The waiting time distribution of a type k customer in a discrete-time MMAP[K]/PH[K]/c (c=1,2) queue using QBDs

B. Van Houdt[1], C. Blondia

University of Antwerp
Department of Mathematics and Computer Science
Performance Analysis of Telecommunication Systems Research Group
Universiteitsplein, 1, B-2610 Antwerp - Belgium
{benny.vanhoudt,chris.blondia}@ua.ac.be

Abstract

This paper presents an improved method to calculate the delay distribution of a type k customer in a first-come-first-serve (FCFS) discrete-time queueing system with multiple types of customers, where each type has different service requirements, and c servers, with $c = 1, 2$ (the MMAP[K]/PH[K]/c queue). The first algorithms to compute this delay distribution, using the GI/M/1 paradigm, were presented in [9, 10]. The two most limiting properties of these algorithms are: (i) the computation of the rate matrix R related to the GI/M/1 type Markov chain, (ii) the amount of memory needed to store the transition matrices A_l and B_l. In this paper we demonstrate that each of the three GI/M/1 type Markov chains used to develop the algorithms in [9, 10] can be reduced to a QBD with a block size which is only marginally larger than that of its corresponding GI/M/1 type Markov chain. As a results, the two major limiting factors of each of these algorithms are drastically reduced to computing the G matrix of the QBD and storing the 6 matrices that characterize the QBD. Moreover, these algorithms are easier to implement, especially for the system with $c = 2$ servers. We also include some numerical examples that further demonstrate the reduction in computational resources.

[1]B. Van Houdt is a postdoctoral Fellow of the FWO Flanders.

1 Introduction

In this paper we study a class of queues with c (= 1 or 2) servers, correlated interarrival times and multiple types of customers, where each type has different service requirements, known as the discrete time MMAP[K]/PH[K]/c queue. The MMAP[K] arrival process, introduced in [3], is a Markovian arrival process that generates customers of K different types and is a generalization of the batch Markovian arrival process (BMAP). Its potential applications to telecommunications, manufacturing and service industries have been demonstrated extensively in [3, 2].

Queues with MMAP[K] input, e.g., MMAP[K]/G[K]/1 queues, with a first-come-first-served (FCFS) service discipline have been studied in [2, 1, 7, 8]. Within these papers, explicit formulas for the Laplace Stieltjes Transform (LST) of the actual waiting times of a customer of type k were obtained. In [9], we developed two algorithms that allowed us to calculate the delay distribution of a type k customer in a discrete-time MMAP[K]/PH[K]/1 queue, by constructing a GI/M/1 type Markov chain (MC). The first algorithm in [9] applies to MMAP[K] arrival processes that do not allow batch arrivals, the second provides a (limited) solution for MMAP[K] processes with batch arrivals. The basic idea behind the algorithms in [9] was generalized to the MMAP[K]/PH[K]/2 queue, where the MMAP[K] arrival process does not allow batch arrivals [10]. The two most limiting properties of each of these 3 algorithms are: (i) the computation of the rate matrix R related to the GI/M/1 type Markov chain, (ii) the amount of memory needed to store the transition matrices A_l and B_l. In this paper we demonstrate that the GI/M/1 type Markov chains constructed by each of these algorithms can be reduced to a QBD with a block size which is only marginally larger than that of the corresponding GI/M/1 type Markov chain. As a results, the two major limiting factors of each of these algorithms are drastically reduced to computing the G matrix of the QBD and storing the 6 matrices that characterize the QBD.

The paper is structured as follows. Section 2 presents the MMAP[K]/PH[K]/c queue in some detail, Sections 3 and 4 discuss the algorithms for the single server case without and with batch arrivals, respectively. The system with 2 servers is dealt with in Section 5. Finally, Section 6 presents some numerical examples to demonstrate the magnitude of the reduction in computational

resources.

2 The discrete-time MMAP[K]/PH[K]/c queue

The arrival process of the queueing system of interest is a discrete time Markov arrival process with marked transitions (MMAP[K]). Customers are distinguished into K different types. An MMAP[K] that does not allow batch arrivals to occur, is characterized by a set of $m \times m$ matrices $\{D_k \mid 0 \leq k \leq K\}$, with m a positive integer. The $(j_1, j_2)^{th}$ entry of the matrix D_k, for $k > 0$, represents the probability that a customer of type k arrives and the underlying Markov chain makes a transition from state j_1 to state j_2. The matrix D_0 covers the case when there are no arrivals. If batch arrivals are allowed, an MMAP[K] is characterized by a set of $m \times m$ matrices D_C where C is a arbitrary string of integers between 1 and K (denote $|C|$ as the length of C), that is, $C = c_1 \dots c_{|C|}$ with $1 \leq c_l \leq K$ and $1 \leq l \leq |C|$. The $(j_1, j_2)^{th}$ entry of the matrix D_C, for C different from the empty string $\emptyset$, represents the probability that $|C|$ customers arrive, the type of the i-th customer equals the i-th element of the string C, and the underlying Markov chain makes a transition from state j_1 to state j_2. The matrix D, defined as

$$D = \sum_C D_C,$$

represents the stochastic $m \times m$ transition matrix of the underlying Markov chain of the arrival process. Let θ be the stationary probability vector of D, that is, $\theta D = \theta$ and $\theta e = 1$, where e is a column vector with all entries equal to one. The stationary arrival rate of type k customers is given by $\lambda_k = \theta \sum_C N(C,k) D_C e$, where $N(C,k)$ counts the number of occurrences of k in C, hence, $\lambda_k = \theta D_k e$ if there are no batch arrivals.

The service times of type k customers have a common phase-type distribution function with a matrix representation (m_k, α_k, T_k), where m_k is a positive integer, α_k is a $1 \times m_k$ nonnegative stochastic vector and T_k is an $m_k \times m_k$ substochastic matrix. Let $t_k = e - T_k e$, then the mean service time of a type k customer equals $1/\mu_k = \alpha_k (I - T_k)^{-1} e$. Define $m_{ser} = \sum_{k=1}^{K} m_k$, the

$m_{ser} \times m_{ser}$ matrix T_{ser} and the $m_{ser} \times 1$ vector t_{ser} as

$$T_{ser} = \begin{bmatrix} T_1 & 0 & \dots & 0 \\ 0 & T_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & T_K \end{bmatrix}, \qquad t_{ser} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_K \end{bmatrix},$$

and let $m_{tot} = m_{ser}m$. The customers are served, by c servers, with $c = 1$ or 2, according to a first-come-first-serve (FCFS) service discipline.

3 The MMAP[K]/PH[K]/1: no batch arrivals

In [9, Section 2], we constructed an MC of the GI/M/1 type that allowed us to obtain the delay distribution of a type k customer from its steady state probabilities. For reasons of completeness, the transition matrix P corresponding to this MC is presented next:

$$P = \begin{bmatrix} B_1 & B_0 & 0 & 0 & 0 & \dots \\ B_2 & A_1 & A_0 & 0 & 0 & \dots \\ B_3 & A_2 & A_1 & A_0 & 0 & \dots \\ B_4 & A_3 & A_2 & A_1 & A_0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{1}$$

where A_l are $m_{tot} \times m_{tot}$ matrices, $B_l, l > 1$, are $m_{tot} \times m$ matrices, B_1 is an $m \times m$ matrix and B_0 is an $m \times m_{tot}$ matrix. In order to express the matrices A_l and B_l, for $l \geq 0$, we defined the following $m \times m_{tot}$ matrix L:

$$L = [(\alpha_1 \otimes D_1), (\alpha_2 \otimes D_2), \dots, \ (\alpha_K \otimes D_K)] \, .$$

Based on the probabilistic interpretation of the matrices A_l and B_l we found:

$$\begin{aligned} A_0 &= T_{ser} \otimes I_m, \\ A_l &= t_{ser} \otimes (D_0)^{l-1} L, \\ B_0 &= L, \\ B_1 &= D_0, \\ B_l &= t_{ser} \otimes (D_0)^{l-1}, \end{aligned}$$

where $\otimes$ denotes the Kronecker product between matrices and I_m the $m \times m$ unity matrix. Remark, in [9] we denoted t_{ser} as T^0_{ser}. Each of the states of this Markov chain has a physical interpretation: State j of level zero corresponds to the situation in which the queue and the server are empty, while the current state of the MMAP[K] is j. State (k, s, j) of level i of the MC correspond to the situation in which there is a customer of type k in service, that arrived i time units ago, while the service process is currently in phase s and the MMAP[K] arrival process is in state j at time $n - i + 1$, where n is the current time instant.

Next, we introduce a method to obtain the steady state probability vector π of P, by constructing a QBD characterized by the matrices A^*_0, A^*_1 and A^*_2 each of dimension $m_{tot} + m$, avoiding the computation of the matrix R related to the GI/M/1 type Markov chain.

The idea is the following: Suppose that the GI/M/1 type MC makes a transition from level i to level $i-l$ with $l > 1$, then we could split this transition into l transitions that each decreases the level by one at a time. Ramaswami [6] already developed such a procedure for a general GI/M/1 type MC, however, applying his method leads to a level dependent QBD where the dimensions of the blocks are multiples of m_{tot} (the dimension of the blocks of level $i > 0$ is im_{tot}). For the Markov chain defined by the transition matrix P, one can do much better by noticing the particular form of A_l and B_l, for $l > 1$:

$$A_l = t_{ser} \otimes D_0^{l-1} B_0, \tag{2}$$

$$B_l = t_{ser} \otimes D_0^{l-1}, \tag{3}$$

where $B_0 = L$ and t_{ser} are independent of l. Using the geometric nature of A_l and B_l, we can construct the following QBD:

$$P^* = \begin{bmatrix} B^*_1 & B^*_0 & 0 & 0 & 0 & \dots \\ B^*_2 & A^*_1 & A^*_0 & 0 & 0 & \dots \\ 0 & A^*_2 & A^*_1 & A^*_0 & 0 & \dots \\ 0 & 0 & A^*_2 & A^*_1 & A^*_0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{4}$$

where A^*_l are $(m + m_{tot}) \times (m + m_{tot})$ matrices, B^*_2 is an $(m + m_{tot}) \times m$ matrix, B^*_1 an $m \times m$ matrix and B^*_0 an $m \times (m + m_{tot})$. The matrices A^*_l,

for $l = 0, 1$ and 2, are defined as follows:

$$A_0^* = \begin{bmatrix} 0 & 0 \\ 0 & A_0 \end{bmatrix}, \tag{5}$$

$$A_1^* = \begin{bmatrix} 0 & B_0 \\ 0 & A_1 \end{bmatrix}, \tag{6}$$

$$A_2^* = \begin{bmatrix} D_0 & 0 \\ t_{ser} \otimes D_0 & 0 \end{bmatrix}. \tag{7}$$

While the matrices B_l^*, for $l = 0, 1$ and 2, are equal to

$$B_0^* = [0 \;\; B_0], \tag{8}$$

$$B_1^* = B_1, \tag{9}$$

$$B_2^* = \begin{bmatrix} D_0 \\ t_{ser} \otimes D_0 \end{bmatrix}, \tag{10}$$

where A_0, A_1, B_0 and B_1 were defined at the start of this section and 0 is a zero matrix with the appropriate dimension. The states of level 0 are denoted as $\{j \mid 1 \leq j \leq m\}$, while the states of level $i > 0$ are written as $\{j \mid 1 \leq j \leq m\} \cup \{(k, s, j) \mid 1 \leq k \leq K, 1 \leq s \leq m_k, 1 \leq j \leq m\}$. The states of level zero and the states of the form (k, s, j) of level i have the same physical interpretation as the states of the GI/M/1 type MC characterized by the transition matrix P. The states of the form j of level i, for $i > 0$, have no real physical interpretation and are therefore called *artificial* states. It is easy to see that if we observe this QBD only at the time instants when it visits a state other than an artificial state, we obtain the GI/M/1 type MC with transition matrix P.

The key in finding the steady state probability vector $\pi^* = (\pi_0^*, \pi_1^*, \ldots)$ of P^*, where π_0^* and π_i^*, for $i > 0$, are $1 \times m$ and $1 \times (m + m_{tot})$ vectors, respectively, is to solve the following equation:

$$G = A_0^* + A_1^* G + A_2^* G^2. \tag{11}$$

We propose to use the Cyclic Reduction algorithm to compute G [5]. This algorithm is very easy to implement, requires a low amount of memory, converges quadratically and is numerically stable. Having found G, one computes R as $A_0^*(I - A_1^* - A_0^* G)^{-1}$ [4]. The steady state probability vectors π_i^*

are then found as:

$$[\pi_0^* \ \pi_1^*] = [\pi_0^* \ \pi_1^*] \begin{bmatrix} B_1^* & B_0^* \\ B_2^* & A_1^* + RA_2^* \end{bmatrix}, \tag{12}$$

$$\pi_i^* = \pi_{i-1}^* R, \tag{13}$$

where $i > 1$, π_0^* and π_1^* are normalized as $\pi_0^* e + \pi_1^*(I - R)^{-1} e = 1$.

Let $\pi = (\pi_0, \pi_1, \ldots)$ be the steady state vector of P, where π_0 and π_i, for $i > 0$, are $1 \times m$ and $1 \times m_{tot}$ vectors, respectively. Denote π_i^*, for $i > 0$, as $[\pi_i^*(m), \ \pi_i^*(m_{tot})]$, with $\pi_i^*(s)$ a $1 \times s$ vector. Then, $\pi_0 = \pi_0^*/(1-c)$ and $\pi_i = \pi_i^*(m_{tot})/(1-c)$, for $i > 0$. The constant c equals $\sum_{i>0} \pi_i^*(m)e$. Using π, it is easy to obtain the probability $P[d_k = i]$ that a type k customer experiences a delay[2] of i time units [9]:

$$P[d_k = i] = \sum_{s=1}^{m_k} \frac{(t_k)_s}{\lambda_k} \sum_{j=1}^{m} \pi_i(k, s, j),$$

for $i \geq 1$, with λ_k the arrival rate of the type k customers, while $(t_k)_s$ represents the s-th component of the column vector t_k. Notice, $P[d_k = 0] = 0$, because a customer spends at least one time unit in the server.

4 The MMAP[K]/PH[K]/1: batch arrivals

The algorithm developed in [9] to calculate the delay distribution of a type k customer in an MMAP[K]/PH[K]/1 queue with batch arrivals is composed of two steps: First, we create a new MMAP[K] arrival process, characterized by the $am \times am$ matrices $\tilde{D}_k$ for $k = 0, 1, \ldots, K$, that does not allow batch arrivals (for some $a > 1$, with m the number of states of the original MMAP[K]), see [9, Section 4]. Afterwards, we create a GI/M/1 type MC using the new MMAP[K] as indicated in [9, Section 3] and calculate the delay distributions from its steady state.

[2]Here, the delay is defined as the waiting time plus the service time. One can obtain the waiting time distribution from d_k by means of a deconvolution, as the service time is independent from the waiting time.

In this section, we demonstrate that one can easily reduce the GI/M/1 type MC developed in [9, Section 3] to a QBD by adding m artificial states as we did in the previous section. Let $\tilde{P}$ be the transition matrix of the GI/M/1 type MC:

$$\tilde{P} = \begin{bmatrix} \tilde{B}_1 & \tilde{B}_0 & 0 & 0 & 0 & \dots \\ \tilde{B}_2 & \tilde{A}_1 & \tilde{A}_0 & 0 & 0 & \dots \\ \tilde{B}_3 & \tilde{A}_2 & \tilde{A}_1 & \tilde{A}_0 & 0 & \dots \\ \tilde{B}_4 & \tilde{A}_3 & \tilde{A}_2 & \tilde{A}_1 & \tilde{A}_0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{14}$$

where $\tilde{A}_l$ are $am_{tot} \times am_{tot}$ matrices, $\tilde{B}_l, l > 1$, are $am_{tot} \times m$ matrices, $\tilde{B}_1$ is an $m \times m$ matrix and $\tilde{B}_0$ is an $m \times am_{tot}$ matrix. The matrices $\tilde{A}_l$ and $\tilde{B}_l$, for $l > 1$, can be written as

$$\tilde{A}_l = \begin{bmatrix} t_{ser} \\ 0 \end{bmatrix} \otimes D_0^{l-1} \tilde{B}_0, \tag{15}$$

$$\tilde{B}_l = \begin{bmatrix} t_{ser} \\ 0 \end{bmatrix} \otimes D_0^{l-1}, \tag{16}$$

where D_0 is the $m \times m$ matrix of the original MMAP[K], $\tilde{B}_0$ is determined by the first m rows of $\tilde{D}_k$ and by the vectors α_k, for $k = 1, \dots, K$, and 0 is a zero matrix of the appropriate dimension (that is, $(a-1)m_{ser} \times 1$). For further details on $\tilde{A}_l$ and $\tilde{B}_l$, for $l \geq 0$, we refer to [9, Section 3].

Similar to what we did in the previous section, we construct a QBD, characterized by its transition matrix $\tilde{P}^*$, as

$$\tilde{P}^* = \begin{bmatrix} \tilde{B}_1^* & \tilde{B}_0^* & 0 & 0 & 0 & \dots \\ \tilde{B}_2^* & \tilde{A}_1^* & \tilde{A}_0^* & 0 & 0 & \dots \\ 0 & \tilde{A}_2^* & \tilde{A}_1^* & \tilde{A}_0^* & 0 & \dots \\ 0 & 0 & \tilde{A}_2^* & \tilde{A}_1^* & \tilde{A}_0^* & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{17}$$

where $\tilde{A}_l^*$ are $(m + am_{tot}) \times (m + am_{tot})$ matrices, $\tilde{B}_2^*$ is a $(m + am_{tot}) \times m$ matrix, $\tilde{B}_1^*$ an $m \times m$ matrix and $\tilde{B}_0^*$ an $m \times (m + am_{tot})$. The matrices $\tilde{A}_l^*$,

for $l = 0, 1$ and 2, are defined as follows:

$$\tilde{A}_0^* = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{A}_0 \end{bmatrix}, \tag{18}$$

$$\tilde{A}_1^* = \begin{bmatrix} 0 & \tilde{B}_0 \\ 0 & \tilde{A}_1 \end{bmatrix}, \tag{19}$$

$$\tilde{A}_2^* = \begin{bmatrix} D_0 & 0 \\ \begin{bmatrix} t_{ser} \\ 0 \end{bmatrix} \otimes D_0 & 0 \end{bmatrix}. \tag{20}$$

While, the matrices $\tilde{B}_l^*$, for $l = 0, 1$ and 2, are equal to

$$\tilde{B}_0^* = \begin{bmatrix} 0 & \tilde{B}_0 \end{bmatrix}, \tag{21}$$

$$\tilde{B}_1^* = \tilde{B}_1, \tag{22}$$

$$\tilde{B}_2^* = \begin{bmatrix} D_0 \\ \begin{bmatrix} t_{ser} \\ 0 \end{bmatrix} \otimes D_0 \end{bmatrix}, \tag{23}$$

where 0 is a zero matrix with the appropriate dimension. The remaining steps of the procedure are analogue to the previous section.

5 The MMAP[K]/PH[K]/2: no batch arrivals

In [10] we developed a GI/M/1 type MC that allowed us to calculate the waiting time distribution of a type k customer in a MMAP[K]/PH[K]/2 queue without batch arrivals. The age of the oldest customer waiting in the waiting room was reflected by the level of this MC. A short description of this MC that is used further on, is given in Appendix A. It is possible to reduce this MC to a QBD by introducing some artificial states to each level. However, there exists a different procedure that uses an alternative QBD with even smaller blocks as follows. In [10] we indicated that one could also construct a GI/M/1 type MC by defining the level of the MC as the minimal age of the customers in service (the chain is at level 0 if there are less than two

customers in service). The block sizes of both these GI/M/1 type MCs are the same (except for level 0), as is the rate at which the transition matrices A_l decrease to zero, therefore, their performance is similar. However, when making a reduction to a QBD, we found a QBD with smaller blocks using the minimal age approach.

We start by describing the GI/M/1 type MC, followed by the reduction to the QBD. Recall, $m_{ser} = \sum_{k=1}^{K} m_k$ and let $m_{sv} = m_{ser}^2 + \sum_{k=1}^{K} m_k^2$ and $\bar{m}_{tot} = m_{sv}m/2$. Notice, in [10] we use a slightly different notation. Consider an MC with an infinite number of states labeled $1, 2, \ldots$. The set of states $\{1, \ldots, m\}$ is referred to as level 0_a of the MC, the states $\{m+1, \ldots, m+m_{ser}m\}$ as level 0_b, finally, the states $\{m + m_{ser}m + (i-1)\bar{m}_{tot} + 1, \ldots, m + m_{ser}m + i\bar{m}_{tot}\}$ are referred to as level i, $i > 0$, of the MC. To simplify the notation further on, define m_{zero} as $m(1 + m_{ser})$. The states of level 0_a are labeled as $1 \leq j \leq m$, those of level 0_b as (k_1, s_1, j), where $1 \leq k_1 \leq K$, $1 \leq s_1 \leq m_{k_1}$ and $1 \leq j \leq m$, whereas the states of level $i > 0$ are labeled as (k_1, k_2, s_1, s_2, j) where $1 \leq k_2 \leq k_1 \leq K$, $1 \leq s_i \leq m_{k_i}$ (for $i = 1, 2$) and $1 \leq j \leq m$.

The states of the MC have the following interpretation. Assume that we observe the system at an arbitrary time instant n. Then, the MC is in state j of level 0_a at time n if the waiting room and both servers are empty at time n, while the state of the MMAP[K] at time n equals j. The MC is in state (k_1, s_1, j) of level 0_b at time n if the waiting room is empty and one server is busy with a customer of type k_1, the service of which is in phase s_1, and the state of the MMAP[K] at time n equals j. Finally, if both servers are busy (with a type k_1 and a type k_2 customer in phase s_1 and s_2, respectively), then the MC is in state (k_1, k_2, s_1, s_2, j) of level $i > 0$ if the minimal age, i.e., youngest, customer in service arrived at time $n - i$ and the MMAP[K] arrival process is in state j at time $n - i + 1$. In conclusion, the level of the MC at time n corresponds to the "age" at time n of the youngest customer in service (level $0 = 0_a \cup 0_b$ corresponds to at least one empty server)[3].

Using arguments similar to [10] one easily finds that $\bar{P}$, the transition matrix,

[3]Notice, we do not know which customer occupies which server. We can easily add this information to the MC by stating that the type k_i customer occupies server i. However, this would imply that the condition $k_1 \geq k_2$ is lost, hence, the number of states that are part of each level increases (up to a factor 2). Adding this additional info would however simplify the description of the transition probabilities.

has the following form:

$$\bar{P} = \begin{bmatrix} \bar{B}_1 & \bar{B}_0 & 0 & 0 & 0 & \dots \\ \bar{B}_2 & \bar{A}_1 & \bar{A}_0 & 0 & 0 & \dots \\ \bar{B}_3 & \bar{A}_2 & \bar{A}_1 & \bar{A}_0 & 0 & \dots \\ \bar{B}_4 & \bar{A}_3 & \bar{A}_2 & \bar{A}_1 & \bar{A}_0 & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{24}$$

where $\bar{A}_l$ are $\bar{m}_{tot} \times \bar{m}_{tot}$ matrices, $\bar{B}_l, l > 1$, are $\bar{m}_{tot} \times m_{zero}$ matrices, $\bar{B}_1$ is an $m_{zero} \times m_{zero}$ matrix and $\bar{B}_0$ is an $m_{zero} \times \bar{m}_{tot}$ matrix. Instead of describing all these matrices in detail, we restrict ourselves to $\bar{A}_0, \bar{A}_1, \bar{B}_0$ and $\bar{B}_1$, an expression for the other $\bar{B}_l$ and $\bar{A}_l$ matrices can be obtained using similar arguments. We shall not derive them here as they are not required to construct the corresponding QBD.

The matrix $\bar{B}_1$ describes the transitions from level $0 = 0_a \cup 0_b$ to 0. Using the probabilistic interpretation one finds

$$\bar{B}_1 = \begin{bmatrix} D_0 & L \\ t_{ser} \otimes D_0 & T_{ser} \otimes D_0 + t_{ser} \otimes L \end{bmatrix}, \tag{25}$$

where $L = [(\alpha_1 \otimes D_1), \dots, (\alpha_K \otimes D_K)]$. The matrix $\bar{B}_0$ covers the transitions from level 0 to level 1. Clearly, if the MC is in a state of level 0_a, it cannot be in a state of level 1 at the next time instant (because this requires an arrival of two customers at once). Thus, the first m rows of $\bar{B}_0$ are zero. Define $\bar{B}_0(k_1; k_1', k_2')$ as the $mm_{k_1} \times mm_{k_1'} m_{k_2'}$ submatrix of $\bar{B}_0$ that describes the transitions from the states of level 0_b labeled $(k_1, ., .)$ to the states of level 1 labeled $(k_1', k_2', ., ., .)$. Then,

$$\begin{aligned} \bar{B}_0(k_1; k_1', k_2') &= 1_{\{k_1 = k_1'\}} (T_{k_1} \otimes \alpha_{k_2'} \otimes D_{k_2'}) \\ &\quad + 1_{\{k_1' > k_1 = k_2'\}} (\alpha_{k_1'} \otimes T_{k_1} \otimes D_{k_1'}). \end{aligned} \tag{26}$$

Indeed, in order to go from level 0_b to level 1, a new customer has to arrive, while the one in service (of type k_1) needs to remain in service. The distinction between the two terms is caused by the fact that $k_1' \geq k_2'$.

To facilitate the description of $\bar{A}_0$ and $\bar{A}_1$ we define the $mm_{k_1} m_{k_2} \times mm_{k_1'} m_{k_2'}$ submatrices $\bar{A}_0(k_1, k_2;\ k_1', k_2')$ and $\bar{A}_1(k_1, k_2; k_1', k_2')$ in the obvious way. The matrix $\bar{A}_0$ covers the transitions from level i to $i + 1$. Such transitions occur

only if no service completion occurs (otherwise a new arrival, that necessarily arrived after time $n-i$, enters the server, meaning that its age is at most i at time $n+1$). Hence,

$$\bar{A}_0(k_1,k_2;k_1',k_2') = 1_{\{k_1=k_1' \cap\ k_2=k_2'\}}(T_{k_1}\otimes T_{k_2}\otimes I_m). \tag{27}$$

To get a transition from level i to level i, we need one service completion (while the other type k_l customer remains in service) and an arrival (of a type k_l' customer) at time $n-i+1$, therefore,

$$\begin{aligned}\bar{A}_1(k_1,k_2;k_1',k_2') = {} & 1_{\{k_1'>k_1=k_2'\}}\left(\alpha_{k_1'}\otimes T_{k_1}\otimes t_{k_2}\otimes D_{k_1'}\right)+\\ & 1_{\{k_1=k_1'\}}\left(T_{k_1}\otimes\alpha_{k_2'}\otimes t_{k_2}\otimes D_{k_2'}\right)+1_{\{k_2=k_2'\}}\left(t_{k_1}\otimes\alpha_{k_1'}\otimes T_{k_2}\otimes D_{k_1'}\right)+\\ & 1_{\{k_2'<k_2=k_1'\}}\left(t_{k_1}\otimes T_{k_2}\otimes\alpha_{k_2'}\otimes D_{k_2'}\right),\end{aligned} \tag{28}$$

where the first term corresponds to having $k_l=k_1$ and $k_l'>k_1$, the second to $k_l=k_1$ and $k_l'\le k_1$, the third to $k_l=k_2$ and $k_l'\ge k_2$ and the last to $k_l=k_2$ and $k_l'<k_2$.

Next, we construct a QBD, characterized by $\bar{P}^*$, by adding $m_{zero}=m+m_{ser}m$ artificial states to the levels $i>0$ as follows:

$$\bar{P}^* = \begin{bmatrix} \bar{B}_1^* & \bar{B}_0^* & 0 & 0 & 0 & \dots\\ \bar{B}_2^* & \bar{A}_1^* & \bar{A}_0^* & 0 & 0 & \dots\\ 0 & \bar{A}_2^* & \bar{A}_1^* & \bar{A}_0^* & 0 & \dots\\ 0 & 0 & \bar{A}_2^* & \bar{A}_1^* & \bar{A}_0^* & \dots\\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{29}$$

where $\bar{A}_l^*$ are $(m_{zero}+\bar{m}_{tot})\times(m_{zero}+\bar{m}_{tot})$ matrices, $\bar{B}_2^*$ is an $(m_{zero}+\bar{m}_{tot})\times m_{zero}$ matrix, $\bar{B}_1^*$ an $m_{zero}\times m_{zero}$ matrix and $\bar{B}_0^*$ an $m_{zero}\times(m_{zero}+\bar{m}_{tot})$. Intuitively, being in the m artificial states described in the previous 2 sections meant that we were tracking the time instant where the next arrival occurs. In this section we have $m+m_{ser}m$ artificial states. One could say, intuitively, that the first m states, labeled j (with $1\le j\le m$), correspond to tracking the first of two arrivals (both the servers are free), whereas the other $m_{ser}m$ states, labeled (k_1,s_1,j) (with $1\le k_1\le K$, $1\le s_1\le m_{k_1}$ and $1\le j\le m$), correspond to looking for an arrival knowing that one server holds a type k_1 customer in phase s_1. Similar to the previous sections, $\bar{B}_0^*$, $\bar{B}_1^*$, $\bar{A}_0^*$ and $\bar{A}_1^*$ are

defined as

$$\bar{B}_0^* = \begin{bmatrix} 0 & \bar{B}_0 \end{bmatrix}, \tag{30}$$

$$\bar{B}_1^* = \bar{B}_1, \tag{31}$$

$$\bar{A}_0^* = \begin{bmatrix} 0 & 0 \\ 0 & \bar{A}_0 \end{bmatrix} \tag{32}$$

$$\bar{A}_1^* = \begin{bmatrix} 0 & \bar{C}_0 \\ 0 & \bar{A}_1 \end{bmatrix}. \tag{33}$$

The matrix $\bar{C}_0$ is nearly identical to $\bar{B}_0$, that is, its first m rows are also zero, while its remaining $m_{ser}m$ rows are described by Equation (26) if we replace the matrices T_{k_1} by $I_{m_{k_1}}$. The matrices $\bar{A}_2^*$ and $\bar{B}_2^*$ are defined as

$$\bar{A}_2^* = \begin{bmatrix} S_1 & 0 \\ S_2 & 0 \end{bmatrix}, \tag{34}$$

$$\bar{B}_2^* = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}, \tag{35}$$

where S_1 and S_2 are an $m_{zero} \times m_{zero}$ and an $\bar{m}_{tot} \times m_{zero}$ matrix, respectively. These two matrices are discussed next:

$$S_1 = \begin{bmatrix} D_0 & L \\ 0 & I_{m_{ser}} \otimes D_0 \end{bmatrix}, \tag{36}$$

where I_l is an $l \times l$ unity matrix. Define the following matrices $S_2(k_1, k_2)$ and $S_2(k_1, k_2; k_1')$ as the submatrices of S_2 that correspond to the transitions from the states labeled as $(k_1, k_2, ., ., .)$ to the artificial states labeled $(.)$ and $(k_1, ., .)$, respectively. Then,

$$S_2(k_1, k_2) = t_{k_1} \otimes t_{k_2} \otimes D_0, \tag{37}$$

$$\begin{aligned} S_2(k_1, k_2; k_1') = {} & 1_{\{k_1 = k_1'\}}(T_{k_1} \otimes t_{k_2} \otimes D_0) + 1_{\{k_2 = k_1'\}}(t_{k_1} \otimes T_{k_2} \otimes D_0) \\ & + t_{k_1} \otimes t_{k_2} \otimes (\alpha_{k_1'} \otimes D_{k_1'}). \end{aligned} \tag{38}$$

This concludes the description of $\bar{P}^*$.

When we observe this QBD only at the time instants when the MC is not visiting an artificial state, we obtain the GI/M/1 type Markov chain defined

by $\bar{P}$. This follows from the fact that the matrices $\bar{A}_l$ and $\bar{B}_l$, for $l > 1$, can be written as

$$\bar{A}_l = S_2 S_1^{l-2} \bar{C}_0, \tag{39}$$

$$\bar{B}_l = S_2 S_1^{l-2}. \tag{40}$$

This identity can be verified using the probabilistic interpretation of the matrices $\bar{A}_l$ and $\bar{B}_l$. Notice, these equations are similar to those in the previous two sections if we denote the upperleft and lowerleft blocks of the matrices A_2^* and $\tilde{A}_2^*$ as S_1 and S_2 (and B_0 and $\tilde{B}_0$ as C_0), respectively.

Analogue to what we did at the end of Section 3, we can compute the steady state vector of $\bar{P}$, denoted as $\bar{\pi} = (\bar{\pi}_0, \bar{\pi}_1, \bar{\pi}_2, \ldots)$, from the steady state vector of $\bar{P}^*$. In order to compute the waiting time distribution of a type k customer, we proceed as follows. First, we compute the steady state vector $\bar{\pi}^+$ (defined in Appendix A) from $\bar{\pi}$. The physical interpretation of the states of both the GI/M/1 type MCs involved (see Section 5 and Appendix A), leads to

$$[\bar{\pi}_{0_a}^+, \bar{\pi}_{0_b}^+] = \bar{\pi}_0 \tag{41}$$

$$\bar{\pi}_{0_c}^+ = \sum_{j \geq 1} \bar{\pi}_j \begin{bmatrix} D_0^{j-1} & 0 & \ldots & 0 \\ 0 & D_0^{j-1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & D_0^{j-1} \end{bmatrix}, \tag{42}$$

$$\bar{\pi}_i^+ = \sum_{j \geq i+1} \bar{\pi}_j \begin{bmatrix} D_0^{j-i-1}E & 0 & \ldots & 0 \\ 0 & D_0^{j-i-1}E & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & D_0^{j-i-1}E \end{bmatrix}, \tag{43}$$

where $i > 0$ and E equals the diagonal matrix $diag((D - D_0)e)$. Afterwards, we can simply apply the formula presented in [10] to obtain the probability $P[w_k^+ = i]$ that a type k customer experiences a waiting time of i time units

from the steady state vector $\bar{\pi}^+$:

$$P[w_k^+ = i] = \frac{1}{\lambda_k}\left[\sum_{k_1,k_2,s_1,s_2,j} \frac{\bar{\pi}_i^+(k_1,k_2,s_1,s_2,j)(D_k e)_j}{((D-D_0)e)_j}\left((t_{k_1})_{s_1} + (t_{k_2})_{s_2} - (t_{k_1})_{s_1}(t_{k_2})_{s_2}\right) + \sum_{i'>i}\sum_{k_1,k_2,s_2,s_2,j} \bar{\pi}_{i'}^+(k_1,k_2,s_1,s_2,j)\frac{((D-D_0)D_0^{i'-i-1}D_k e)_j}{((D-D_0)e)_j}(t_{k_1})_{s_1}(t_{k_2})_{s_2}\right] \quad (44)$$

for $i \geq 1$, with λ_k the arrival rate of the type k customers, while $(t_{k_i})_{s_i}$ represents the s_i-th component of the column vector t_{k_i}.

6 Numerical Examples

6.1 Single server, with batch arrivals

The main computational cost of determining the delay distribution for a type k customer in an MMAP[K]/PH[K]/1 queue (with or without batch arrivals) is to find the rate matrix R. This is true whether we use the GI/M/1 (see [9]) or QBD approach[4]. For the GI/M/1 approach one generally considers the matrices $\tilde{A}_l$ and $\tilde{B}_l$ as zero for $l > n_\epsilon$, where n_ϵ is the minimal integer such that $\sum_{l=n_\epsilon+1}^{\infty} \tilde{A}_l e < 10^{-14}e$. The value of n_ϵ thus depends on the rate at which $\tilde{A}_l$ decreases to zero, which in turn depends on the rate that D_0^l decreases to zero. In many cases n_ϵ easily reaches a value of several hundreds, and in some cases—e.g., if the MMAP[K] has a state where no arrivals occur and the mean sojourn time in this state is 1000 slots—even many thousands. The storage of the matrices $\tilde{A}_l$ and $\tilde{B}_l$ requires $16n_\epsilon(am_{tot})^2$ bytes, while the time complexity is at least $2n_\epsilon(am_{tot})^3$ flops per iteration (depending on the algorithm used to compute R). If we compare this with the QBD approach we find that the matrices $\tilde{A}_l^*$ and $\tilde{B}_l^*$ occupy $\approx 48(am_{tot}+m)^2$ bytes, while an a single iteration requires $4(am_{tot})^3$ flops, using the classic iterative scheme by Neuts, or $14(am_{tot})^3$ flops, using the Cyclic Reduction Algorithm [5]. Moreover, for

[4]Of course, both systems have a different R matrix.

the QBD a variety of quadratically converging algorithms exist to determine R (via G), e.g., the Cyclic Reduction Algorithm.

Let us demonstrate the difference between both approaches once more by repeating the numerical example presented in [9]. Consider a single server queue with three correlated input sources A, B and C; their customers are referred to as type one, two and three. Each source generates zero or one customer during a time instant. The superposition of these three correlated sources is assumed to be a 3 state MMAP[3]. The three states are traversed one by one and the sojourn time in each state is geometrically distributed with a mean of 1000 time units. While in state one, source A generates a customer with probability 1/5, source C with probability 1/100, while source B is silent. In state two, source A and C generate a customer with probability 1/100, while source B generates a customer with probability 1/28. Finally, in state three, source B generates a customer with probability 1/100, source C with probability 1/20, while source A is silent. Given that we are in state $1 \leq j \leq 3$, the three sources A, B and C are independent (e.g., the probability that a type one and type three customer are generated while in state two is $9.643\ 10^{-5}$). In this example, the majority of the arriving customers while in state j, are customers of type j. We further assume that the batches are ordered, that is, whenever a batch arrival occurs, the type one customer arrives first, followed by the type two customer and finally the type three customer.

The service times are assumed to be as follows. Type one customers have a deterministic service time of two time units. The service time distribution of a type two customer on the other hand, is phase-type with three phases, being three geometric phases with a mean of two, three and two time units. Finally, type three customers require a geometric service time with a mean of 5 time units. Hence,

$$T_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 2/3 & 1/3 \\ 0 & 0 & 1/2 \end{bmatrix}, \quad T_3 = [4/5],$$

and $\alpha_1 = [1\ 0]$, $\alpha_2 = [1\ 0\ 0]$, and $\alpha_3 = [1]$. As a result, the matrices $\tilde{A}_l$ are 72×72 matrices (see [9]). Figure 1 represents the delay distribution of type one, two and three customers. Both the GI/M/1 and QBD approach found the same delay distributions. The value for n_ϵ turned out to be 565, thus the

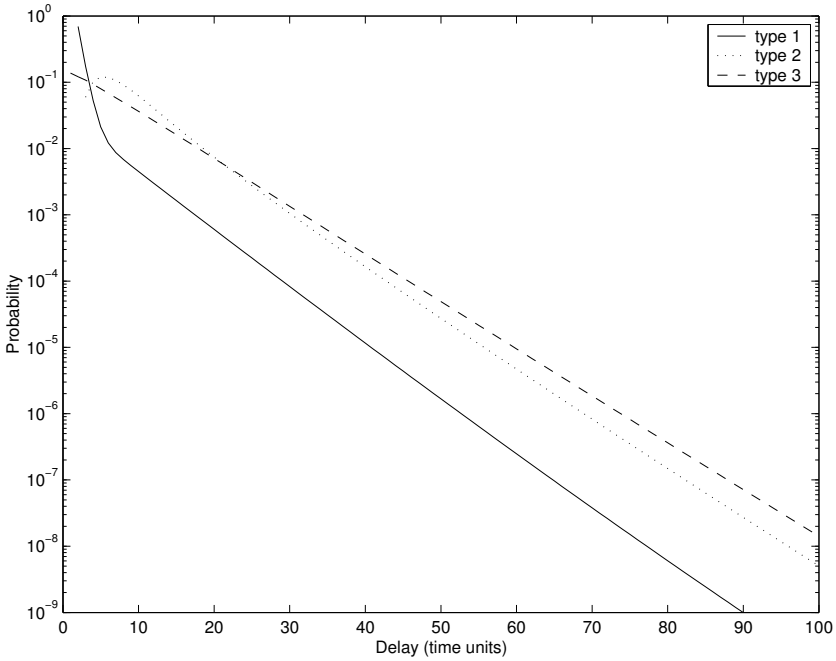

Figure 1: Delay distribution of type one, two and three customers

GI/M/1 approach uses about 46.86 Mb to store the $\tilde{A}_l$ and $\tilde{B}_l$ matrices, for the QBD approach we need 270 Kb. The computation time was also reduced by more than a factor 130, resulting in a computation time of 0.22 seconds on a 167 MHz (dual) processor (using MATLAB).

6.2 Two servers, no batch arrivals

In this case we find a similar result as in the previous section, that is, the time and memory requirements of the GI/M/1 approach depends in a linear manner on the rate at which the matrices $\bar{A}_l$ decrease to zero. We also reproduced the results presented in [10] using the QBD approach. The amount of memory needed, was reduced from 28.89 Mb to 442 Kb, the computation time by more than a factor 12 (actually, computing R for the QBD is no longer the bottleneck in this example).

Appendix

A Two servers, no batch arrivals

In this section we shortly describe the physical interpretation of the steady state vector of the GI/M/1 type MC developed in [10] to compute the waiting time distribution of a type k customer in a MMAP[K]/PH[K]/2 queue.

The MC in [10] has an infinite number of states labeled $1, 2, \ldots$. The set of states $\{1, \ldots, m\}$ was referred to as level 0_a of the MC, the states $\{m+1, \ldots, m+m_{ser}m)\}$ as level 0_b, the states $\{m+m_{ser}m+1, \ldots, m+m_{ser}m+\bar{m}_{tot}\}$ as level 0_c and finally, the states $\{m+m_{ser}m\ +i\bar{m}_{tot}+1,\ \ldots, m+m_{ser}m+(i+1)\bar{m}_{tot}\}$ were referred to as level i, $i>0$, of the MC. The variable $\bar{m}_{tot}$ is defined at the start of Section 5. The states of level 0_a are labeled as $1 \leq j \leq m$, those of 0_b as (k_1, s_1, j), where $1 \leq k_1 \leq K$, $1 \leq s_1 \leq m_{k_1}$ and $1 \leq j \leq m$, whereas the states of level 0_c and $i>0$ are labeled as (k_1, k_2, s_1, s_2, j) where $1 \leq k_2 \leq k_1 \leq K$, $1 \leq s_i \leq m_{k_i}$ (for $i=1,2$) and $1 \leq j \leq m$.

The states of the MC have the following interpretation. Assume that we observe the system at an arbitrary time instant n. Then, the MC is in state j of level 0_a at time n if the waiting room and both servers are empty at time n, while the state of the MMAP[K] at time n equals j. The MC is in state (k_1, s_1, j) of level 0_b at time n if the waiting room is empty and one server is busy with a customer of type k_1, the service of which is in phase s_1, and the state of the MMAP[K] at time n equals j. If the waiting room is empty at time n and both servers are occupied, with customers of type k_1 and k_2 (with $k_1 \geq k_2$), the service phases equal to s_1 and s_2, respectively and the state of the MMAP[K] at time n equals j, then the MC is in state (k_1, k_2, s_1, s_2, j) of level 0_c at time n. Finally, if both servers are busy (with a type k_1 and a type k_2 customer in phase s_1 and s_2, respectively) and there is at least one customer waiting in the waiting room, then the MC is in state (k_1, k_2, s_1, s_2, j) of level $i>0$ if the first, i.e., oldest, customer waiting in the waiting room arrived at time $n-i$ and the MMAP[K] arrival process is in state j at time $n-i$.

Denote the steady state vector of this MC as $\bar{\pi}^+ = (\bar{\pi}^+_{0_a}, \bar{\pi}^+_{0_b}, \bar{\pi}^+_{0_c}, \bar{\pi}^+_1, \bar{\pi}^+_2, \ldots)$,

where the subvectors $\bar{\pi}_x^+$ have a length equal to the number of states belonging to level x.

References

[1] Q. He. Queues with marked customers. *Adv. Appl. Prob.*, 28:567–587, 1996.

[2] Q. He. The versatility of the MMAP[K] and the MMAP[K]/G[K]/1 queue. *Queueing Systems*, 38:397–418, 2001.

[3] Q. He and M.F. Neuts. Markov chains with marked transitions. *Stochastic Processes and their Applications*, 74:37–52, 1998.

[4] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods and stochastic modeling.* SIAM, Philadelphia, 1999.

[5] B. Meini. Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method. *Advances in Performance Analysis*, 1:215–225, 1998.

[6] V. Ramaswami. The generality of QBD processes. In *Proc. of the 2nd Int. Conference on Matrix Analytical Methods*, Winnipeg, Manitoba, 1998.

[7] T. Takine. Queue length distribution in a FIFO single-server queue with multiple arrival streams having different service time distributions. *Queueing Systems and Applications (QUESTA)*, 39(4):349–375, 2001.

[8] T. Takine and T. Hasegawa. The workload in a MAP/G/1 queue with state-dependent services: its applications to a queue with preemptive resume priority. *Stochastic Models*, 10(1):183–204, 1994.

[9] B. Van Houdt and C. Blondia. The delay distribution of a type k customer in a first come first served MMAP[K]/PH[K]/1 queue. *J. of Appl. Probab.*, 39(1):213–222, 2002.

[10] B. Van Houdt and C. Blondia. The waiting time distribution of a type k customer in a FCFS MMAP[K]/PH[K]/2 queue. *Submitted to OR Letters*, 2002.