.70	5.97
LG	1.168	1.94	3.04	7.71

## 6. LARGE-SCALE NUMERICAL STUDY

In this section, we perform a numerical experiment on a larger scale to study the accuracy of our approximation. We vary the following six characteristics of the polling system over a number of values: The total load  $\rho$ , the number of queues  $N$ , the service discipline, the level of symmetry, the arrival processes, and the routing matrix. We consider all possible combinations, i.e., every possible load is combined with every possible value of  $N$ , every possible service discipline, and so on. An overview of the values of these characteristics can be found in Table 4. In total, the experiment comprises  $5 \cdot 4 \cdot 4 \cdot 3 \cdot 3 \cdot 2 = 1440$  polling systems.

**Table 4: The numerical experiment.**

$\rho$	0.5	0.6	0.7	0.8	0.9
$N$	2	3	4	5	
$q$	0	0.3	0.7	1	
Symmetry	Symmetric	Asymmetric	Very asymm.		
Arrival process	Bernoulli	Poisson	Geometric		
Routing	Cyclic	Uniform			

We assume that  $q(i) = q$ , i.e., within one polling system considered in the experiment, all queues have the same service discipline. We further assume that  $\rho_i = \nu_i \rho$ , where  $\sum_i \nu_i = 1$ . The constants  $\nu_i$  are determined by the level of symmetry, which is either symmetric, asymmetric, or very asymmetric: In the symmetric case, every queue gets a fraction  $1/N$  of the load;  $\nu_i = 1/N$ . In the asymmetric case,  $(\nu_1, \dots, \nu_N) \sim (1, 2, \dots, N)$ , where  $\sim$  means ‘proportional to’. For example, if  $N = 2$ ,  $(\nu_1, \nu_2) = (1/3, 2/3)$ , if  $N = 3$ ,  $(\nu_1, \nu_2, \nu_3) = (1/6, 2/6, 3/6)$ , and so on. In the very asymmetric case each queue receives a fraction 0.1 of the load, except queue  $N$  which gets the rest.

The distribution of the number of packets arriving to queue  $i$  is from the same family for all  $i$ , but its mean  $\rho_i$  depends on  $i$ . The distribution can be Bernoulli with parameter  $\rho_i$ , Poisson with parameter  $\rho_i$ , or geometric with parameter  $1/(1 + \rho_i)$ . Here the parameter is  $1/(1 + \rho_i)$  so that the mean number of packets arriving each time slot is  $\rho_i$ . Note that the geometric distribution we use has positive mass at 0.

Finally, the routing discipline used is either cyclic or uniform. With uniform routing, if the server leaves a certain queue, it selects one of the other queues at random, each with the same probability, i.e.,  $P(i, j) = 1/(N - 1)$  for  $i \neq j$ .

We compare our approximation with  $B = 2$  with simulation outcomes. For every queue of a polling system, the error of the approximation is defined as the total variation distance between the approximated and simulated queue length distributions, cf. (5).

Table 5 shows the characteristics of the 100 systems with the largest average error (averaged over the queues). This table should be read as follows: The 100 systems all have  $\rho = 0.9$ , 26 have two queues, 27 have three, etc.

Table 5 reveals that the first and foremost cause of a high error is a high load. Related to this observation, Table 5 shows that a higher variance of the arrival process also leads to a higher error. The variances of the Bernoulli, Poisson, and Geometric arrival processes are given by  $\rho_i(1 - \rho_i)$ ,  $\rho_i$ , and  $\rho_i(1 + \rho_i)$  respectively. The cause of this error is the truncation of queue lengths. As the load or the variance of the arrival processes increases, queue lengths increase as well, and hence the error induced by truncation.

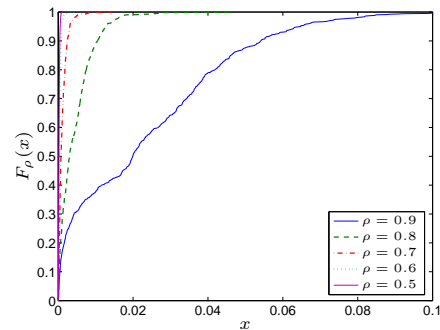
**Table 5: Systems with the highest errors**

$\rho$	0.5	0.6	0.7	0.8	<b>0.9</b>
	0	0	0	0	<b>100</b>
$N$	2	3	4	5	
	26	27	25	22	
$q(i)$	0	0.3	0.7	1	
	12	13	21	<b>54</b>	
Symmetry	<b>Symmetric</b>	Asymmetric	Very asymm.		
	<b>59</b>	25	16		
Arrival process	Bernoulli	<b>Poisson</b>	<b>Geometric</b>		
	6	<b>40</b>	<b>54</b>		
Routing	Cyclic	Uniform			
	53	47			

Second, Table 5 indicates that the error increases if exhaustive service is used. The pivotal assumption of our approximation is that the contents of the truncated queues go from ‘ $B$  or more’ to  $B - 1$  with a fixed probability. With exhaustive service, the time spent serving one queue consecutively is larger than with 1-limited service. As a result, when the server finally starts serving one of the truncated queues, this queue will have  $B$  or more packets with a larger probability. The error of our approximation is therefore larger if exhaustive service is used.

Third, Table 5 suggests that the average error is largest if the system is symmetric. Indeed we found that, on the whole, the average error decreases as the system becomes more asymmetric. Consider, as an extreme example, a polling system where one queue receives almost the entire load, and other queues receive only a very small fraction. In this case, the approximation is indeed very accurate, since the lightly loaded queues hardly ever have ‘ $B$  or more’ packets.

In Fig. 1, we show  $F_\rho(x)$ , which is defined as the fraction of queues in systems with load  $\rho$ , whose error, defined by (5), is less than  $x$ . Fig. 1 indeed shows that the most important cause of a high error is a high load.



**Figure 1: The function  $F_\rho(x)$ .**

Likewise, Fig. 2 shows the function  $G_q(x)$ , which is defined as the fraction of queues in systems with service discipline parameter  $q$ , whose error is less than  $x$ . Fig. 2 clearly illustrates that the error is the largest for exhaustive service, whereas there is only a small difference between 1-limited, Bernoulli with parameter 0.3, and Bernoulli with parameter 0.7.

We conclude that for  $B = 2$  the error is certainly acceptable for loads up to 0.7 or 0.8, depending on for instance the service discipline and the variance in the arrival process.

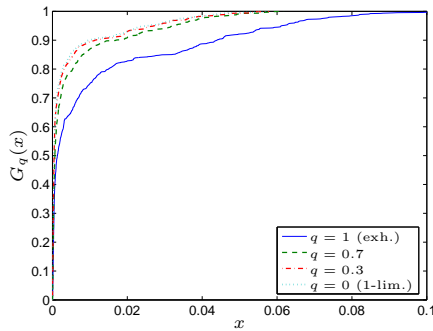


Figure 2: The function  $G_q(x)$ .

For higher loads,  $B$  should be increased to reduce the error if computationally feasible.

## 7. CONCLUSIONS

We devised an algorithm that approximates the marginal queue length distributions in a discrete-time polling system with Bernoulli service and Markovian routing. The key step in this approximation is the translation of the queue length process to a structured Markov chain, where the contents of one queue are stored in the level and truncated contents of the other queues in the phase, i.e., we store whether there are  $0, 1, \dots, B-1$ , or ' $B$  or more' packets in the other queues. We furthermore use an iterative procedure to determine the probability that the contents of the other queues go from ' $B$  or more' to  $B-1$ .

As  $B$  tends to infinity, the approximation becomes exact. However, it was shown through numerical experiments that, with  $B = 2$ , the approximation is already very accurate in general. Furthermore, it was shown that the accuracy of the approximation decreases if the load increases, and if the variance in the arrival process increases. In addition to this, it turned out that the accuracy decreases if  $q(i)$  - the probability that after a service completion at queue  $i$ , the server serves queue  $i$  again - increases. This, for example, implies that the approximation is more accurate for the 1-limited service discipline ( $q(i) = 0$ ) than for the exhaustive service discipline ( $q(i) = 1$ ). In cases where the inaccuracy reaches an unacceptable level,  $B$  can be increased further at the cost of a higher running time.

The memory and computation requirements of the approximation are exponential in the number of queues. This clearly entails that the approximation is only practical for polling systems with few queues. For our application, networks on chips, however, this does not pose a problem since switches in these networks typically have only few queues.

Throughout this paper, we assumed that all buffers are infinite. For finite buffers, the same procedure involving queue truncation and iterative determination of the truncation parameters can still be applied. Another interesting extension of the approximation described here is an extension to polling systems with Markovian arrival processes. The extension of our approximation to such systems is straightforward, but the computation time and memory requirements would increase.

## 8. REFERENCES

- [1] P. Beekhuizen, T. Denteneer, and J. Resing. Reduction of a polling network to a single node. *Queueing Systems*, 58(4):303–319, April 2008.
- [2] P. Beekhuizen, T. Denteneer, and J. Resing. End-to-end delays in polling tree networks. In *Proc. of VALUETOOLS*, 2008, Athens, Greece.
- [3] D. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: Algorithms. In *Proc. of SMCTools*, 2006, Pisa, Italy.
- [4] D. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: Software tools. In *Proc. of SMCTools*, 2006, Pisa, Italy.
- [5] J. Blanc. A numerical approach to cyclic-service queueing models. *Queueing Systems*, 6:173–188, Dec. 1990.
- [6] J. Blanc. The power-series algorithm applied to cyclic polling systems. *Stochastic Models*, 7:527–545, 1991.
- [7] J. Blanc. An algorithmic solution of polling models with limited service disciplines. *IEEE Transactions on Communications*, 40:1152–1155, 1992.
- [8] J. Blanc. Performance evaluation of polling systems by means of the power-series algorithm. *Annals of Operations Research*, 35:155–186, June 1992.
- [9] O. Boxma and B. Meister. Waiting-time approximations in multi-queue systems with cyclic service. *Performance Evaluation*, 7:59–70, 1987.
- [10] O. Boxma and J. Weststrate. Waiting times in polling systems with Markovian server routing. In G. Stiege and J. Lie, editors, *Messung, Modellierung und Bewertung von Rechensystemen und Netzen*, pages 89–105. Springer-Verlag Berlin, 1989.
- [11] P. Buchholz. Adaptive decomposition and approximation for the analysis of stochastic Petri nets. *Performance Evaluation*, 56:23–52, 2004.
- [12] W. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proc. of the Design Automation Conference*, pages 684–689, 2001.
- [13] W. Feng, M. Kowada, and K. Adachi. A two-queue model with Bernoulli service schedule and switching times. *Queueing Systems*, 30:405–434, Dec. 1998.
- [14] I. Frigui and A. Alfa. Analysis of a discrete time table polling system with MAP input and time-limited service discipline. *Telecommunication Systems*, 12:51–77, 1999.
- [15] S. Fuhrmann and Y. Wang. Analysis of cyclic service systems with limited service: Bounds and approximations. *Performance Evaluation*, 9:35–54, Nov. 1988.
- [16] W. Groenendijk and H. Levy. Performance analysis of transaction driven computer systems via queueing analysis of polling models. *IEEE Transactions on Computers*, 41(4):455–466, April 1992.
- [17] J. Keilson and L. Servi. Oscillating random walk models for GI/G/1 vacation systems with Bernoulli schedules. *Journal of Applied Probability*, 23:790–802, Sept. 1986.
- [18] L. Kleinrock and H. Levy. The analysis of random polling systems. *Operations Research*, 36:716–732, Oct. 1988.
- [19] G. Latouche and V. Ramaswami. *Introduction to*

*Matrix Analytic Methods in Stochastic Modeling.* SIAM, Philadelphia, 1999.

- [20] D.-S. Lee and B. Sengupta. An approximate analysis of a cyclic server queue with limited service and reservations. *Queueing Systems*, 11:153–178, Mar. 1992.
- [21] K. Leung. Cyclic-service systems with probabilistically-limited service. *IEEE Selected Areas in Communications*, 9:185–193, 1991.
- [22] A.S. Miner, G. Ciardo, and S. Donatelli. Using the exact state space of a Markov model to compute approximate stationary measures. In *Proc. of SIGMETRICS*, 2000, Santa Clara, US.
- [23] M. Neuts. *Structured Stochastic Matrices of M/G/1 Type and their Applications*, volume 5 of *Probability: Pure and Applied*. Marcel Dekker Inc., New York, 1989.
- [24] J. Resing. Polling systems and multitype branching processes. *Queueing Systems*, 13:409–426, 1993.
- [25] S. Resnick. *Adventures in Stochastic Processes*. Birkhäuser, Boston, 2nd edition, 1994.
- [26] M. Srinivasan. An approximation for mean waiting times in cyclic server systems with nonexhaustive service. *Performance Evaluation*, 9:17–33, 1988.
- [27] M. Srinivasan. Non-deterministic polling systems. Technical Report 88-18, The University of Michigan, December 1988.
- [28] T. Tedijanto. Exact results for the cyclic-service queue with a Bernoulli schedule. *Performance Evaluation*, 11:107–115, July 1990.
- [29] M. van Vuuren and E. Winands. Iterative approximation of k-limited polling systems. *Queueing Systems*, 55(3):161–178, March 2007.
- [30] J. Weststrate. *Analysis and Optimization of Polling Models*. PhD thesis, Tilburg University, 1992.

## APPENDIX

### A. KRONECKER PRODUCTS

In this section, we derive alternative expressions for the matrices  $A_{i,l}$  and  $B_{i,l}$  using Kronecker products. The matrices  $\Psi_i$  and  $\tilde{\Psi}_i$  describing the server movements from an empty queue to a non-empty queue are easily determined algorithmically so they will not be dealt with here.

For an  $n_A \times m_A$  matrix  $A$  and an  $n_B \times m_B$  matrix  $B$ , the Kronecker product is an  $n_A n_B \times m_A m_B$  matrix defined as

$$A \otimes B = \begin{pmatrix} A(1,1)B & A(1,2)B & \dots & A(1,m_A)B \\ \vdots & \vdots & \ddots & \vdots \\ A(n_A,1)B & A(n_A,2)B & \dots & A(n_A,m_A)B \end{pmatrix}.$$

Kronecker products are especially useful in describing Markov chain transitions on multidimensional sets; if transitions on a set  $V \times W$  can be decomposed into independent transitions on  $V$  and  $W$ , then the transition probability matrix on  $V \times W$  is  $P \otimes Q$ , where  $P$  and  $Q$  are the transition probability matrices on  $V$  and  $W$  respectively, provided  $V \times W$  is ordered lexicographically.

If, more importantly, transitions on  $W$  only depend on the current state in  $V$  but do not depend on the destination state in  $V$  (i.e., a transition from  $(v, w)$  to  $(v', w')$  occurs with probability  $p(v, v')q_v(w, w')$ , where  $p(v, v')$  is the probability

of going from  $v$  to  $v'$ , and  $q_v(w, w')$  that of going from  $w$  to  $w'$  given  $v$ ), then the transition probability matrix on  $V \times W$  is given by

$$\begin{pmatrix} p_1 \otimes Q_1 \\ p_2 \otimes Q_2 \\ \vdots \\ p_m \otimes Q_m \end{pmatrix}. \quad (6)$$

Here,  $V = \{1, \dots, m\}$ ,  $p_v$  is a vector with elements  $p(v, v')$ , and  $Q_v$  is a matrix with elements  $q_v(w, w')$ .

In order to give the alternative expressions for  $A_{i,l}$  and  $B_{i,l}$ , we assume that  $\Phi_i$  and  $\tilde{\Phi}_i$  are ordered lexicographically. The matrix  $A_{i,l}$  describes changes where queue  $i$  goes up by  $l-1$  levels. This happens if there are either  $l-1$  arrivals and no service completion, or  $l$  arrivals and a service completion:

$$A_{i,l} = x_i(l-1)D_{i,0} + x_i(l)D_{i,1}, \quad \text{for } l = 0, 1, \dots,$$

where  $x_i(-1) := 0$ , and  $D_{i,0}$  and  $D_{i,1}$  are the transition probability matrices within phases, without and with a service completion at queue  $i$ , respectively.

The matrix  $D_{i,0}$  describes phase transitions *without* a service completion at queue  $i$ . Such transitions are caused by changes in the service index, arrivals and a service completion at queue  $j$ , and arrivals at the other queues. The changes in the contents of the queues only depend on the service index through the service completion. In particular, the changes in queue contents do not depend on the queue the server moves to in the next time slot, so we obtain, conform (6):

$$D_{i,0} = \begin{pmatrix} r_1 \otimes Y_{i,1} \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ r_2 \otimes X_1 \otimes Y_{i,2} \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ \vdots \\ r_{i-1} \otimes X_1 \otimes X_2 \otimes \dots \otimes Y_{i,i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ 0 \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ r_{i+1} \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes Y_{i,i+1} \otimes \dots \otimes X_N \\ \vdots \\ r_N \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes Y_{i,N} \end{pmatrix},$$

where 0 is the length  $N$  zero vector.

In the first block row queue 1 is currently in service. The changes in the service index are thus given by  $r_1$ , the transitions of queue 1 by  $Y_{i,1}$ , and the transitions of the other queues by  $X_k$ . In the second block row, queue 2 is in service so a  $Y_{i,2}$  appears, and so on. The zero vector in the  $i$ th block row represents the fact that a transition *without* a service completion cannot occur if the service index is  $i$ .

The matrix  $D_{i,1}$  describes phase transitions *with* a service completion at queue  $i$ . Using similar arguments as in the derivation of  $D_{i,0}$ , and observing that a transition *with* a service completion at queue  $i$  can only occur if the service index is equal to  $i$ , we obtain:

$$D_{i,1} = \begin{pmatrix} 0 \otimes Y_{i,1} \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ 0 \otimes X_1 \otimes Y_{i,2} \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ \vdots \\ 0 \otimes X_1 \otimes X_2 \otimes \dots \otimes Y_{i,i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ r_i \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ 0 \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes Y_{i,i+1} \otimes \dots \otimes X_N \\ \vdots \\ 0 \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes Y_{i,N} \end{pmatrix}.$$

The matrices  $B_{i,l}$  can be expressed as

$$B_{i,l} = x_i(l)\tilde{D}_i, \quad \text{for } l = 0, 1, \dots,$$

where  $\tilde{D}_i$  is the phase transition probability matrix for level  $n_i = 0$ . Note that, because we are in level  $n_i = 0$ , there is never a service completion at queue  $i$ .

The matrix  $\tilde{D}_i$  is given by:

$$\tilde{D}_i = \begin{pmatrix} e_1 \otimes x_1 \otimes x_2 \otimes \dots \otimes x_{i-1} \otimes x_{i+1} \otimes \dots \otimes x_N \\ r_1 \otimes \tilde{Y}_{i,1} \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ e_2 \otimes x_1 \otimes x_2 \otimes \dots \otimes x_{i-1} \otimes x_{i+1} \otimes \dots \otimes x_N \\ r_2 \otimes X_1 \otimes \tilde{Y}_{i,2} \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ \vdots \\ e_{i-1} \otimes x_1 \otimes x_2 \otimes \dots \otimes x_{i-1} \otimes x_{i+1} \otimes \dots \otimes x_N \\ r_{i-1} \otimes X_1 \otimes X_2 \otimes \dots \otimes \tilde{Y}_{i,i-1} \otimes X_{i+1} \otimes \dots \otimes X_N \\ e_i \otimes x_1 \otimes x_2 \otimes \dots \otimes x_{i-1} \otimes x_{i+1} \otimes \dots \otimes x_N \\ e_{i+1} \otimes x_1 \otimes x_2 \otimes \dots \otimes x_{i-1} \otimes x_{i+1} \otimes \dots \otimes x_N \\ r_{i+1} \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes \tilde{Y}_{i,i+1} \otimes \dots \otimes X_N \\ \vdots \\ e_N \otimes x_1 \otimes x_2 \otimes \dots \otimes x_{i-1} \otimes x_{i+1} \otimes \dots \otimes x_N \\ r_N \otimes X_1 \otimes X_2 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes \tilde{Y}_{i,N} \end{pmatrix},$$

where  $e_j$  is the  $j$ th row of the  $N \times N$  identity matrix and  $x_k = (x_k(0), \dots, x_k(B-1), \sum_{l \geq B} x_k(l))$ .

The first block row (which is a block consisting of one row) represents transitions from state  $(1, 0, \dots, 0)$ , which is the state where all queues are empty and the server is waiting at queue 1. In this state, the position of the server does not change, i.e., its changes are given by  $e_1$ . The changes in the contents of the other queues are given by the independent arrivals to those queues, i.e., by  $x_k$ .

The second block row describes transitions from states where queue 1 is non-empty and in service. Like before, changes in the service index are given by  $r_1$ , and changes in all queues except queue 1 by  $X_k$ . Changes in queue 1 are given by  $\tilde{Y}_{i,1}$  because the process is in level  $n_i = 0$ . By following similar reasonings for the other block rows, the expression for  $\tilde{D}_i$  follows.

REMARK A.1. For the computation of the equilibrium distribution of an SMC of  $M/G/1$  type, one has to compute the  $G$ -matrix. Several algorithms to do so are available (see, for instance, [3]). In our numerical studies we found that Functional Iterations with the default  $U$ -based scheme is the fastest. In particular, this iterative method allows one to start with an initial  $G$ -matrix. Starting with the  $G$ -matrix as computed in the previous iteration speeds up the computations.

REMARK A.2. If the arrival processes are ordinary (i.e., non-batch) Bernoulli arrival processes, we have an SMC of the Quasi-Birth-Death (QBD) type instead of  $M/G/1$  (see, e.g., [19]). In this case, an  $R$ -matrix has to be computed, which considerably simplifies actual implementations.

REMARK A.3. The use of Kronecker based matrix representation in conjunction with fixed point methods has also been used in [11] and [22] in the analysis of large finite Markov chains.