Discrete-Time Analysis of *MAP*/*PH*/1 Multiclass General Preemptive Priority Queue

Attahiru Sule Alfa,[1] Bin Liu,[2] Qi-Ming He[3]

[1] *Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba, R3T 5V6, Canada*

[2] *Institute of Applied Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, People's Republic of China*

[3] *Department of Industrial Engineering, Dalhousie University, Halifax, Nova Scotia, B3J 2X4, Canada*

Received 28 September 2001; revised 21 February 2003; accepted 21 February 2003

DOI 10.1002/nav.10083

Abstract: We use the matrix-geometric method to study the $MAP/PH/1$ general preemptive priority queue with a multiple class of jobs. A procedure for obtaining the block matrices representing the transition matrix P is presented. We show that the special upper triangular structure of the matrix R obtained by Miller [Computation of steady-state probabilities for $M/M/1$ priority queues, Oper Res 29(5) (1981), 945–958] can be extended to an upper triangular block structure. Moreover, the subblock matrices of matrix R also have such a structure. With this special structure, we develop a procedure to compute the matrix R. After obtaining the stationary distribution of the system, we study two primary performance indices, namely, the distributions of the number of jobs of each type in the system and their waiting times. Although most of our analysis is carried out for the case of $K = 3$, the developed approach is general enough to study the other cases ($K \geq 4$). © 2003 Wiley Periodicals, Inc. Naval Research Logistics 50: 662–682, 2003.

Keywords: priority queues; preemptive; multiclass; Markovian arrivals; phase type distributions; infinite blocks; matrix-geometric method; discrete-time analysis

1. INTRODUCTION

Priority queueing systems are very common occurrences in communication systems. Developing analytical models to be used for analyzing their performance is a very important subject, which has been dealt with by several researchers. Most of the existing models focus on the

Correspondence to: A.S. Alfa

systems with two classes of jobs. Our current paper is motivated by the fact that most realistic problems usually have more than two classes of jobs, especially in the analysis of Asynchronous Transfer Mode (ATM) traffics. With this motivation in mind, we attempt to capture the correlation aspects observed in the arrival process of such traffic by employing a class of Markovian arrival process. These aspects have not received much attention in the context of priority queues.

Most communication systems are becoming more digital than analog; hence a discrete-time approach is appropriate for their analysis. In this paper, we shall model this system as a discrete time Markov chain. The number of waiting spaces for all types of jobs are infinite. We model the arrivals as the Markovian arrival process (MAP) which allows correlated job arrivals. The processing times are assumed to be of phase type. The $M/M/1$ version of this problem was studied by Miller [10] using the matrix-geometric method. In an earlier paper [1], Alfa showed that the method can be extended to the discrete time $MAP/PH/1$ case with two types of jobs. He obtained the distribution of the number of jobs of each type in the system at steady state. He also obtained the waiting time distributions for each type of job. For the lower priority job, what he obtained is the distribution of its waiting time before it becomes the leading job of its class. In the present paper we show how to extend these results to the multiclass system. We consider $K \geq 3$ classes of jobs and let type 1 jobs have the highest priority in the system followed by type 2 jobs, etc. Hence type K job has the lowest priority.

Let us review some related papers. Gail, Hantler, and Taylor [5] studied the nonpreemptive multiserver version of the $M/M/c$ queue with two classes of jobs. Kao and Narayanan [8] used the matrix-geometric approach in conjunction with the state reduction method to develop a computationally efficient procedure for solving the model presented by Gail et al. [5]. Wagner [19] recently studied the nonpreemptive multiserver system with k classes of jobs, where k could be any positive finite integer. He allows a MAP arrival process and phase-type processing times, with the same dimension of phases for all job types. These papers [5, 8, 19] did not present any procedures for obtaining the waiting time distributions. Takine [16, 17] studied the continuous-time nonpreemptive priority $MAP/G/1$ queue, and derived various formulas on the generating function of the queue length distribution and the Laplace–Stieltjes transform (LST) of the waiting time for each class of customers. All these works deal with nonpreemptive cases only.

On the preemptive priority queue, Takine [15] studied a preemptive resume priority model with independent MAP arrival streams and state-dependent service time distributions. By analyzing the first passage time to the idle state, he derived the LST of waiting time distribution for each class of customers. However, the queue length distribution is not discussed, and numerical study is not proceeded. In fact, a numerical inversion of the double LST is of necessity for proceeding with a numerical study from his results. Choudhury, Lucantoni, and Whitt [4] have developed an inversion method for multidimensional transforms. In this current paper, we develop the matrix-geometric solution technique (Neuts [12]) to analyze a $MAP/PH/1$ queue with a fairly general preemptive priority discipline. Our results is free of the LST, which allows us to numerically compute the distributions of queue length and waiting time for each class of customers.

The remainder of the paper is organized as follows. In Section 2, we give some descriptions of the arrival and service processes of the $MAP/PH/1$ multiclass general preemptive priority queue. Our discussion from Section 3 to Section 7 is focused on the model with three classes of jobs, which is chosen as our core model. The model is mathematically described as a discrete-time Markov chain with the Quasi-Birth-Death structure in Section 3. The block matrices composing the transition probability matrix P are obtained in Section 4. In Section 5, we discuss the structure of the matrix R and develop algorithms for computation of the matrix-geometric solution. The probability distributions of queue lengths and waiting times are discussed in Section 6. Numerical results are reported in Section 7. In the last section, we point out that the proposed approach in previous sections can also be used to analyze the general model with more than three types of jobs.

2. ARRIVAL AND SERVICE PROCESSES

Our study is motivated by communication systems which involve several sources that are generating packets which get multiplexed before processing. We assume that the discrete time interval is selected such that each source can only generate a maximum of one packet during each time slot. Each packet needs at least one unit time slot for service. We then describe the arrival and service processes of the jobs based on these assumptions.

The arrivals are modeled as the Markovian arrival process (*MAP*) in which we allow correlation of the interarrival times within each class of jobs and also between all the classes. Let K be the number of classes of jobs. The *MAP* is described by the 2^K, m-phase substochastic matrices $D_{\boldsymbol{v}}$, where m is a positive integer, $\boldsymbol{v} = (v_1, v_2, \ldots, v_K)$ and $v_k \in \{0, 1\}$, with $D = \sum_{\boldsymbol{v}\in\Pi} D_{\boldsymbol{v}}$, where $\Pi = \{\boldsymbol{v}\,|v_k \in \{0, 1\};\, k = 1, 2, \ldots, K\}$ and D is a stochastic matrix. The element $(D_{\boldsymbol{v}})_{i,j}$ represents the probability of a transition from phase i to phase j with v_k arrivals of class k jobs, $k = 1, 2, \ldots, K$. The arrival rate of type k jobs is given as $\lambda_k^* = \boldsymbol{\psi}[\sum_{\boldsymbol{v}\in\Pi, v_k=1} D_{\boldsymbol{v}}]\boldsymbol{e}$, where $\boldsymbol{\psi}$ is the solution of $\boldsymbol{\psi} = \boldsymbol{\psi}D$, $\boldsymbol{\psi e} = 1$, and $\boldsymbol{e}$ is a column vector of ones. Let $\lambda^* = \sum_{k=1}^{K} \lambda_k^*$ be the total arrival rate. Denote by $J(t)$ the phase of the underlying Markov chain D at time t. This arrival process allows correlation of arrivals of jobs of the same type and between arrivals of jobs of different types. It can be classified as the discrete analogue of the Markov arrival process with marked transitions presented by He and Neuts [7]. The *MAP* was introduced by Neuts [11] as a generalization of the Poisson process. For the literature and discussions on *MAP*, see Neuts [11, 13–14] and Lucantoni [9].

One special case of this arrival process is that there are two independent discrete arrival streams described by the matrices $D_k^{(0)}$ and $D_k^{(1)}$ $(k = 1, 2)$. In such a case, $D_{(0,0)} = D_1^{(0)} \otimes D_2^{(0)}$, $D_{(1,0)} = D_1^{(1)} \otimes D_2^{(0)}$, $D_{(0,1)} = D_1^{(0)} \otimes D_2^{(1)}$, and $D_{(1,1)} = D_1^{(1)} \otimes D_2^{(1)}$, where $\otimes$ is for the Kronecker product (see Gantmacher [6]). For two matrices X and Y, $X \otimes Y$ denotes the matrix made up of blocks $[X]_{ij}Y$. A simpler case of this arrival process is that the two arrival streams are Bernoulli with parameters p_k $(k = 1, 2)$, where p_k is the probability that a type k job arrives at one unit time. Then $D_{0,0} = q_1q_2$, $D_{1,0} = p_1q_2$, $D_{0,1} = q_1p_2$, and $D_{1,1} = p_1p_2$, where $q_k = 1 - p_k$.

Processing times are of discrete phase type distribution represented by $(\boldsymbol{\beta}_k, S_k)$, with dimension m_k and mean $\bar{b}_k$ for $k = 1, 2, \ldots, K$. In this representation, $\boldsymbol{\beta}_k$ is an m_k row vector and S_k is a substochastic matrix of dimension m_k. Define that $\mathbf{S}_k^0 = \boldsymbol{e} - S_k\boldsymbol{e}$.

We study queueing systems with multiple types of customers (or jobs) and preemptive service. Type k jobs have priority over type $k + 1$ jobs $(k = 1, 2, \ldots, K - 1)$. In a preemptive system, the service process is interrupted whenever a high priority job arrives. When the server returns to serve the preempted low priority job, the service process begins from the phase where the service was interrupted. In order to incorporate this aspect into consideration, we have to record the phase at which a low priority job service was interrupted or the phase at which the interrupted service will start whenever its service resumes. For that reason let us define an $m_k \times m_k$ matrix Q_k, $k = 2, 3, \ldots, K$. The elements $[Q_k]_{i,j}$ represent the probability that for a type k job whose service was interrupted at phase i, its service resumes at phase j whenever it does resume. Clearly, the standard preemptive resume will be the case of $Q_k = \mathcal{I}_{m_k}$, where $\mathcal{I}_{m_k}$

represents an identity matrix of order m_k (note: this notation also appears in other place of this paper with the same meaning), and the preemptive repeat model will be the case of $Q_k = \boldsymbol{e\beta}_k$.

3. MARKOV CHAIN

Let us start with the case of $K = 3$ (our *core model*), which is our object for studying till Section 8. The approach developed in studying this core model will be extended further to deal with the general case of $K \geq 4$ in Section 8. As a convention, we assume that the departure always occurs prior to the arrival, and both occur before time t. Let $N_k(t)$ be the number of type k jobs in the system at time t, $J(t)$ be the arrival phase of *MAP*, and $I_k(t)$ be the present service phase of the first type k job (if it is being served at time t) or the service phase at which its service will be resumed in future (if it is not being served at time t), where $k = 1, 2, 3$ and $t = 1, 2, \ldots$. If there is no type k job in queue at t, then $I_k(t)$ is defined as value 0. Denote by $\boldsymbol{N}(t) = (N_1(t), N_2(t), N_3(t))$ and $\boldsymbol{I}(t) = (I_1(t), I_2(t), I_3(t))$. Define $\boldsymbol{X}(t) = (\boldsymbol{N}(t), J(t), \boldsymbol{I}(t))$. Then it is easy to see that $\boldsymbol{X}(t)$ is a Markov chain. Denote by $(\boldsymbol{n}, j, \boldsymbol{i}) = (n_1, n_2, n_3, j, i_1, i_2, i_3)$ its generic state. The state space of this Markov chain is $\Delta = \Delta_0 \cup \Delta_1 \cup \Delta_2 \cup \Delta_3$, where

$$\Delta_0 = \{(0, 0, 0, j, 0, 0, 0) \mid 1 \leq j \leq m\},$$

$$\Delta_1 = \{(0, 0, n_3, j, 0, 0, i_3) \mid n_3 \geq 1;\ 1 \leq j \leq m;\ i_3 \in \mathscr{E}_3^{n_3}\},$$

$$\Delta_2 = \{(0, n_2, n_3, j, 0, i_2, i_3) \mid n_2 \geq 1;\ n_3 \geq 0;\ 1 \leq j \leq m;\ i_2 \in \mathscr{E}_2^{n_2};\ i_3 \in \mathscr{E}_3^{n_3}\},$$

$$\Delta_3 = \left\{(n_1, n_2, n_3, j, i_1, i_2, i_3) \;\middle|\; \begin{array}{l} n_1 \geq 1;\ n_2 \geq 0;\ n_3 \geq 0;\ 1 \leq j \leq m; \\ i_1 \in \mathscr{E}_1^{n_1};\ i_2 \in \mathscr{E}_2^{n_2};\ i_3 \in \mathscr{E}_3^{n_3} \end{array} \right\},$$

where $\mathscr{E}_k^0 = \{0\}$ and $\mathscr{E}_k^{n_k} = \{1, 2, \ldots, m_k\}$ for all $n_k \geq 1$, $k = 1, 2, 3$.

Being in states $(0, 0, 0, j, 0, 0, 0) \in \Delta_0$ means that the server is idle with the *MAP* in its jth phase. Being in states $(\boldsymbol{n}, j, \boldsymbol{i}) \in \Delta_k$ $(k = 1, 2, 3)$ means that the *MAP* is in its jth phase, the jobs with the highest priority are of type $4 - k$ in which the first one is being processed and the processing time is in phase $i_{4-k} \in \{1, 2, \ldots, m_{4-k}\}$, and all of type h $(h \geq 5 - k)$ jobs (if exist) are waiting and the leading one will have its next service started from phase $i_h \in \{1, 2, \ldots, m_h\}$.

Let $P_{(\boldsymbol{n},j,\boldsymbol{i})\to(\boldsymbol{n}',j',\boldsymbol{i}')}$ represent the state transition probability of the Markov chain $\boldsymbol{X}(t)$ from state $(\boldsymbol{n}, j, \boldsymbol{i})$ to state $(\boldsymbol{n}', j', \boldsymbol{i}')$. All states are labeled in lexicographic order. Denote by P the transition matrix of the Markov chain $\boldsymbol{X}(t)$. Beside, we introduce some new matrices as follows. First, we focus on those state transitions, in which $N_1(t) = n_1$ and $N_1(t + 1) = n_1'$, i.e., we consider state transitions of the following form

$$(n_1, *, *, *, *, *, *) \;\to\; (n_1', *', *', *', *', *', *').$$

For fixed n_1 and n_1', the probabilities corresponding to these state transitions can be written out in a matrix form, which will be denoted by $A^{(1)}_{n_1\to n_1'}$. Similarly, we introduce matrix notations $A^{(2)}_{(n_1,n_2)\to(n_1',n_2')}$ and $A^{(3)}_{(n_1,n_2,n_3)\to(n_1',n_2',n_3')}$, which respectively correspond to the state transition probabilities

$$(n_1, n_2, *, *, *, *, *) \rightarrow (n'_1, n'_2, *', *', *', *', *')$$

and

$$(n_1, n_2, n_3, *, *, *, *) \rightarrow (n'_1, n'_2, n'_3, *', *', *', *').$$

Applying the above notations, we can write out the transition matrix P of Markov chain $\boldsymbol{X}(t)$ as follows:

$$P = \begin{bmatrix} A^{(1)}_{0\to0} & A^{(1)}_{0\to1} & & & & \\ A^{(1)}_{1\to0} & A^{(1)}_{1\to1} & A^{(1)}_{1\to2} & & & \\ & A^{(1)}_{2\to1} & A^{(1)}_{1\to1} & A^{(1)}_{1\to2} & & \\ & & \ddots & \ddots & \ddots & \\ & & & A^{(1)}_{2\to1} & A^{(1)}_{1\to1} & A^{(1)}_{1\to2} \\ & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{1}$$

Notice that, in the above formula, we have used an equality $A^{(1)}_{n_1\to n'_1} = A^{(1)}_{n_1+h\to n'_1+h}$ for all $n_1 \geq 1$ and $h \geq 1$. The proof of this equality is straightforward and omitted here.

Obviously, $\boldsymbol{X}(t)$ is a level-independent Quasi-Birth-and-Death (QBD) Markov chain. The matrix blocks $A^{(1)}_{0\to0}$, $A^{(1)}_{0\to1}$, $A^{(1)}_{1\to0}A^{(1)}_{1\to1}$, $A^{(1)}_{1\to2}$, and $A^{(1)}_{2\to1}$ will be specified in the next section.

4. BLOCK MATRICES COMPOSING THE MATRIX P

A major task in studying $\boldsymbol{X}(t)$ is to create the block matrices representing the transition matrix P. It is cumbersome because of the large number of indices that have to be considered. The aim of this section is to present a procedure for creating these matrices.

To construct the transition matrix P, we start with block matrices $A^{(3)}_{(n_1,n_2,n_3)\to(n'_1,n'_2,n'_3)}$. It is immediately clear that these matrices are of finite dimension. Notice that the dimension of the matrix $A^{(3)}_{(n_1,n_2,n_3)\to(n'_1,n'_2,n'_3)}$ is different for different (n_1, n_2, n_3) and (n'_1, n'_2, n'_3). It is because the number of possible values taken by $I_k(t)$ depends on the value of $N_k(t)$, $k = 1, 2, 3$. Specifically, $I_k(t)$ takes value 0 if the number of type k jobs is zero at t; otherwise, $I_k(t)$ may take m_k possible values $1, 2, \ldots, m_k$. All finite blocks $A^{(3)}_{(n_1,n_2,n_3)\to(n'_1,n'_2,n'_3)}$ is written out explicitly in Appendix A.

With these blocks, we are ready to construct infinite blocks $A^{(2)}_{(n_1,n_2)\to(n'_1,n'_2)}$. For convenience of exposition, we introduce notation $\boldsymbol{n}_{1:2} = (n_1, n_2)$, $\boldsymbol{n}'_{1:2} = (n'_1, n'_2)$ and $\boldsymbol{0}_{1:2} = (0, 0)$. If $\boldsymbol{n}_{1:2} = \boldsymbol{0}_{1:2}$, then we have

$$A^{(2)}_{\boldsymbol{0}_{1:2}\to\boldsymbol{n}'_{1:2}} = \begin{bmatrix} A^{(3)}_{(\boldsymbol{0}_{1:2},0)\to(\boldsymbol{n}'_{1:2},0)} & A^{(3)}_{(\boldsymbol{0}_{1:2},0)\to(\boldsymbol{n}'_{1:2},1)} & & & \\ A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},0)} & A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},2)} & & \\ & A^{(3)}_{(\boldsymbol{0}_{1:2},2)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},2)} & \\ & \ddots & \ddots & \ddots & \\ & & A^{(3)}_{(\boldsymbol{0}_{1:2},2)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{0}_{1:2},1)\to(\boldsymbol{n}'_{1:2},2)} \\ & & \ddots & \ddots & \ddots \end{bmatrix}; \tag{2}$$

otherwise (i.e., $\boldsymbol{n}_{1:2} \neq \boldsymbol{0}_{1:2}$), we have

$$A^{(2)}_{\boldsymbol{n}_{1:2}\to\boldsymbol{n}'_{1:2}} = \begin{bmatrix} A^{(3)}_{(\boldsymbol{n}_{1:2},0)\to(\boldsymbol{n}'_{1:2},0)} & A^{(3)}_{(\boldsymbol{n}_{1:2},0)\to(\boldsymbol{n}'_{1:2},1)} & & & \\ & A^{(3)}_{(\boldsymbol{n}_{1:2},1)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{n}_{1:2},1)\to(\boldsymbol{n}'_{1:2},2)} & & \\ & & A^{(3)}_{(\boldsymbol{n}_{1:2},1)\to(\boldsymbol{n}'_{1:2},1)} & A^{(3)}_{(\boldsymbol{n}_{1:2},1)\to(\boldsymbol{n}'_{1:2},2)} & \\ & & \ddots & \ddots & \end{bmatrix}. \tag{3}$$

Notice that in (2) and (3) only the first two rows and columns have different blocks and the rest appears repeatedly, i.e., the above triangular or biangular block matrices have the following property:

$$A^{(3)}_{(\boldsymbol{n}_{1:2},n_3)\to(\boldsymbol{n}'_{1:2},n'_3)} = A^{(3)}_{(\boldsymbol{n}_{1:2},n_3+h)\to(\boldsymbol{n}'_{1:2},n'_3+h)} \qquad \text{for all } n_3 \geq 1 \text{ and } h \geq 1.$$

Following a similar procedure, we now construct infinite blocks $A^{(1)}_{n_1\to n'_1}$ by using the obtained blocks $A^{(2)}_{(n_1,*)\to(n'_1,*')}$. If $n_1 = 0$, then we have

$$A^{(1)}_{0\to n'_1} = \begin{bmatrix} A^{(2)}_{(0,0)\to(n'_1,0)} & A^{(2)}_{(0,0)\to(n'_1,1)} & & & & \\ A^{(2)}_{(0,1)\to(n'_1,0)} & A^{(2)}_{(0,1)\to(n'_1,1)} & A^{(2)}_{(0,1)\to(n'_1,2)} & & & \\ & A^{(2)}_{(0,2)\to(n'_1,1)} & A^{(2)}_{(0,1)\to(n'_1,1)} & A^{(2)}_{(0,1)\to(n'_1,2)} & & \\ & \ddots & \ddots & \ddots & & \\ & & A^{(2)}_{(0,2)\to(n'_1,1)} & A^{(2)}_{(0,1)\to(n'_1,1)} & A^{(2)}_{(0,1)\to(n'_1,2)} & \\ & & \ddots & \ddots & \ddots & \end{bmatrix}; \tag{4}$$

otherwise (i.e., $n_1 \neq 0$), we have

$$A^{(1)}_{n_1\to n'_1} = \begin{bmatrix} A^{(2)}_{(n_1,0)\to(n'_1,0)} & A^{(2)}_{(n_k,0)\to(n'_k,1)} & & & \\ & A^{(2)}_{(n_1,1)\to(n'_1,1)} & A^{(2)}_{(n_1,1)\to(n'_1,2)} & & \\ & & A^{(2)}_{(n_1,1)\to(n'_1,1)} & A^{(2)}_{(n_1,1)\to(n'_1,2)} & \\ & & \ddots & \ddots & \end{bmatrix}. \tag{5}$$

Again, the above matrices have following property

$$A^{(2)}_{(n_1,n_2)\to(n'_1,n'_2)} = A^{(2)}_{(n_1,n_2+h)\to(n'_1,n'_2+h)} \qquad \text{for all } n_2 \geq 1 \text{ and } h \geq 1.$$

By using Eqs. (2)–(5) and the above property, we can obtain $A^{(1)}_{0\to 0}$, $A^{(1)}_{0\to 1}$, $A^{(1)}_{1\to 0}$, $A^{(1)}_{1\to 2}$, $A^{(1)}_{1\to 1}$, and $A^{(1)}_{2\to 1}$. Further, the transition matrix P can be obtained by using (1).

5. MATRIX-GEOMETRIC SOLUTION

5.1. Structure of Matrix *R*

The matrix P is of the Quasi-Birth-Death (QBD) type and can be analyzed by the matrix-geometric method (see Neuts [12]). Unlike most cases, its blocks are of infinite dimensions. The first step in analyzing this system is to determine the matrix R, which is the minimal nonnegative solution to the following matrix quadratic equation:

$$R = A^{(1)}_{1\to 2} + RA^{(1)}_{1\to 1} + R^2 A^{(1)}_{2\to 1}. \tag{6}$$

The matrix R has all its eigenvalues within the unit disk if the stability condition is satisfied.

REMARK 1: The stability condition is $\rho = \lambda_1^*\bar{b}_1 + \lambda_2^*\bar{b}_2 + \lambda_3^*\bar{b}_3 < 1$ for the preemptive resume case, i.e., $Q_2 = \mathcal{I}_{m_2}$ and $Q_3 = \mathcal{I}_{m_3}$, which can be proved by using the same argument in Theorem 1 in Alfa [1]. For general Q_2 and Q_3, conditions for positive recurrence are yet to be identified.

The result in Alfa [1] and Miller [10] can be extended to show that for the QBD case the matrix R has the form:

$$R = \begin{bmatrix} R_{0,0} & R_{0,1} & R_{0,2} & R_{0,3} & R_{0,4} & \cdots \\ & R_{1,1} & R_{1,2} & R_{1,3} & R_{1,4} & \cdots \\ & & R_{1,1} & R_{1,2} & R_{1,3} & \cdots \\ & & & R_{1,1} & R_{1,2} & \cdots \\ & & & & \ddots & \ddots \end{bmatrix}, \tag{7}$$

where $R_{n_2,n_2'}$ ($n_2 = 0, 1$; $n_2' = n_2 + h_2$, $h_2 \geq 0$) can be further partitioned into smaller blocks $R_{(n_2,n_3),(n_2',n_3')}$ ($n_3 = 0, 1$; $n_3' = n_3 + h_3$, $h_3 \geq 0$), namely,

$$R_{n_2,n_2'} = \begin{bmatrix} R_{(n_2,0),(n_2',0)} & R_{(n_2,0),(n_2',1)} & R_{(n_2,0),(n_2',2)} & R_{(n_2,0),(n_2',3)} & R_{(n_2,0),(n_2',4)} & \cdots \\ & R_{(n_2,1),(n_2',1)} & R_{(n_2,1),(n_2',2)} & R_{(n_2,1),(n_2',3)} & R_{(n_2,1),(n_2',4)} & \cdots \\ & & R_{(n_2,1),(n_2',1)} & R_{(n_2,1),(n_2',2)} & R_{(n_2,1),(n_2',3)} & \cdots \\ & & & R_{(n_2,1),(n_2',1)} & R_{(n_2,1),(n_2',2)} & \cdots \\ & & & & \ddots & \ddots \end{bmatrix}. \tag{8}$$

The arguments leading to the above formulae are as follows. Notice that the element $R_{(n_2,n_3,j,\boldsymbol{i}),(n_2',n_3',j',\boldsymbol{i}')}$ of the matrix R can be explained as the expected number of visits to the state $(n_1 + 1, n_2', n_3', j', \boldsymbol{i}')$ before the first return to the level n_1, given that the process starts in the state $(n_1, n_2, n_3, j, \boldsymbol{i})$ (see Neuts [12]). By this definition it is straightforward to show that the matrix R is of the upper triangular block type. This is because the server will never serve the lower priority jobs as long as there is a type 1 job in the system. In addition, since the arrivals and services are independent of the state of the system when there is at least one type 1 job in the system, we have $R_{n_2,n_2'} = R_{n_2+h,n_2'+h}$, where $n_2' \geq n_2 \geq 1$ and $h \geq 0$. This leads to (7). Equation (8) can be explained similarly.

5.2. Useful Results on Computation of Matrix *R*

To develop an algorithm of computing the stationary distribution, those blocks of matrix R must be studied first. In this subsection, we shall present some preliminary results, which play an important role for us to develop an algorithm on computing these blocks in next subsections.

DEFINITION 1: A matrix X is called as *Special Upper Triangular Block Type* (*SUTBT*) matrix if it has the following form:

$$X = \begin{bmatrix} C_0 & C_1 & C_2 & C_3 & C_4 & \cdots \\ & D_1 & D_2 & D_3 & D_4 & \cdots \\ & & D_1 & D_2 & D_3 & \cdots \\ & & & D_1 & D_2 & \cdots \\ & & & & \ddots & \ddots \end{bmatrix},$$

where $\{C_n\}_{n=0}^{\infty}$ and $\{D_n\}_{n=1}^{\infty}$ are two strings of matrices, which satisfy that all D_n ($n \geq 1$) must be square matrices with same dimension, and C_0 is also a square matrix but it may have different dimension from D_n. We shall call X as the *SUTBT* matrix generated by $\{C_n\}_{n=0}^{\infty}$ and $\{D_n\}_{n=1}^{\infty}$. Notice that C_n ($n \geq 1$) may not be square matrices.

PROPOSITION 1: If X is a *SUTBT* matrix generated by $\{C_n\}_{n=0}^{\infty}$ and $\{D_n\}_{n=1}^{\infty}$, and Y is a *SUTBT* matrix generated by $\{E_n\}_{n=0}^{\infty}$ and $\{F_n\}_{n=1}^{\infty}$, where C_0 and E_0 have the same dimension, and also D_n and F_n have the same dimension, then $Z = XY$ is still a *SUTBT* matrix, which are generated by $\{G_n\}_{n=0}^{\infty}$ and $\{H_n\}_{n=1}^{\infty}$, where

$$G_n = C_0E_n + \sum_{k=1}^{n} C_kF_{n+1-k}, \qquad n = 0, 1, \ldots,$$

$$H_n = \sum_{k=1}^{n} D_kF_{n+1-k}, \qquad n = 1, 2, \ldots,$$

where $\Sigma_{k=1}^{0} C_kF_{1-k} = 0$, by definition.

COROLLARY 1: If X is a *SUTBT* matrix generated by $\{C_n\}_{n=0}^{\infty}$ and $\{D_n\}_{n=1}^{\infty}$, then X^2 is also a *SUTBT* matrix.

The proof of Proposition 1 and Corollary 1 are simple and omitted.

COROLLARY 2: Define X^{-1} as the inverse of the *SUTBT* matrix X such that $X^{-1}X = I$. Assume that C_0 and D_1 are invertible. Define a *SUTBT* matrix generated by $\{E_n\}_{n=0}^{\infty}$ and $\{F_n\}_{n=1}^{\infty}$, where E_n and F_n are given by

$$E_0 = C_0^{-1}, \qquad F_1 = D_1^{-1},$$

$$E_n = -C_0^{-1} \sum_{k=1}^{n} C_kF_{n+1-k}, \qquad n = 1, 2, \ldots,$$

$$F_n = -D_1^{-1} \sum_{k=2}^{n} D_kF_{n+1-k}, \qquad n = 2, 3, \ldots.$$

Then that *SUTBT* matrix is an inverse matrix of X. Furthermore, when it exists, the inverse matrix is unique. Therefore, we denote the above constructed inverse matrix as X^{-1}.

PROOF: It is easy to verify that the above constructed matrix is an inverse matrix. Suppose that the inverse matrix is not unique. Then there can be two matrices Y and W such that $YX = I$ and $WX = I$. That leads to $(Y - W)X = 0$. Consider the first row of the matrix $Y - W$. Since C_0 and D_1 are invertible, it can be verified that the first row of the matrix $Y - W$ must be zero.

Similarly, every row of $Y - W$ must be zero. Thus, $Y = W$, which is a contradiction. Therefore, the inverse of X is unique. This completes the proof.

5.3. Computation of Matrix R

To obtain R numerically, a standard procedure is to use Eq. (6) recursively starting with the initial assumption $R = 0$. It is worthwhile to note that both matrix R and its subblock matrices in our model are of infinite dimension. For purpose of computation, the truncation has to be done. However, the truncated R usually has very large dimension, which causes the computation of R time-consuming significantly if one simply follows this standard procedure. In order to reduce the workload of computation, we shall make some improvement on this standard procedure.

For simplification of notations, in the next subsections we often drop the superscripts from our notations, e.g., we shall use $A_{(1,0)\to(2,0)}$ to simply represent $A^{(2)}_{(1,0)\to(2,0)}$. Let us observe those matrices which appear in (6) now. We easily find that they are all *SUTBT* matrices [see (5) and (7)]. Specifically, (i) R is generated by $\{R_{0,n}\}_{n=0}^{\infty}$ and $\{R_{1,n}\}_{n=1}^{\infty}$, (ii) $A_{1\to2}$ is generated by $\{A_{(1,0)\to(2,n)}\}_{n=0}^{\infty}$ and $\{A_{(1,1)\to(2,n)}\}_{n=1}^{\infty}$, (iii) $A_{1\to1}$ is generated by $\{A_{(1,0)\to(1,n)}\}_{n=0}^{\infty}$ and $\{A_{(1,1)\to(1,n)}\}_{n=1}^{\infty}$, (iv) $A_{1\to2}$ is generated by $\{A_{(2,0)\to(1,n)}\}_{n=0}^{\infty}$ and $\{A_{(2,1)\to(1,n)}\}_{n=1}^{\infty}$. Moreover, $A_{(1,0)\to(2,n)} \equiv A_{(1,0)\to(1,n)} \equiv A_{(2,0)\to(1,n)} \equiv 0$ for $n \geq 2$, $A_{(1,1)\to(2,n)} \equiv A_{(1,1)\to(1,n)} \equiv A_{(2,1)\to(1,n)} \equiv 0$ for $n \geq 3$. Beside, if we let $B = R^2$, then B is also a *SUTBT* matrix (by Corollary 1) which is generated by $\{B_{0,n}\}_{n=0}^{\infty}$ and $\{B_{1,n}\}_{n=1}^{\infty}$, where

$$B_{0,n} = R_{0,0}R_{0,n} + \sum_{k=1}^{n} R_{0,k}R_{1,n+1-k}, \qquad n = 0, 1, \ldots, \tag{9}$$

$$B_{1,n} = \sum_{k=1}^{n} R_{1,k}R_{1,n+1-k}, \qquad n = 1, 2, \ldots, \tag{10}$$

Applying Proposition 1, we can rewrite Eq. (6) as follows:

$$R_{0,0} = A_{(1,0)\to(2,0)} + R_{0,0}A_{(1,0)\to(1,0)} + B_{0.0}A_{(2,0)\to(1,0)}, \tag{11}$$

$$R_{0,1} = A_{(1,0)\to(2,1)} + R_{0,0}A_{(1,0)\to(1,1)} + R_{0.1}A_{(1,1)\to(1,1)}, + B_{0,0}A_{(2,0)\to(1,1)} + B_{0,1}A_{(2,1)\to(1,1)}, \tag{12}$$

$$R_{0,n} = R_{0,n-1}A_{(1,1)\to(1,2)} + R_{0,n}A_{(1,1)\to(1,1)} + B_{0,n-1}A_{(2,1)\to(1,2)} + B_{0,n}A_{(2,1)\to(1,1)}, \qquad n = 2, 3, \ldots, \tag{13}$$

$$R_{1,1} = A_{(1,1)\to(2,1)} + R_{1,1}A_{(1,1)\to(1,1)} + B_{1,1}A_{(2,1)\to(1,1)}, \tag{14}$$

$$R_{1,n} = \delta_2(n)A_{(1,1)\to(2,2)} + R_{1,n-1}A_{(1,1)\to(1,2)} + R_{1,n}A_{(1,1)\to(1,1)} + B_{1,n-1}A_{(2,1)\to(1,2)} + B_{1,n}A_{(2,1)\to(1.1)}, \qquad n = 2, 3, \ldots, \tag{15}$$

where we define $\delta_2(n) = 1$ if $n = 2$; otherwise $\delta_2(n) = 0$. As can be seen in (11)–(15), $R_{0,0}$ can be calculated from Eq. (11), then the rest of $R_{i,j}$ can be calculated recursively.

Notice that all matrices appearing in (9)–(15) are still *SUTBT* matrices [see (3) and (8)]. For instance, $R_{1,2}$ is generated by $\{R_{(1,0),(2,n)}\}_{n=0}^{\infty}$ and $\{R_{(1,1),(2,n)}\}_{n=1}^{\infty}$; $A_{(1,0)\to(2,1)}$ is generated by $\{A_{(1,0,0)\to(2,1,n)}\}_{n=0}^{\infty}$ and $\{A_{(1,0,1)\to(2,1,n)}\}_{n=1}^{\infty}$, where $A_{(1,0,0)\to(2,1,n)} \equiv 0$ for $n \geq 2$ and $A_{(1,0,1)\to(2,1,n)} \equiv 0$ for $n \geq 3$; $B_{1,2}$ is generated by $\{B_{(1,0),(2,n)}\}_{n=0}^{\infty}$ and $\{B_{(1,1),(2,n)}\}_{n=1}^{\infty}$.

To obtain the matrix R, we only need to calculate its basic subblocks [see (7)–(8)] $R_{(n_2,n_3),(n_2',n_3')}$, where n_2, $n_3 \in \{0, 1\}$, $n_2' = n_2 + h_2$, $n_3' = n_3 + h_3$, $h_2 \geq 0$, $h_3 \geq 0$. This can be done by using the following iteration algorithm, which is based on Eqs. (9)–(15) and Proposition 1.

ALGORITHM 1:

Step 1. Making an initial assumption: $R_{(n_2,n_3),(n_2',n_3')} \equiv 0$, where n_2, $n_3 \in \{0, 1\}$, $n_2' = n_2 + h_2$, $n_3' = n_3 + h_3$, $h_2 \geq 0$, $h_3 \geq 0$.

Step 2. Calculating all subblocks $B_{(n_2,n_3),(n_2',n_3')}$ of matrix $B_{n_2,n_2'}$ by using (9)–(10) and Proposition 1, where $\forall n_2$, $n_3 \in \{0, 1\}$, $n_2' = n_2 + h_2$, $n_3' = n_3 + h_3$, $\forall h_2 \geq 0$, $\forall h_3 \geq 0$.

Step 3. Calculating all subblocks $R_{(n_2,n_3),(n_2',n_3')}$ of new matrix $R_{n_2,n_2'}$ by using (11)–(15) and Proposition 1, where $\forall n_2$, $n_3 \in \{0, 1\}$, $n_2' = n_2 + h_2$, $n_3' = n_3 + h_3$, $\forall h_2 \geq 0$, $\forall h_3 \geq 0$.

Step 4. Go back to Step 2 and start a new iteration. The iteration will be terminated if the maximum absolute error between two iterations is smaller than a given constant (e.g., 10^{-10}).

5.4. Stationary Distribution

Denote by $\boldsymbol{x}$ the steady-state distribution of Markov chain $X(t)$. Hence $\boldsymbol{x} = \boldsymbol{x}P$ and $\boldsymbol{x}\boldsymbol{e} = 1$. Corresponding to the construction of matrix P, we partition $\boldsymbol{x}$ into subvectors and assume that $\boldsymbol{x} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots]$, where $\boldsymbol{x}_{n_1}$ is partitioned into smaller subvectors

$$\boldsymbol{x}_{n_1} = [\boldsymbol{x}_{(n_1,0)}, \boldsymbol{x}_{(n_1,1)}, \boldsymbol{x}_{(n_1,2)}, \cdots], \qquad n_1 \geq 0,$$

and also $\boldsymbol{x}_{(n_1,n_2)}$ can be further partitioned as

$$\boldsymbol{x}_{(n_1,n_2)} = [\boldsymbol{x}_{(n_1,n_2,0)}, \boldsymbol{x}_{(n_1,n_2,1)}, \boldsymbol{x}_{(n_1,n_2,2)}, \cdots].$$

Finally, all of these vectors are partitioned into basic subvectors $\boldsymbol{x}_{(n_1,n_2,n_3)}$ (n_1, n_2, $n_3 \geq 0$). Notice that the dimension of vector $\boldsymbol{x}_{(n_1,n_2,n_3)}$ is different for different (n_1, n_2, n_3). In fact, $\boldsymbol{x}_{(n_1,n_2,n_3)}$ is an $m \prod_{k=1}^{3} \zeta_k(n_k)$ dimension row vector, where $\zeta_k(0) = 1$ and $\zeta_k(n_k) = m_k$ if $n_k \geq 1$ ($k = 1, 2, 3$).

It is shown in Tweedie [18] that

$$\boldsymbol{x}_{n_1+1} = \boldsymbol{x}_{n_1}R, \qquad n_1 \geq 1. \tag{16}$$

and also $(\boldsymbol{x}_0, \boldsymbol{x}_1)$ can be determined from the relationship

$$(\boldsymbol{x}_0, \boldsymbol{x}_1) = (\boldsymbol{x}_0, \boldsymbol{x}_1)\begin{bmatrix} A_{0\to0} & A_{0\to1} \\ A_{1\to0} & A_{1\to1} + RA_{2\to1} \end{bmatrix}, \tag{17}$$

$$\boldsymbol{x}_0\boldsymbol{e}_{\boldsymbol{x}_0} + \boldsymbol{x}_1(I - R)^{-1}\boldsymbol{e}_{\boldsymbol{x}_1} = 1, \tag{18}$$

where $\boldsymbol{e}_{\boldsymbol{x}_0}$ and $\boldsymbol{e}_{\boldsymbol{x}_1}$ represent two column vectors in which all elements are 1, and the numbers of elements are same as that of $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$, respectively.

Let us define $U = I - A_{1\to1} - RA_{2\to1}$ and $V = (I - R)U$. By Proposition 1 and Corollary 2, we know that U is a *SUTBT* matrix. Since $\begin{bmatrix} A_{0\to0} & A_{0\to1} \\ A_{1\to0} & A_{1\to1} + RA_{2\to1} \end{bmatrix}$ is stochastic and $A_{1\to0}$ is not zero, then $A_{1\to1} + RA_{2\to1}$ is strictly substochastic. Hence by Theorem 5.1.1 in Bhat [3] it is clear that the inverse of U exists. Thus we can rewrite (17) and (18) in the following form:

$$\boldsymbol{x}_1 = \boldsymbol{x}_0A_{0\to1}U^{-1}, \tag{19}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_0(A_{0\to0} + A_{0\to1}U^{-1}A_{1\to0}), \tag{20}$$

$$1 = \boldsymbol{x}_0(\boldsymbol{e}_{\boldsymbol{x}_0} + A_{0\to1}V^{-1}\boldsymbol{e}_{\boldsymbol{x}_1}). \tag{21}$$

By (20) and (21), we now present a simple iterative algorithm to calculating $\boldsymbol{x}_0$:

$$\boldsymbol{x}_0^{[k+1]} = (\boldsymbol{x}_0^{[k]}B)/(\boldsymbol{x}_0^{[k]}B\boldsymbol{b}), \qquad k = 0, 1, 2, \ldots, \tag{22}$$

where $B = A_{0\to0} + A_{0\to1}U^{-1}A_{1\to0}$ and $\boldsymbol{b} = \boldsymbol{e}_{\boldsymbol{x}_0} + A_{0\to1}V^{-1}\boldsymbol{e}_{\boldsymbol{x}_1}$

REMARK 2: In essence, the iterative approach (22) is a classical *power method* (see Atkinson [2]) with a minor modification on scaling. Its convergence can be proved in a manner similar to Atkinson [2].

ALGORITHM 2:

Step 1. Starting with an initial assumption

$$\boldsymbol{x}_{(n_1,n_2,n_3)}^{[0]} = \left\{ \begin{array}{ll} (1, 0, \ldots, 0), & \text{if } \boldsymbol{n} = (0, 0, 0), \\ \text{zero vector}, & \text{if } \boldsymbol{n} \neq \mathbf{(0, 0, 0)}, \end{array} \right\}$$

we can obtain the initial value $\boldsymbol{x}_0^{[0]}$.

Step 2. Calculate a new value of $\boldsymbol{x}_0$ repeatedly by using Eq. (22). The iteration will be terminated if the maximum absolute error between two iterations is smaller than a given constant (e.g., 10^{-10}).

In the above algorithm, we need to calculate U^{-1} and V^{-1}. It can be done as follows. Notice that U is a *SUTBT* matrix generated by $\{U_{0,n}\}_{n=0}^{\infty}$ and $\{U_{1,n}\}_{n=1}^{\infty}$, where

$$U_{0,0} = I - A_{(1,0)\to(1,0)} - R_{0,0}A_{(2,0)\to(1,0)}, \tag{23}$$

$$U_{0,1} = -A_{(1,0)\to(1,1)} - R_{0,0}A_{(2,0)\to(1,1)} - R_{0,1}A_{(2,1)\to(1,1)}, \tag{24}$$

$$U_{0,n} = -R_{0,n-1}A_{(2,1)\to(1,2)} - R_{0,n}A_{(2,1)\to(1,1)}, \qquad n \geq 2, \tag{25}$$

$$U_{1,1} = I - A_{(1,1)\to(1,1)} - R_{1,1}A_{(2,1)\to(1,1)}, \tag{26}$$

$$U_{1,n} = -\delta_2(n)A_{(1,1)\to(1,2)} - R_{1,n-1}A_{(2,1)\to(1,2)} - R_{1,n}A_{(2,1)\to(1,1)}, \qquad n \geq 2. \tag{27}$$

Based on (23)–(27), we can obtain U^{-1} by using Corollary 2. Similarly, we can get V^{-1}.

6. PERFORMANCE MEASURES

6.1. Queue Lengths

It is quite straightforward to obtain the stationary distributions of queue lengths from the stationary distribution of $X(t)$:

$$\text{Prob}\{N_1 = n_1, N_2 = n_2, N_3 = n_3\} = \boldsymbol{x}_{(n_1,n_2,n_3)}\boldsymbol{e}_{\boldsymbol{x}(n_1,n_2,n_3)}, \qquad \forall\, n_1, n_2, n_3 = 0, 1, 2, \ldots, \tag{28}$$

where N_k represents the queue length of type k jobs in the system when it is in steady-state, and $\boldsymbol{e}_{\boldsymbol{x}(n_1,n_2,n_3)}$ represents a column vector in which all elements are 1.

6.2. Waiting Times

Let W_k be the waiting time of a type k job, i.e., the time from its arrival to the completion of its processing, $k = 1, 2, 3$. In this subsection we shall study their probability distributions.

TYPE 3 JOB: Let us focus on an arbitrary *marked* type 3 job which arrives while the system is in steady state. If the system state at the moment immediately after its arrival is $(n_1, n_2, n_3, j, i_1, i_2, i_3)$, then its waiting time will be the accumulated processing time of the present n_1 type 1 jobs, n_2 type 2 jobs and n_3 type 3 jobs (including itself), added by any possible other type 1 and type 2 jobs, which may arrive during the marked job's waiting and being processed. It should be pointed out that any other type 3 jobs arriving after the marked job has no impact on the waiting time of the marked job, therefore we can neglect them while studying the waiting time of the marked job. Without loss of generality, we assume that this marked job arrives at time $t = 0$ and the system has been in steady state.

To obtain the distribution of waiting time W_3 of this marked type 3 job, the first step is to determine the stationary probability distribution of the system's states immediately after its arrival. This is actually the stationary distribution of the number of jobs in the system as seen by this marked job at its arrival time. Denote by $\boldsymbol{\pi}$ this distribution which can be obtained from $\boldsymbol{x}$ by a standard method, i.e., $\boldsymbol{\pi}$ can be interpreted as a conditional probability distribution of the system's states conditioning on the occurrence of a type 3 job arrival. Note that there are only four possible situations of the type-3 job's arrivals, which correspond to $D_{(0,0,1)}$, $D_{(0,1,1)}$, $D_{(1,0,1)}$, and $D_{(1,1,1)}$. Let $D^*_{(0,0,1)} = D_{(0,0,1)}$, $D^*_{(0,1,1)} = D_{(0,1,1)}$, $D^*_{(1,0,1)} = D_{(1,0,1)}$, $D^*_{(1,1,1)} = D_{(1,1,1)}$, and $D^*_{(0,0,0)} = D^*_{(0,1,0)} = D^*_{(1,0,0)} = D^*_{(1,1,0)} = 0$ (zero matrix). By using these D^*_v to replace D_v and then following the procedure of creating the matrix P, we can write out a new matrix P^*_3. Further, we have $\boldsymbol{\pi} = \frac{1}{\lambda^*_3}\cdot xP^*_3$ (see Section 2 for the definition of λ^*_3).

Next we present an auxiliary Markov chain $\hat{\boldsymbol{X}}(t)$ with some absorbing states, which can be constructed simply based on $\boldsymbol{X}(t)$ by making two modifications as follows. Any other type 3 job arriving after this marked job is considered to be lost naturally, and all states in set Δ_0 are assumed as absorbing states. Such modifications are easy to be implemented in computation. Denote by $\hat{P}$ the transition matrix of Markov chain $\hat{X}(t)$. Following the procedure in Sections 3–4, one can write out the matrix $\hat{P}$ by making two minor modifications as follows:

MODIFICATION 1: Arrival process is modified as a new *MAP* with substochastic matrices $\hat{D}_{\boldsymbol{v}}$, where $\hat{D}_{(0,0,0)} = D_{(0,0,0)} + D_{(0,0,1)}$, $\hat{D}_{(0,1,0)} = D_{(0,1,0)} + D_{(0,1,1)}$, $\hat{D}_{(1,0,0)} = D_{(1,0,0)} + D_{(1,0,1)}$, $\hat{D}_{(1,1,0)} = D_{(1,1,0)} + D_{(1,1,1)}$, and $\hat{D}_{(v_1,v_2,1)} = 0, \quad \forall v_1, v_2 \in \{0. 1\}$.

MODIFICATION 2: Case A in Section 4 is modified as

$$A^{(3)}_{(0,0,0)\to\boldsymbol{n}'} = \begin{cases} \text{an identity matrix of order } m, & \text{if } \boldsymbol{n}' = (0,0,0), \\ \text{an } m \times m \prod_{k=1}^{3} \zeta_k(n'_k) \text{ zero matrix}, & \text{otherwise.} \end{cases}$$

Let $\hat{\boldsymbol{x}}(t)$ be the probability distribution of $\hat{X}(t)$ at time $t = 0, 1, 2, \ldots$. Thus

$$\hat{\boldsymbol{x}}(t) = \hat{\boldsymbol{x}}(0)\hat{P}^t, \qquad t = 1, 2, \ldots. \tag{29}$$

Now let us return to consider the waiting time W_3 of the marked job, we easily find that it can be viewed as the time that $\hat{X}(t)$ visits an absorbing state for the first time given that its initial distribution is $\hat{\boldsymbol{x}}(0) = \boldsymbol{\pi}$. In other words, we have

$$\text{Prob}\{W_3 \le t\} = \text{Prob}\{\hat{X}(t) \in \Delta_0 \mid \hat{X}(0) \text{ having distribution } \boldsymbol{\pi}\}, \qquad t = 1, 2, \ldots. \tag{30}$$

Since the probability distribution $\hat{\boldsymbol{x}}(t)$ can be calculated recursively by (29), the cumulative probability distribution of W_3 can be obtained by (30).

TYPE 2 JOB: As far as the type 2 job is concerned, all type 3 jobs do not affect its waiting time W_2 and hence can be neglected; i.e., the system can be viewed as with type 1 and type 2 jobs only. The corresponding *MAP* is described by four m-phase substochastic matrices $D_{(0,0)}$, $D_{(0,1)}$, $D_{(1,0)}$, and $D_{(1,1)}$, where $D_{(0,0)} = D_{(0,0,0)} + D_{(0,0,1)}$, $D_{(0,1)} = D_{(0,1,0)} + D_{(0,1,1)}$, $D_{(1,0)} = D_{(1,0,0)} + D_{(1,0,1)}$, and $D_{(1,1)} = D_{(1,1,0)} + D_{(1,1,1)}$. On waiting time W_2, its cumulative probability distribution has been dealt with in Alfa [1].

TYPE 1 JOB: To obtain the distribution of W_1, we can simply treat our system as a single server queue with only type 1 jobs in it. This has been done in Alfa [1].

7. NUMERICAL COMPUTATIONS AND DISCUSSION

In this section we apply the results obtained in the previous sections to an example. Our algorithm is developed into a computer code in MATLAB language. We shall let it run on IBM PC Pentium III-800 machine.

7.1. An Example

In this example, we assume that the arrival are modeled as a *MAP* with the following parameter setting:

$$D_{(0,0,0)} = \begin{bmatrix} 0.30 & 0.30 \\ 0.40 & 0.20 \end{bmatrix}, \quad D_{(0,0,1)} = \begin{bmatrix} 0.07 & 0.02 \\ 0.03 & 0.05 \end{bmatrix}, \quad D_{(0,1,0)} = \begin{bmatrix} 0.04 & 0.05 \\ 0.07 & 0.03 \end{bmatrix},$$

$$D_{(0,1,1)} = \begin{bmatrix} 0.01 & 0.00 \\ 0.00 & 0.01 \end{bmatrix}, \quad D_{(1,0,0)} = \begin{bmatrix} 0.15 & 0.00 \\ 0.10 & 0.05 \end{bmatrix}, \quad D_{(1,0,1)} = \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 0.02 \end{bmatrix},$$

$$D_{(1,1,0)} = \begin{bmatrix} 0.02 & 0.00 \\ 0.01 & 0.01 \end{bmatrix}, \quad D_{(1,1,1)} = \begin{bmatrix} 0.01 & 0.00 \\ 0.00 & 0.01 \end{bmatrix}.$$

The processing times of type 1, type 2, and type 3 jobs are assumed to have phase type distributions with parameters

$$\boldsymbol{\beta}_1 = [0.80 \quad 0.20], \qquad \boldsymbol{\beta}_2 = [0.95 \quad 0.05], \qquad \boldsymbol{\beta}_3 = [0.95 \quad 0.05],$$

$$S_1 = \begin{bmatrix} 0.05 & 0.00 \\ 0.00 & 0.05 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 0.40 & 0.55 \\ 0.00 & 0.05 \end{bmatrix}, \quad S_3 = \begin{bmatrix} 0.40 & 0.30 \\ 0.00 & 0.05 \end{bmatrix},$$

We consider two different priority disciplines.

(a) The Standard Preemptive Resume Model (SPRM), $Q_2 = Q_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

(b) The Preemptive Repeat Processing Model (PRPM), $Q_k = \begin{bmatrix} \boldsymbol{\beta}_k \\ \boldsymbol{\beta}_k \end{bmatrix}$ $(k = 2, 3)$.

7.2. Approximate Solution

In order to obtain the stationary measures of system, we first need to compute R. Since the matrix R and its block matrices $R_{0,n}$ $(n \geq 0)$ and $R_{1,n}$ $(n \geq 1)$ are all of infinite dimension, we have to truncate them for our computational purpose. We use a positive integer κ as our truncation point. That is, the truncated R is composed of blocks $\{R_{0,n}\}_{n=0}^{\kappa}$ and $\{R_{1,n}\}_{n=1}^{\kappa}$, and the truncated $R_{n_2,n_2'}$ is composed of subblocks $\{R_{(n_2,0),(n_2',n)}\}_{n=0}^{\kappa}$ and $\{R_{(n_2,1),(n_2',n)}\}_{n=1}^{\kappa}$. The selection of value κ is an important issue. An ideal value of κ should be one that guarantees the accuracy of computational results, but does not involve too much computational effort. After truncating the matrix R, we shall use Algorithms 1 and 2 to compute R and $\boldsymbol{x}_0$, respectively. Our iterations terminate whenever the maximum absolute error of two neighboring iterative results is smaller than $\epsilon = 10^{-10}$. In other words, the error bound objectives are $\|R^{[n-1]} - R^{[n]}\|_\infty < \epsilon$ and $\|\boldsymbol{x}_0^{[n+1]} - \boldsymbol{x}_0^{[n]}\|_\infty < \epsilon$, respectively.

To evaluate the accuracy of computational results under different truncations, we let κ take different truncation values for our SPRM. In Table 1, we make a comparison among the computational results on the distributions of queue length for $\kappa = 5, 10, 12, 15$. Moreover, we provide the "exact solution" on type 1 jobs in comparison with our approximate solution. For this exact solution, we consider a new *MAP/PH/1* system with only type 1 jobs. This system can be obtained from our original system by ignoring all type 2 and type 3 jobs because they do not affect the service process of type 1 jobs under the preemptive priority discipline. Therefore, its arrival process is determined by the two substochastic matrices D_0 and D_1, where $D_0 = \sum_{v_2=0}^{1} \sum_{v_3=0}^{1} D_{(0,v_2,v_3)}$ and $D_1 = \sum_{v_2=0}^{1} \sum_{v_3=0}^{1} D_{(1,v_2,v_3)}$. Because the matrix R is of finite dimension, the stationary measures can be computed directly without any truncation involved. In this sense, we can take this solution as an "exact solution." We shall use it to check the error of our approximate results. As shown in Table 1, the accuracy of computational results is

Table 1. Comparison of exact and approximate results under different truncations.

	Prob$\{N_1 = n\}$				Prob$\{N_2 = n\}$				Prob$\{N_3 = n\}$			
n	$\kappa = 5$	10	15	Exa.	$\kappa = 5$	10	12	15	$\kappa = 5$	10	12	15
0	.7805	.7794	.7793	.7789	.5531	.5460	.5454	.5449	.3455	.2848	.2756	.2678
1	.2165	.2175	.2177	.2180	.3188	.3186	.3186	.3186	.2733	.2295	.2225	.2165
2	.0030	.0030	.0030	.0030	.0974	.1001	.1004	.1006	.1738	.1536	.1496	.1459
3	.0000	.0000	.0000	.0000	.0242	.0263	.0265	.0267	.1097	.1069	.1049	.1029
4	.0000	.0000	.0000	.0000	.0055	.0067	.0068	.0069	.0651	.0757	.0752	.0744
5	.0000	.0000	.0000	.0000	.0010	.0017	.0017	.0018	.0326	.0536	.0543	.0544
6	.0000	.0000	.0000	.0000	—	.0004	.0004	.0005	—	.0376	.0392	.0399
7	.0000	.0000	.0000	.0000	—	.0001	.0001	.0001	—	.0257	.0280	.0293
8	.0000	.0000	.0000	.0000	—	.0000	.0000	.0000	—	.0169	.0198	.0215
9	.0000	.0000	.0000	.0000	—	.0000	.0000	.0000	—	.0103	.0136	.0156
10	.0000	.0000	.0000	.0000	—	.0000	.0000	.0000	—	.0053	.0089	.0112
11	.0000	.0000	.0000	.0000	—	—	.0000	.0000	—	—	.0055	.0080
12	.0000	.0000	.0000	.0000	—	—	.0000	.0000	—	—	.0028	.0055
13	.0000	.0000	.0000	.0000	—	—	—	.0000	—	—	—	.0036
14	.0000	.0000	.0000	.0000	—	—	—	.0000	—	—	—	.0022
15	.0000	.0000	.0000	.0000	—	—	—	.0000	—	—	—	.0011

improved as the truncation value κ increases. A comparison between $\kappa = 10$ and $\kappa = 15$ shows that the maximum errors of computational results are 2×10^{-4} for type 1 jobs, 1.1×10^{-3} for type 2 jobs, and 1.7×10^{-2} for type 3 jobs.

Besides, in order to measure the amount of computational effort, we record the numbers of iterations required for computing R and $\boldsymbol{x}_0$ before the error bound objective being satisfied, and the accumulated running time of our algorithm. The results are reported in Table 2, where we use NoI_1 and NoI_2 to represent the numbers of iterations in Algorithm 1 and 2, respectively.

7.3. Comparison of Numerical Results under Different Priority Disciplines

To reveal the impact of different priority disciplines on the performance measures, we study SPRM and PRPM numerically. Our results on the queue lengths and waiting time distributions are reported in Figures 1 and 2. The probability of $N_k = 0$ decreases as k ($k = 1, 2, 3$) increases. For type k ($k = 2, 3$) jobs, the probability of $N_k = 0$ under the standard preemptive resume discipline is greater than that under the preemptive repeat processing discipline. The waiting time of type k jobs is stochastically smaller than that of type $k + 1$ jobs, where $k = 1, 2$. For type k ($k = 2, 3$) jobs, the waiting time under the standard preemptive resume discipline is stochastically smaller than that under the preemptive repeat processing discipline. All of these are as we have expected.

Table 2. Numbers of iterations and running time required in computation.

κ	NoI_1	NoI_2	CPU time (secs)
5	13	154	8
10	13	402	118
12	13	514	426
15	13	683	12,531

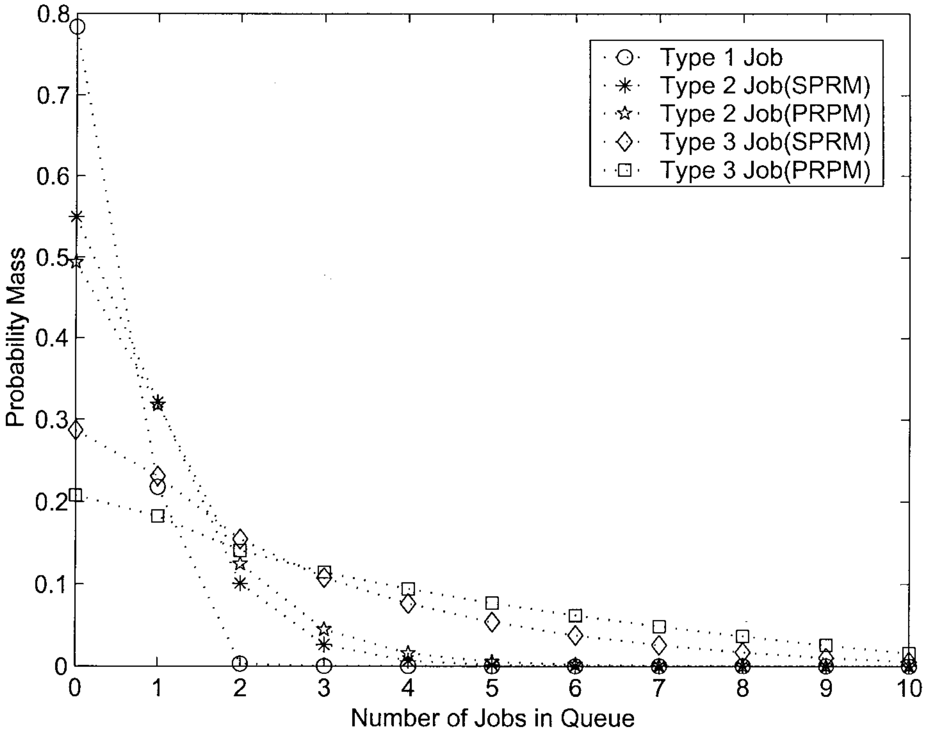

Figure 1. Probability distribution of queue lengths.

8. THE CASES WITH $K \geq 4$

Our study in previous sections is mainly focused on the case of $K = 3$. In this section, we shall show that the same approach can be used to investigate the cases of $K \geq 4$.

8.1. The Markov Chain

First, we point out that some notations [e.g., $N_k(t)$] in previous sections will be used with the same definition except that the subscript k is allowed to take values $1, 2, \dots, K$. Define that $\boldsymbol{N}(t) = (N_1(t), \dots, N_K(t))$ and $\boldsymbol{I}(t) = (I_1(t), \dots, I_K(t))$. The state space of Markov chain $\boldsymbol{X}(t) = (\boldsymbol{N}(t), J(t), \boldsymbol{I}(t))$ is $\Delta = \cup_{k=0}^{K} \Delta_k$, where

$$\begin{array}{lcl} \Delta_0 & = & \{(0, \dots, 0, j, 0, \dots, 0) \mid 1 \leq j \leq m\}, \\ \Delta_1 & = & \{(0, \dots, 0, n_K, j, 0, \dots, 0, i_K) \mid n_K \geq 1;\ 1 \leq j \leq m;\ 1 \leq i_K \leq m_K\}, \\ & \vdots & \\ \Delta_K & = & \left\{ (n_1, \dots, n_K, j, i_1, \dots, i_K) \;\middle|\; \begin{array}{l} n_1 \geq 1;\ n_2, \dots, n_K \geq 0;\ 1 \leq j \leq m; \\ i_1 \in \mathscr{C}_1^{n_1};\ i_2 \in \mathscr{C}_2^{n_2};\ \dots;\ i_K \in \mathscr{C}_K^{n_K} \end{array} \right\}. \end{array}$$

All states are labeled in lexicographic order. The transition matrix of $\boldsymbol{X}(t)$ is still given by (1), in which the block matrix $A^{(1)}_{n_1 \to n_1'}$ is composed by those probabilities of state transitions in the form: $(n_1, *, \dots, *) \to (n_1', *', \dots, *')$. More general, we define $A^{(k)}_{(n_1, \dots, n_k) \to (n_1', \dots, n_k')}$ ($k = 1, \dots, K$), which correspond to the state transitions of the following form:

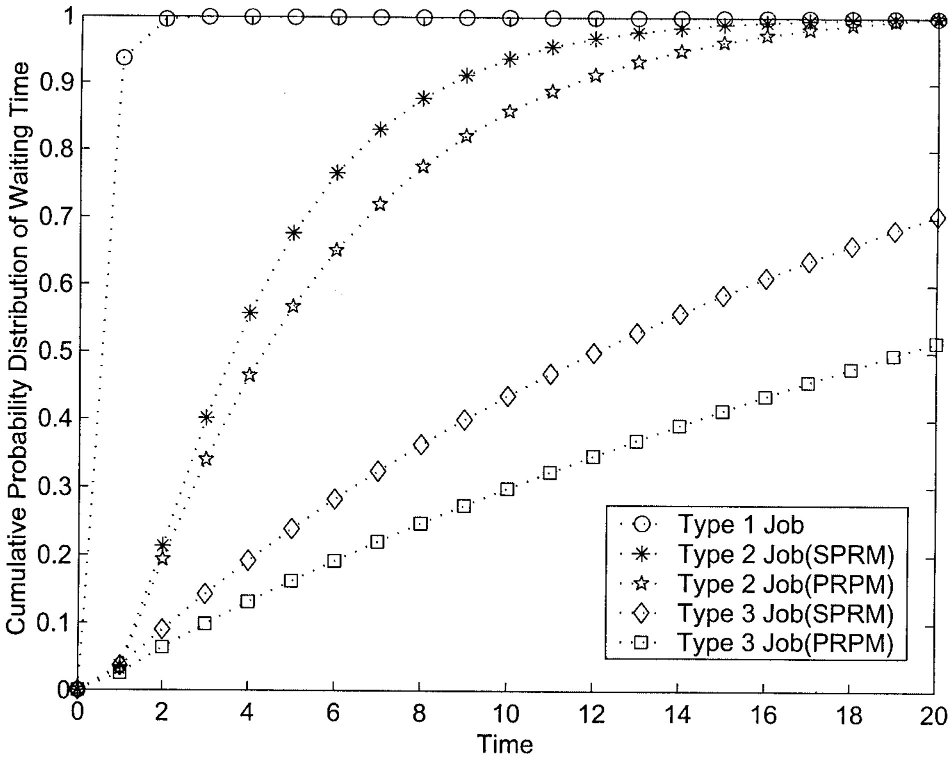

Figure 2. Cumulative probability distribution of waiting times.

$$(n_1, \ldots, n_k, *, \ldots, *) \rightarrow (n'_1, \ldots, n'_k, *', \ldots, *').$$

In what follows, we give a recusive formulae which allow us to obtain $A^{(k)}_{(n_1,\ldots,n_k)\rightarrow(n'_1,\ldots,n'_k)}$ from $A^{(k+1)}_{(n_1,\ldots,n_{k-1})\rightarrow(n'_1,\ldots,n'_{k+1})}$. Let $\boldsymbol{n}_{1:k} = (n_1, \ldots, n_k)$, $\boldsymbol{n}'_{1:k} = (n'_1, \ldots, n'_k)$, and $\mathbf{0}_{1:k}$ be a zero row vector of dimension k. For $\boldsymbol{n}_{1:k} = \boldsymbol{0}_{1:k}$, we have

$$A^{(k)}_{\mathbf{0}_{1:k}\rightarrow\boldsymbol{n}'_{1:k}} = \begin{bmatrix} A^{(k+1)}_{(\mathbf{0}_{1:k},0)\rightarrow(\boldsymbol{n}'_{1:k},0)} & A^{(k+1)}_{(\mathbf{0}_{1:k},0)\rightarrow(\boldsymbol{n}'_{1:k},1)} & & & \\ A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},0)} & A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},2)} & & \\ & A^{(k+1)}_{(\mathbf{0}_{1:k},2)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},2)} & \\ & \ddots & \ddots & \ddots & \\ & & A^{(k+1)}_{(\mathbf{0}_{1:k},2)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\mathbf{0}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},2)} \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad (31)$$

For $\boldsymbol{n}_{1:k} \neq \mathbf{0}_{1:k}$, we have

$$A^{(k)}_{\boldsymbol{n}_{1:k}\rightarrow\boldsymbol{n}'_{1:k}} = \begin{bmatrix} A^{(k+1)}_{(\boldsymbol{n}_{1:k},0)\rightarrow(\boldsymbol{n}'_{1:k},0)} & A^{(k+1)}_{(\boldsymbol{n}_{1:k},0)\rightarrow(\boldsymbol{n}'_{1:k},1)} & & \\ & A^{(k+1)}_{(\boldsymbol{n}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\boldsymbol{n}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},2)} & \\ & & A^{(k+1)}_{(\boldsymbol{n}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},1)} & A^{(k+1)}_{(\boldsymbol{n}_{1:k},1)\rightarrow(\boldsymbol{n}'_{1:k},2)} \\ & & \ddots & \ddots \end{bmatrix}. \quad (32)$$

Therefore, the transition matrix P is completely determined by the block matrices $A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}}$, which are written out in Appendix B.

8.2. Structure and Computation of Matrix *R*

As for the structure of matrix R, Eqs. (7) and (8) always holds. Moreover, the similar arguments leading to that matrix $R_{(n_2,n_3),(n'_2,n'_3)}$ can be further partitioned into smaller blocks. More general, for $k = 2, 3, \ldots, K - 1$, there are the following relations between $R_{(n_2,\ldots,n_k)\to(n'_2,\ldots,n'_k)}$ and $R_{(n_2,\ldots,n_{k+1})\to(n'_2,\ldots,n'_{k+1})}$:

$$R_{\boldsymbol{n}_{2:k},\boldsymbol{n}'_{2:k}} = \begin{bmatrix} R_{(\boldsymbol{n}_{2:k},0),(\boldsymbol{n}'_{2:k},0)} & R_{(\boldsymbol{n}_{2:k},0),(\boldsymbol{n}'_{2:k},1)} & R_{(\boldsymbol{n}_{2:k},0),(\boldsymbol{n}'_{2:k},2)} & R_{(\boldsymbol{n}_{2:k},0),(\boldsymbol{n}'_{2:k},3)} & \cdots \\ & R_{(\boldsymbol{n}_{2:k},1),(\boldsymbol{n}'_{2:k},1)} & R_{(\boldsymbol{n}_{2:k},1),(\boldsymbol{n}'_{2:k},2)} & R_{(\boldsymbol{n}_{2:k},1),(\boldsymbol{n}'_{2:k},3)} & \cdots \\ & & R_{(\boldsymbol{n}_{2:k},1),(\boldsymbol{n}'_{2:k},1)} & R_{(\boldsymbol{n}_{2:k},1),(\boldsymbol{n}'_{2:k},2)} & \cdots \\ & & & R_{(\boldsymbol{n}_{2:k},1),(\boldsymbol{n}'_{2:k},1)} & \cdots \\ & & & & \ddots \end{bmatrix}, \quad (33)$$

where $\boldsymbol{n}_{2:k} = (n_2, n_3, \ldots, n_k)$, $\boldsymbol{n}'_{2:k} = (n'_2, n'_3, \ldots, n'_k) = \boldsymbol{n}_{2:k} + \boldsymbol{h}_{2:k}$, $\boldsymbol{h}_{2:k} = (h_2, h_3, \ldots, h_k)$, and $n_i \in \{0, 1\}$, $h_i \ge 0$ for all $i = 2, 3, \ldots, k$. It is worthwhile to note that the order of matrix $R_{\boldsymbol{n}_{2:K},\boldsymbol{n}'_{2:K}}$ is $m \prod_{k=2}^{K} \zeta_k(n_k) \times m \prod_{k=2}^{K} \zeta_k(n'_k)$.

Following the steps presented in subsection 5.3, one can write out an iterative expression of $R_{\boldsymbol{n}_{2:K},\boldsymbol{n}'_{2:K}}$ although there are some cumbersome derivations involved. Furthermore, the stationary distribution and the performance measures can be obtained in a manner similar to that presented in subsection 5.4 and in Section 6, respectively.

APPENDIX A: BLOCK MATRICES $A^{(3)}_{(n_1,n_2,n_3)\to(n'_1,n'_2,n'_3)}$ FOR CASE $K = 3$

We shall use notations $\boldsymbol{n} = (n_1, n_2, n_3)$, $\boldsymbol{n}' = (n'_1, n'_2, n'_3)$, and $\boldsymbol{v} = (v_1, v_2, v_3) \in \Pi$. Notice that the possible state transitions $\boldsymbol{n} \to \boldsymbol{n}'$ should satisfy one of the following conditions:

Condition A. $\boldsymbol{n} = (0, 0, 0)$, $\boldsymbol{n}' = (v_1, v_2, v_3)$.
Condition B. $\boldsymbol{n} = (0, 0, n_3)$, $n_3 \ge 1$, $\boldsymbol{n}' = (v_1, v_2, n'_3)$, $n'_3 \in \{n_3 - 1, n_3, n_3 + 1\}$.
Condition C. $\boldsymbol{n} = (0, n_2, n_3)$, $n_2 \ge 1$, $\boldsymbol{n}' = (v_1, n'_2, n_3 + v_3)$, $n'_2 \in \{n_2 - 1, n_2, n_2 + 1\}$.
Condition D. $\boldsymbol{n} = (n_1, n_2, n_3)$, $n_1 \ge 1$, $\boldsymbol{n}' = (n'_1, n_2 + v_2, n_3 + v_3)$, $n'_1 \in \{n_1 - 1, n_1, n_1 + 1\}$.

In other words, $A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} \equiv 0$ for all those $\boldsymbol{n}$ and $\boldsymbol{n}'$ not satisfying these conditions. To write out $A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'}$ for all possible cases, we define the following auxiliary function as preliminary:

$$\Gamma_k(n_k, n'_k) = \begin{cases} 1, & \text{if } n_k = 0, \quad n'_k = 0, \\ \boldsymbol{\beta}_k, & \text{if } n_k = 0, \quad n'_k = 1, \\ \mathcal{I}_{m_k}, & \text{if } n_k \ge 1. \end{cases}$$

Now we are ready to write out $A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'}$ as follows.

CASE A: $\boldsymbol{n} = (0, 0, 0)$, $\boldsymbol{n}' = \boldsymbol{v}$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_v, & \text{if } \boldsymbol{v} = (0,0,0), \\ D_v \otimes \boldsymbol{\beta}_3, & \text{if } \boldsymbol{v} = (0,0,1), \\ D_v \otimes \boldsymbol{\beta}_2 \otimes \Gamma_3(0, v_3), & \text{if } \boldsymbol{v} = (0,1,v_3), \\ D_v \otimes \boldsymbol{\beta}_1 \otimes \Gamma_2(0, v_2) \otimes \Gamma_3(0, v_3), & \text{if } \boldsymbol{v} = (1, v_2, v_3). \end{cases}$$

CASE B1: $\boldsymbol{n} = (0, 0, n_3)$, $n_3 \ge 1$, $\boldsymbol{n}' = (v_1, v_2, n_3 + 1)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,1)} \otimes S_3, & \text{if } v_1 = v_2 = 0, \\ D_{(0,1,1)} \otimes \boldsymbol{\beta}_2 \otimes (S_3 Q_3), & \text{if } v_1 = 0, \quad v_2 = 1, \\ D_{(1,v_2,1)} \otimes \boldsymbol{\beta}_1 \otimes \Gamma_2(0, v_2) \otimes (S_3 Q_3), & \text{if } v_1 = 1. \end{cases}$$

CASE B2: $\boldsymbol{n} = (0, 0, n_3)$, $n_3 \geq 1$, $\boldsymbol{n}' = (v_1, v_2, n_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,0)} \otimes S_3 + D_{(0,0,1)} \otimes (\boldsymbol{S}_3^0 \boldsymbol{\beta}_3), & \text{if } v_1 = v_2 = 0, \\ D_{(0,1,0)} \otimes \boldsymbol{\beta}_2 \otimes (S_3 Q_3) + D_{(0,1,1)} \otimes \boldsymbol{\beta}_2 \otimes (\boldsymbol{S}_3^0 \boldsymbol{\beta}_3), & \text{if } v_1 = 0, \quad v_2 = 1, \\ D_{(1,v_2,0)} \otimes \boldsymbol{\beta}_1 \otimes \Gamma_2(0, v_2) \otimes (S_3 Q_3) & \\ \qquad + D_{(1,v_2,1)} \otimes \boldsymbol{\beta}_1 \otimes \Gamma_2(0, v_2) \otimes (\boldsymbol{S}_3^0 \boldsymbol{\beta}_3), & \text{if } v_1 = 1. \end{cases}$$

CASE B3: $\boldsymbol{n} = (0, 0, n_3)$, $n_3 \geq 2$, $\boldsymbol{n' = (v_1, v_2, n_3 - 1)}$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,0)} \otimes (\boldsymbol{S}_3^0 \boldsymbol{\beta}_3), & \text{if } v_1 = v_2 = 0, \\ D_{(0,1,0)} \otimes \boldsymbol{\beta}_2 \otimes (\boldsymbol{S}_3^0 \boldsymbol{\beta}_3), & \text{if } v_1 = 0, \quad v_2 = 1, \\ D_{(1,v_2,0)} \otimes \boldsymbol{\beta}_1 \otimes \Gamma_2(0, v_2) \otimes (\boldsymbol{S}_3^0 \boldsymbol{\beta}_3), & \text{if } v_1 = 1. \end{cases}$$

CASE B4: $\boldsymbol{n} = (0, 0, 1)$, $\boldsymbol{n}' = (v_1, v_2, 0)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,0)} \otimes \boldsymbol{S}_3^0, & \text{if } v_1 = v_2 = 0, \\ D_{(0,1,0)} \otimes \boldsymbol{\beta}_2 \otimes \boldsymbol{S}_3^0, & \text{if } v_1 = 0, \quad v_2 = 1, \\ D_{(1,v_2,0)} \otimes \boldsymbol{\beta}_1 \otimes \Gamma_2(0, v_2) \otimes \boldsymbol{S}_3^0, & \text{if } v_1 = 1. \end{cases}$$

CASE C1: $\boldsymbol{n} = (0, n_2, n_3)$, $n_2 \geq 1$, $\boldsymbol{n}' = (v_1, n_2 + 1, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,1,v_3)} \otimes S_2 \otimes \Gamma_3(n_3, n_3'), & \text{if } v_1 = 0, \\ D_{(1,1,v_3)} \otimes \boldsymbol{\beta}_1 \otimes (S_2 Q_2) \otimes \Gamma_3(n_3, n_3'), & \text{if } v_1 = 1. \end{cases}$$

CASE C2: $\boldsymbol{n} = (0, n_2, n_3)$, $n_2 \geq 1$, $\boldsymbol{n}' = (v_1, \mathrm{n}_2, \mathrm{n}_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} (D_{(0,0,v_3)} \otimes S_2 + D_{(0,1,v_3)} \otimes (\boldsymbol{S}_2^0 \boldsymbol{\beta}_2)) \otimes \Gamma_3(n_3, n_3'), & \text{if } v_1 = 0, \\ (D_{(1,0,v_3)} \otimes \boldsymbol{\beta}_1 \otimes (S_2 Q_2) + & \\ \qquad D_{(1,1,v_3)} \otimes \boldsymbol{\beta}_1 \otimes (\boldsymbol{S}_2^0 \boldsymbol{\beta}_2) \otimes \Gamma_3(n_3, n_3'), & \text{if } v_1 = 1. \end{cases}$$

CASE C3: $\boldsymbol{n} = (0, n_2, n_3)$, $n_2 \geq 2$, $\boldsymbol{n}' = (v_1, n_2 - 1, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,v_3)} \otimes (\boldsymbol{S}_2^0 \boldsymbol{\beta}_2) \otimes \Gamma_3(n_3, n_3'), & \text{if } v_1 = 0, \\ D_{(1,0,v_3)} \otimes \boldsymbol{\beta}_1 \otimes (\boldsymbol{S}_2^0 \boldsymbol{\beta}_2) \otimes \Gamma_3(n_3, n_3'), & \text{if } v_1 = 1. \end{cases}$$

CASE C4: $\boldsymbol{n} = (0, 1, n_3)$, $n_3 \geq 1$, $\boldsymbol{n}' = (v_1, 0, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,v_3)} \otimes \boldsymbol{S}_2^0 \otimes \mathcal{I}_{m_3}, & \text{if } v_1 = 0, \\ D_{(1,0,v_3)} \otimes \boldsymbol{\beta}_1 \otimes \boldsymbol{S}_2^0 \otimes \mathcal{I}_{m_3}, & \text{if } v_1 = 1. \end{cases}$$

CASE C5: $\boldsymbol{n} = (0, 1, 0)$, $\boldsymbol{n}' = (v_1, 0, v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,0)} \otimes \boldsymbol{S}_2^0, & \text{if } v_1 = v_3 = 0, \\ D_{(0,0,1)} \otimes \boldsymbol{S}_2^0 \otimes \boldsymbol{\beta}_3, & \text{if } v_1 = 0, \quad v_3 = 1, \\ D_{(1,0,v_3)} \otimes \boldsymbol{\beta}_1 \otimes \boldsymbol{S}_2^0 \otimes \Gamma_3(0, v_3), & \text{if } v_1 = 1. \end{cases}$$

CASE D1: $\boldsymbol{n} = (n_1, n_2, n_3)$, $n_1 \geq 1$, $\boldsymbol{n}' = (n_1 + 1, n_2 + v_2, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = D_{(1,v_2,v_3)} \otimes S_1 \otimes \Gamma_2(n_2, n_2') \otimes \Gamma_3(n_3, n_3').$$

CASE D2: $\boldsymbol{n} = (n_1, n_2, n_3)$, $n_1 \geq 1$, $\boldsymbol{n}' = (n_1, n_2 + v_2, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = (D_{(0,v_2,v_3)} \otimes S_1 + D_{(1,v_2,v_3)} \otimes (\boldsymbol{S}_1^0 \boldsymbol{\beta}_1)) \otimes \Gamma_2(n_2, n_2') \otimes \Gamma_3(n_3, n_3').$$

CASE D3: $\boldsymbol{n} = (n_1, n_2, n_3)$, $n_1 \geq 2$, $\boldsymbol{n}' = (n_1 - 1, n_2 + v_2, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = D_{(0,v_2,v_3)} \otimes (\boldsymbol{S}_1^0 \boldsymbol{\beta}_1) \otimes \Gamma_2(n_2, n_2') \otimes \Gamma_3(n_3, n_3').$$

CASE D4: $\boldsymbol{n} = (1, n_2, n_3)$, $n_2 \geq 1$, $\boldsymbol{n}' = (0, n_2 + v_2, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = D_{(0,v_2,v_3)} \otimes \boldsymbol{S}^0_1 \otimes \mathcal{I}_{m_2} \otimes \Gamma_3(n_3, n_3').$$

CASE D5: $\boldsymbol{n} = (1, 0, n_3)$, $n_3 \geq 1$, $\boldsymbol{n}' = (0, v_2, n_3 + v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,v_3)} \otimes \boldsymbol{S}^0_1 \otimes \mathcal{I}_{m_3}, & \text{if } v_2 = 0, \\ D_{(0,1,v_3)} \otimes \boldsymbol{S}^0_1 \otimes \boldsymbol{\beta}_2 \otimes \mathcal{I}_{m_3}, & \text{if } v_2 = 1. \end{cases}$$

CASE D6: $\boldsymbol{n} = (1, 0, 0)$, $\boldsymbol{n}' = (0, v_2, v_3)$,

$$A^{(3)}_{\boldsymbol{n}\to\boldsymbol{n}'} = \begin{cases} D_{(0,0,0)} \otimes \boldsymbol{S}^0_1, & \text{if } v_2 = v_3 = 0, \\ D_{(0,0,1)} \otimes \boldsymbol{S}^0_1 \otimes \boldsymbol{\beta}_3, & \text{if } v_2 = 0, \quad v_3 = 1, \\ D_{(0,1,v_3)} \otimes \boldsymbol{S}^0_1 \otimes \boldsymbol{\beta}_2 \otimes \Gamma_3(0, v_3), & \text{if } v_2 = 1. \end{cases}$$

APPENDIX B: BLOCK MATRICES $A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}}$ FOR CASE $K \geq 4$

It is immediately clear that all matrices $A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}}$ are of finite dimension and their dimension are different for different $(n_1, \ldots, n_K)$ and $(n_1', \ldots, n_K')$. It is because the number of possible values taken by $I_k(t)$ depends on the value of $N_k(t)$, $1 \leq k \leq K$. Specifically, $I_k(t)$ take value 0 if the number of type-k jobs is zero at t; otherwise, $I_k(t)$ may take m_k different values $1, 2, \ldots, m_k$.

There are a number of cases to be considered for writing out $A^{(K)}_{\boldsymbol{n}_{1:K}}$. But we only need to consider those nonzero cases. We shall use a simple notation Γ_k to represent $\Gamma_k(n_k, n_k')$. Let $\boldsymbol{e}_k^T = (\underbrace{0, \ldots, 0, 1}_{k}, 0, \ldots, 0)$ and $\boldsymbol{v}_{1:K} = (v_1, \ldots, v_K) \in \Pi$.

CASE 1: $\boldsymbol{n}_{1:K} = \boldsymbol{0}_{1:K}$, $\boldsymbol{n}'_{1:K} = \boldsymbol{v}_{1:K}$,

$$A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}} = D_{v_{1:K}} \otimes \Gamma_1 \otimes \cdots \otimes \Gamma_K.$$

CASE 2: $\boldsymbol{n}_{1:K} \neq \boldsymbol{0}_{1:K}$, $s = \min\{k \mid n_k \geq 1\}$, $n_s' = n_s + 1$:

(2a) If $\min\{k \mid v_k = 1\} = s$, then

$$A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}} = D_{v_{1:K}} \otimes S_s \otimes \Gamma_{s+1} \otimes \cdots \otimes \Gamma_K.$$

(2b) If $\min\{k \mid v_k = 1\} \leq s - 1$, then

$$A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}} = D_{\mathbf{v_{1:K}}} \otimes \boldsymbol{\Gamma_1} \otimes \cdots \otimes \boldsymbol{\Gamma_{s-1}} \otimes (\mathbf{S_s Q_s}) \otimes \boldsymbol{\Gamma_{s+1}} \otimes \cdots \otimes \boldsymbol{\Gamma_K}.$$

CASE 3: $\boldsymbol{n}_{1:K} \neq \boldsymbol{0}_{1:K}$, $s = \min\{k \mid n_k \geq 1\}$, $n_s' = n_s$:

(3a) If $\boldsymbol{v}_{1:K} = \boldsymbol{0}_{1:K}$ or $\boldsymbol{v}_{1:K} \neq \boldsymbol{0}_{1:K}$ and $\min\{k|v_k = 1\} \geq s + 1$, then

$$A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}} = (D_{v_{1:K}} \otimes S_s + D_{\mathbf{v_{1:K}+e_s^T}} \otimes (\boldsymbol{S}^0_s\boldsymbol{\beta}_s)) \otimes \Gamma_{s+1} \otimes \cdots \otimes \Gamma_K.$$

(3b) If $\boldsymbol{v}_{1:K} \neq \boldsymbol{0}_{1:K}$ and $\min\{k \mid v_k = 1\} \leq s - 1$, then

$$A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}} = (D_{v_{1:K}} \otimes \Gamma_1 \otimes \cdots \otimes \Gamma_{s-1} \otimes (S_s Q_s) + D_{v_{1:K}+\boldsymbol{e}_s} \otimes \Gamma_1 \otimes \cdots \otimes \Gamma_{s-1} \otimes (\boldsymbol{S}^0_s\boldsymbol{\beta}_s)) \otimes \Gamma_{s+1} \otimes \cdots \otimes \Gamma_K.$$

CASE 4: $\boldsymbol{n}_{1:K} \neq \boldsymbol{0}_{1:K}$, $s = \min\{k \mid n_k \geq 1\}$, $n_s' = n_s - 1$:

(4a) If $n_s \geq 2$, then

$$A^{(K)}_{\boldsymbol{n}\to\boldsymbol{n}'} = D_{v_{1:K}} \otimes \Gamma_1 \otimes \cdots \otimes \Gamma_{s-1} \otimes (\boldsymbol{S}^0_s\boldsymbol{\beta}_s) \otimes \Gamma_{s+1} \otimes \cdots \otimes \Gamma_K.$$

(4b) If $n_s = 1$, then

$$A^{(K)}_{\boldsymbol{n}_{1:K}\to\boldsymbol{n}'_{1:K}} = D_{v_{1:K}} \otimes \Gamma_1 \otimes \cdots \otimes \Gamma_{s-1} \otimes \boldsymbol{S}^0_s \otimes \Gamma_{s+1} \otimes \cdots \otimes \Gamma_K.$$

ACKNOWLEDGMENTS

The authors thank the anonymous referees for the comments which helped to improve the presentation of the paper. The research of A.S. Alfa was supported in part by Grant No. OGP0006584 from the Natural Sciences and Engineering Research Council of Canada and that of Q-M. He was also supported in part by Natural Science and Engineering Research Council of Canada.

REFERENCES

[1] A.S. Alfa, Matrix-geometric solution of discrete time $MAP/PH/1$ priority queue, Naval Res Logist 45 (1998), 23–50.
[2] K.E. Atkinson, An introduction to numerical analysis, 2nd edition, Wiley, New York, 1988.
[3] N.U. Bhat, Elements of applied stochastic processes, Wiley, New York, 1972.
[4] G. Choudhury, D. Lucantoni, and W. Whitt, Multidimensional transform inversion with applications to the transient $M/G/1$ queue, Ann Appl Probab 4(3) (1994), 719–740.
[5] H.R. Gail, S.L. Hantler, and B.A. Taylor, Analysis of a non-preemptive priority multiserver queue, Adv in Appl Probab 20 (1988), 852–879.
[6] F.R. Gantmacher, The theory of matrices, Chelsea, New York, 1959.
[7] Q.-M. He and M.F. Neuts, Markov arrival processes with marked transitions, Stochastic Process Appl 74 (1998), 37–52.
[8] E.P.C. Kao and K.S. Narayanan, Computing steady-state probabilities of a non-preemptive priority multiserver queue, ORSA J Comput 2 (1990), 211–218.
[9] D.M. Lucantoni, New results on the single server queue with a batch Markovian arrival process, Stoch Models 7 (1991), 1–46.
[10] D.G. Miller, Computation of steady-state probabilities for $M/M/1$ priority queues, Oper Res 29(5) (1981), 945–958.
[11] M.F. Neuts, A versatile Markovian point process, J Appl Probab 16 (1979), 764–779.
[12] M.F. Neuts, Matrix-geometric solutions in stochastic models, Johns Hopkins University Press, Baltimore, 1981.
[13] M.F. Neuts, Structured stochastic matrices of the $M/G/1$ types and their applications, Marcel Dekker, New York, 1989.
[14] M.F. Neuts, Models based on the Markovian arrival process, IEICE Trans Commun E75-B (1992), 1255–1265.
[15] T. Takine and T. Hasegawa, The workload in the $MAP/G/1$ queue with state-dependent services: Its application to a queue with preemptive resume priority, Stoch Models 10(1) (1994), 183–204.
[16] T. Takine, A nonpreemptive priority $MAP/G/1$ queue with two classes of customers, J Oper Res Soc of Japan, 39(2) (1996), 266–290.
[17] T. Takine and T. Hasegawa, The nonpreemptive priority $MAP/G/1$ queue, Oper Res 47(6) (1999), 917–927.
[18] R.L. Tweedie, Operator-geometric stationary distributions for Markov Chains, with applications to queueing models, Adv Appl Prob 14(2) (1982), 368–391.
[19] D. Wagner, Analysis of a multiserver model with non-preemptive priorities and non-renewal input, Teletraffic in the Evolution of Telecommunications Networks, J. Labetoulle and J.W. Roberts (Editors), Elsevier, Amsterdam, 1994, pp. 757–766.