Vietnam J Comput Sci (2014) 1:71–78
DOI 10.1007/s40595-013-0005-2

REVIEW

# Analysis of a queue with two priority classes and feedback controls

**Hung T. Tran · Tien V. Do · Laszlo Pap**

Received: 24 September 2013 / Accepted: 5 October 2013 / Published online: 23 November 2013


**Abstract** In this paper, a queueing system with two priority classes of customers is investigated. In the queueing system, customers of high priority can preempt the service of customers of low priority. Customers can wait in buffers of finite size. A congestion control mechanism is proposed to model various practical problems. The steady-state probabilities are computed with the help of the framework of quasi-birth–death processes using the theory of generalized invariant subspace. Performance measures of interest are also derived and demonstrated by numerical results.

**Keywords** Priority · Feedback control · Queue · QBD

## 1 Introduction

The preemptive queue with two priority classes has been treated in earlier works [10,11,13] because it can be used for the performance evaluation of practical systems. Customers of two classes (high and low priority) arrive into the system. A server has a service capacity of multiple identical service units. Each high-priority customer requires a single service unit. The system works under the preemptive discipline, i.e. an admitted high-priority customer immediately gets service upon his arrival either by getting an idle service unit or by occupying service unit being used by low-priority customers. High-priority customers are served according to the first in first out (FIFO) and low-priority customers are served according to the processor sharing (PS) principle. Furthermore, the service capacity for low-priority customers depends on the number of high-priority customers in the system.

To take into account practical aspects, buffers of finite size for both two classes of customers are assumed in this paper. Furthermore, a proposed queue includes a congestion control mechanism. Once the number of low-priority customers in the system reaches the first control threshold, the arrival rate of low-priority customers is reduced to a guarantee arrival rate. If the number of low-priority customers is above the second control threshold, each arriving batch of low-priority customers will be discarded with a certain probability. Therefore, the proposed queue can be used to model Differentiated Service Architecture [14] in IP networks.

We show that this queue can be analyzed within the framework of Quasi Birth and Death (QBD) processes [4,5,7,8]. Closed-form expressions are given for the mean number of customers of each class, the batch and the customer loss probability of each class and the mean system time of customers.

The rest of the paper is organized as follows. In Sect. 2, a queue with two priority classes is discussed in details. Finally, numerical results illustrating the operation of the system are presented in Sect. 3. The paper ends with some conclusions in Sect. 4.

## 2 A system model and performance analysis

The system consists of a server and two finite queues. The service capacity of the server is assumed to be composed of $N$ service units. There are two categories of customers: high priority (HP) and low priority (LP).

We assume that the interarrival times of batches of HP customers follow an exponential distributions with parameter

H. T. Tran · T. V. Do (✉) · L. Pap
Budapest University of Technology and Economics,
Budapest, Hungary
e-mail: do@hit.bme.hu

H. T. Tran · T. V. Do
Viet-Hung Industrial University, Hanoi City, Vietnam

$\lambda_{\text{HP}}$. The size of the buffer for storing HP customers is $K_{\text{HP}}$. The batch size is upper-bounded by $H$ (note that $K_{\text{HP}} \geq H$). A batch of $b$, $1 \leq b \leq H$, HP customers occurs with probability $h_b$, where $\sum_{b=1}^{H} h_b = 1$. Each HP customer requires a single service unit. The service time of a HP customer has an exponential distribution with rate $\mu_{\text{HP}}$. High-priority customers are served according to the FIFO principle.

Let $I(t)$ denote the number of high-priority customers in the system at time $t$, $0 \leq I(t) \leq \mathcal{N} = N + K_{\text{HP}}$. Upon the arrival of a batch of $b$ HP customers at time $t$,

- if all the $N$ service units are occupied by high-priority customers, then the batch or a portion of the batch is put into the buffer of size $K_{\text{HP}}$, or the whole batch is rejected. A decision regarding the batch depends on the occupation level of the buffer and an applied admission rule.
- if the number $i$ of service units occupied by other high-priority customers is less than $N$ (i.e., $i < N$) and $b \leq N - i$, $i$ HP customers is immediately served. Note that it is assumed that high-priority customers are allowed to push out low-priority customers being in service upon the arrival of high-priority customers.
- if the number $i$ of service units occupied by other high-priority customers is less than $N$ (i.e., $i < N$) and if $b > N - i$, the server is occupied by $N - i$ HP customers from the batch and $b - N + i$ HP customers will be tried to be placed in the buffer.

Two admission rules can be considered for the placement of customers into the buffer: whole batch acceptance (WBA) or partially batch acceptance (PBA). WBA means that the whole batch will be dropped if the system is not able to accept the whole batch. Under PBA only the part of batch is dropped, for which available room cannot be assured.

Batches of low-priority customers arrive according to a Poisson process. The batch size is upper-bounded by $L$. A batch of $b$, $1 \leq b \leq L$, LP customers occurs with probability $l_b$, where $\sum_{b=1}^{L} l_b = 1$. The service discipline of LP customers is processor sharing, a low-priority batch will get service immediately if there is service capacity not used by HP customers. Otherwise the batch is put into a buffer of size $K_{\text{LP}}$. Due to the preemptive nature of the system, low-priority customers under service may be pushed back into waiting room by high-priority customers. It is highly recommended that once a customer has been accepted into system, it will not be lost because of the aforementioned push-back mechanism. Therefore, at any time we require that no more than $K_{\text{LP}}$ of low-priority customers (either being in service or waiting in queue) are in the system. Let $J(t)$ be the number of low-priority customers in the system, $0 \leq J(t) \leq K_{\text{LP}}$. If there are $i$ high-priority and $j$ low-priority customers in the system ($0 \leq i \leq N + K_{\text{HP}}$, $j \geq 1$), then each of the low-priority customers receives service at rate $\frac{\max(N-i,0)}{j}\mu_{\text{LP}}$.

**Table 1** Parameters of the considered queue

| | |
|---|---|
| $\lambda_{\text{HP}}$ | Arrival rate of high-priority batches |
| $\lambda_{\text{LP}}$ | Minimum arrival rate of low-priority batches |
| $H$ | Upper bound of high-priority batches |
| $L$ | Upper bound of low-priority batches |
| $K_{\text{HP}}$ | Buffer size for the high-priority class |
| $K_{\text{LP}}$ | Buffer size for the low-priority class |
| $m_1$ | The first control level for low-priority class |
| $m_2$ | The second control level $K_{\text{LP}} - m_2$ |
| $p_{\text{dropp}}$ | Dropping probability used for overload control |

It is assumed that batches of low-priority customers arrive at rate $\lambda_{LP,j}$ for level $j$ ($0 \leq j < m_1$). When the number of low-priority customers in the system reaches the level $m_1$, its (batch) arrival rate is forced to reduce from $\lambda_{L,j}$ to the guarantee value $\lambda_{\text{LP}}$. From the context of congestion control, it is natural to assume $\lambda_{\text{LP},0} \geq \lambda_{\text{LP},1} \geq \cdots \geq \lambda_{\text{LP}}$.

The second control level is set to $j = K_{\text{LP}} - m_2$. That means whenever the difference between the number of low-priority customers and the waiting room's capacity drops below $m_2$, each low-priority arrival batch will be discarded with the pre-defined probability $p_{\text{dropp}}$. The main parameters of the queue are summarized in Table 1.

The system is two-dimensional Markov process $(I(t), J(t)) = (i, j)$ with the state space $\{(i, j) : 0 \leq i \leq K_{\text{HP}} + N,\ 0 \leq j \leq K_{\text{LP}}\}$. Let $p_{i,j}$ denote the steady state probability of the state $(i, j)$ as

$$p_{i,j} = \lim_{t\to\infty} \Pr(I(t) = i, J(t) = j), \quad (i = 0, \ldots, N + K_{\text{HP}};\ j = 0, 1, \ldots, K_{\text{LP}}).$$

We introduce the following vectors

$$\mathbf{v}_j = (p_{0,j}, p_{1,j}, \ldots, p_{\mathcal{N},j}) \quad \text{for} \quad j = 0, 1, \ldots K_{\text{LP}}.$$

In what follows, we provide an analysis for the WBA rule. Note that the analysis for the PBA rule can be performed in the same way. For the WBA rule, the generator matrix of this process will have the form shown in Fig. 1.

For $0 \leq j \leq K_{\text{LP}}$, we define the following matrix $\mathcal{A}_j = A_j - D^{A_j} - C_j - \sum_{s=1}^{L} B_{j-s,s}$, where $D^{A_j}$ is matrix whose diagonal elements are the sum of the elements in the corresponding row of $A_j$.

The balance equations of the system are written as

$$\sum_{s=1}^{L} \mathbf{v}_{j-s} B_{j-s,s} + \mathbf{v}_j \mathcal{A}_j + \mathbf{v}_{j+1} C_j = 0, \tag{1}$$

and the normalization equation is

$$\sum_{j=0}^{K_{\text{LP}}} \mathbf{v}_j \mathbf{e} = 1.0, \tag{2}$$

where $\mathbf{e}$ is the unit vector.

**Fig. 1** The block structure of the generator matrix of the queueing system ($m_1 = 3, m_2 = 4,\ L = 2$)

$$Q = \left[\begin{array}{cccc|cc|cc}
\mathcal{A}_0 & B_{0,1} & B_{0,2} & & & & & \\
C_1 & \mathcal{A}_1 & B_{1,1} & B_{1,2} & & & & \\
 & C_1 & \mathcal{A}_2 & B_{2,1} & B_{2,2} & & & \\
\hline
 & & C_1 & \mathcal{A}_3 & B_1 & B_2 & & \\
 & & & C_1 & \mathcal{A}_4 & B_1 & B_2 & \\
 & & & & C_1 & \mathcal{A}_5 & B_1 & B_2 \\
 & & & & & C_1 & \mathcal{A}_6 & B_1 \\
 & & & & & & \ddots & \ddots \\
 & & & & & & C_1 & \mathcal{A}_{K_{LP}-4} \\
\hline
 & & & & & & C_1 & \mathcal{A}_{K_{LP}-3} \\
 & & & & & & & C_1 \\
 & & & & & & & \\
 & & & & & & & \\
\end{array}\right.
\left.\begin{array}{cc}
 & \\
 & \\
 & \\
\hline
 & \\
 & \\
 & \\
 & \\
 & \\
 & \\
\hline
B_1 & B_2 \\
B_{K_{LP}-3,1} & B_{K_{LP}-3,2} \\
\mathcal{A}_{K_{LP}-2} & B_{K_{LP}-2,2} \\
C_1 & \mathcal{A}_{K_{LP}-1} \\
 & C_1 \\
 & \mathcal{A}_{K_{LP}}
\end{array}\right]$$

**Fig. 2** Examples of transition matrices

$$A_j = A = \begin{bmatrix}
0 & h_1\lambda_{HP} & h_2\lambda_{HP} & \dots & h_H\lambda_{HP} & & & & & 0 \\
\mu_{HP} & 0 & h_1\lambda_{HP} & \dots & \dots & h_H\lambda_{HP} & & & & 1 \\
 & \ddots & \ddots & \ddots & & & & & & \vdots \\
 & & i\mu_{HP} & 0 & h_1\lambda_{HP} & \dots & & & h_H\lambda_{HP} & i \\
 & & & & \ddots & \ddots & \ddots & & & \vdots \\
 & & & & N\mu_{HP} & 0 & & & h_{K_{HP}}\lambda_{K_{HP}} & N \\
 & & & & & \ddots & \ddots & \vdots & & \\
 & & & & & & & h_1\lambda_{HP} & & \vdots \\
 & & & & & & N\mu_{HP} & 0 & & N + K_{HP}
\end{bmatrix}$$

$$B_{j,s} = \begin{bmatrix}
l_s\lambda_{L,j} & & & \\
 & l_s\lambda_{L,j} & & \\
 & & \ddots & \\
 & & & l_s\lambda_{L,j}
\end{bmatrix} \quad \text{for } 0 \le j \le K_{LP}, 1 \le s \le L$$

$$C_{j,1} = C_1 = \begin{bmatrix}
N\mu_L & & & & & & 0 \\
 & \ddots & & & & & \vdots \\
 & & 2\mu_L & & & & N-2 \\
 & & & \mu_L & & & N-1 \\
 & & & & 0 & & N \\
 & & & & & \ddots & \vdots \\
 & & & & & & 0 \\
\end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ N + K_{HP} \end{matrix}$$

For $M_1 = m_1 + L \le j \le M_2 = \min(K_L - 1, K_L - m2)$, the Eq. (1) include only $j$-independent coefficient matrices and have a common form

$$\sum_{s=1}^{L} \mathbf{v}_{j-s} B_s + \mathbf{v}_j \mathcal{A} + \mathbf{v}_{j+1} C_1 = 0. \tag{3}$$

The examples of these matrices are shown in Fig. 2.

In the literature so far, attention is mainly focused on the steady-state analysis of infinite QBD-M processes [1,3,9] performed by a direct approach with spectral expansion method proposed in [3] or an indirect approach with reblocking technique applied in [12].

To cope with the numerical problem, we apply the theory of generalized invariant subspace [2]. Recall that with the theory of generalized invariant subspace, we are able to compute matrices $U = [\,U_1\ U_2\,]$ and $V = [\,V_1\ V_2\,]$, which makes the following decomposition possible

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix} \quad \text{and} \quad U^{-1}GV = \begin{bmatrix} G_{11} & 0 \\ 0 & G_{22} \end{bmatrix} \tag{4}$$

where matrices $E$ and $G$ are constructed from the QBD-M process as follows

$$G = \begin{bmatrix} 0 & 0 & \dots & \dots & -D_0 \\ I & 0 & \dots & \dots & -D_1 \\ 0 & I & \dots & \dots & -D_2 \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \dots & I & -D_{y-1} \end{bmatrix}, \quad E = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & D_y \end{bmatrix} \tag{5}$$

matrix $D_i$ $(i = 0, 1, \dots, L+1)$ is the coefficient matrix of $\mathbf{v}_{j-L+i}$ in the balance equations.

Let $m_s$ and $m_u$ denote the number of singularities of matrix pencil $\lambda E - G$, which lie inside and outside the open unit disk, respectively. Note that the singularities of the matrix pencil $\lambda E - G$, i.e. the roots of $det(\lambda E - G) = 0$ are exactly the roots of the equation $det(D(\lambda)) = 0$, where $D(\lambda) = D_0 + D_1\lambda + \dots + D_{L+1}\lambda^{L+1}$. As a consequence, we have $m_s + m_u = m = (L+1) \times (\mathcal{N}+1)$. Matrices $U_1, V_1$ are of size $m \times m_u$, matrices $U_2, V_2$ are of size $m \times m_s$, matrices $E_{11}, G_{11}$ are of size $m_u \times m_u$ and matrices $E_{22}, G_{22}$ are of size $m_s \times m_s$, respectively.

Let define the following partition and matrices

$$U^{-1} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \quad F_1 = E_{11}G_{11}^{-1}, \quad F_2 = G_{22}E_{22}^{-1},$$

$$T_n = (n\text{-th pos.}) \begin{bmatrix} 0 \\ \vdots \\ I \\ \vdots \\ 0 \end{bmatrix} \quad \text{for } 0 \le n \le L \tag{6}$$

matrices $F_1$ and $F_2$ are of size $m_u \times m_u$ and $m_s \times m_s$, respectively. Let define the grouping

$$\mathbf{w}_k = [\,\mathbf{v}_{M_1-L+k} \ \dots \ \mathbf{v}_{M_1+k}\,] \tag{7}$$

for $0 \le k \le K = M_2 - M_1 + 1$ and

$$[\,\mathbf{p}_k\ \mathbf{q}_k\,] = \mathbf{w}_k[\,U_1\ U_2\,] \tag{8}$$

The vectors $\mathbf{p}_k$ and $\mathbf{q}_k$ $(0 \le k \le K)$ are of size $m_u$ and $m_s$, respectively. It can be easily checked that

$$\mathbf{w}_{k+1}E = \mathbf{w}_k G \qquad 0 \le k \le K \tag{9}$$

By post-multiplying the Eq. (9) by matrix $V$ and combining with Eq. (4), we obtain two un-couple generalized difference equations for $\mathbf{p}_k$ and $\mathbf{q}_k$

$$\mathbf{p}_{k+1}E_{11} = \mathbf{p}_k G_{11} \qquad 0 \le k < K \tag{10}$$

$$\mathbf{q}_{k+1}E_{22} = \mathbf{q}_k G_{22} \qquad 0 \le k < K \tag{11}$$

This follows that the next two expression hold

$$\mathbf{p}_k = \mathbf{p}_K F_1^{K-k} \quad \text{and} \quad \mathbf{q}_k = \mathbf{q}_0 F_2^k \tag{12}$$

We are now able to express the level probability vectors as

$$\begin{aligned} \mathbf{v}_{M_1-L+k} &= \mathbf{w}_k T_0 \\ &= \mathbf{p}_K F_1^{K-k} X_1 T_0 + \mathbf{q}_0 F_2^k X_2 T_0 \quad \text{for} \quad 0 \le k \le K \end{aligned} \tag{13}$$

and

$$\begin{aligned} \mathbf{v}_{M_1-L+K+n} &= \mathbf{w}_K T_n \\ &= \mathbf{p}_K X_1 T_n + \mathbf{q}_0 F_2^K X_2 T_n \quad \text{for} \quad 0 \le n \le L \end{aligned} \tag{14}$$

With the Eqs. (13) and (14), the unknown probability vectors we have to compute now are $\mathbf{v}_0, \dots, \mathbf{v}_{M_1-L-1}$, $\mathbf{p}_K$, $\mathbf{q}_0$ and $\mathbf{v}_{M_2+2}, \dots, \mathbf{v}_{\mathcal{K}}$, i.e. there are $(M_1 - L) + y + (\mathcal{K} - M_2 - 1) = M_1 + \mathcal{K} - M_2$ unknown vectors, each of size $\mathcal{N}+1$. To get these unknown vectors, we have vector Eq. (1) for $j = 0, \dots, M_1 - 1$ (the first $M_1$ equations) and for $j = M_2 + 1, \dots, \mathcal{K}$ (the last $\mathcal{K} - M_2$ equations). However, only $M_1 + \mathcal{K} - M_2 - 1$ of these equations are linearly independent due to the fact that the generator matrix of the Markov process is singular. This means one equation must be replaced by the normalized Eq. (2). Using Eqs. (13) and (14), the equivalent form of (2) is

$$\begin{aligned} &\sum_{j=0}^{M_1-L-1} \mathbf{v}_j\mathbf{e} + \sum_{j=M_2+2}^{\mathcal{K}} \mathbf{v}_j\mathbf{e} \\ &+ \mathbf{p}_K \left[ \sum_{k=0}^{K} {F_1}^{K-k} X_1 T_0 + \sum_{n=1}^{L} X_1 T_n \right] \mathbf{e} \\ &+ \mathbf{q}_0 \left[ \sum_{k=0}^{K} F_2^k X_2 T_0 + \sum_{n=1}^{L} F_2^K X_2 T_n \right] \mathbf{e} = 1 \end{aligned} \tag{15}$$

What remains to be clarified is how to determine the values of $m_u$ and $m_s$. For this aim, we take a corresponding infinite QBD-M process, which possesses the same balance equations (3) with the same level-independent coefficient matrices $\mathcal{A}$, $B_i$ $(1 \le i \le L)$.

Using the interpretation of stability formulated in previous works (see [3,6]), stability condition of the corresponding infinite QBD-M process is checked by evaluating the inequality

$$\mathbf{v}\left(\sum_{s=1}^{L} s B_s\right)\mathbf{e} < \mathbf{v}\,(C_1)\,\mathbf{e} \tag{16}$$

where $\mathbf{v}$ is a vector of size $\mathcal{N}+1$ and is the solution of the equations

$$\mathbf{v}\left(\mathcal{A}+\sum_{s=1}^{L} B_s + C_1\right) = 0; \quad \mathbf{v}.\mathbf{e} = 1 \tag{17}$$

If the inequality (16) is fulfilled, then it is known from [3] that the $det(D(\lambda)) = 0$ equation has exactly $L*(\mathcal{N}+1)$ roots inside the unit disk. In this case, we have $m_u = (\mathcal{N}+1)$ and $m_s = L*(\mathcal{N}+1)$. Also from [3], the case in which the inequality does not hold implies that $m_u = (\mathcal{N}+1)+1$ and $m_s = L*(\mathcal{N}+1)-1$.

### 2.1 Average number of customers

Denote $\boldsymbol{\pi} = \sum_{j=0}^{K_{\mathrm{LP}}} \mathbf{v}_j$ and let $\pi_i$ be the $i$th element of vector $\pi$.

For the high-priority class, the average number of customers in the queue is computed as

$$E_{\mathrm{high}} = \sum_{i=N}^{N+K_{\mathrm{HP}}} (i-N)\pi_i. \tag{18}$$

The average number of low-priority customers in the system is calculated as

$$E_{\mathrm{low}} = \sum_{j=0}^{K_{\mathrm{LP}}} j\mathbf{v}_j\mathbf{e}. \tag{19}$$

### 2.2 Batch loss probability

This parameter is defined as a probability that an arriving batch is rejected due to the lack of storage and/or due to the dropping control level. For the high-priority class, batch of size $k$ is rejected only if the number of high-priority customers in the system is more than $(N+K_{\mathrm{HP}})-k$ upon his arrival. Using PASTA (Poisson Arrivals See Time Average) property, the loss probability for high-priority batch of size $k$ is

$$P(\text{HP batch of size } k \text{ loss}) = \sum_{i=(N+K_{\mathrm{HP}}-k+1)^+}^{N+K_{\mathrm{HP}}} \pi_i, \tag{20}$$

where $(x)^+ = \max(x, 0)$.

The batch loss priority for the high-priority class is

$$P(\text{HP batch loss}) = \sum_{k=1}^{H} h_k \left(\sum_{i=(N+K_{\mathrm{HP}}-k+1)^+}^{N+K_{\mathrm{HP}}} \pi_i\right). \tag{21}$$

For the low-priority class, a batch of size $k$ is always dropped with probability $p_{\mathrm{dropp}}$ if there are more than $K_{\mathrm{LP}}-m_2$ low-priority customers in the system. Moreover, this batch is also dropped if it observes more than $K_{\mathrm{LP}}-k$ customers in the system. Therefore, two kinds of batch loss probability can be introduced. The first one, referred as the blocking probability, is caused merely by a finite buffer and is derived in the same way we have done for the high-priority class.

$$P^{(1)}(\text{LP batch of size } k \text{ loss}) = \sum_{j=(K_{\mathrm{LP}}-k+1)^+}^{K_{\mathrm{LP}}} \mathbf{v}_j.\mathbf{e}, \tag{22}$$

$$P^{(1)}(\text{LP batch loss}) = \sum_{k=1}^{L} l_k \left(\sum_{j=(K_{\mathrm{LP}}-k+1)^+}^{K_{\mathrm{LP}}} \mathbf{v}_j.\mathbf{e}\right). \tag{23}$$

The second kind of loss is caused by the interaction of the control feedback level and a finite buffer. This is referred to as the overall batch loss probability and is computed as

$$P^{(2)}(\text{LP batch of size } k \text{ loss}) = \sum_{j=(K_{\mathrm{LP}}-k+1)^+}^{K_{\mathrm{LP}}} \mathbf{v}_j.\mathbf{e} + \Omega_{(k\le m_2-1)}\, p_{\mathrm{dropp}} \left(\sum_{j=K_{\mathrm{LP}}-m_2+1}^{K_{\mathrm{LP}}-k} \mathbf{v}_j\mathbf{e}\right), \tag{24}$$

$$P^{(2)}(\text{LP batch loss}) = \sum_{k=1}^{L} l_k \cdot \left[\sum_{j=(K_{\mathrm{LP}}-k+1)^+}^{K_{\mathrm{LP}}} \mathbf{v}_j.\mathbf{e} + \Omega_{(k\le m_2-1)} \cdot p_{\mathrm{dropp}} \left(\sum_{j=K_{\mathrm{LP}}-m_2+1}^{K_{\mathrm{LP}}-k} \mathbf{v}_j\mathbf{e}\right)\right], \tag{25}$$

where $\Omega_{\mathcal{X}}$ denotes the indication function, which is 0 if event $\mathcal{X}$ is false and 1 otherwise.

### 2.3 Customer loss probability

Let the average arrival batch size of the high-priority and the low-priority class be $\bar{h}$ and $\bar{l}$, respectively. It is clear that $\bar{h} = \sum_{i=1}^{H} i.h_i$ and $\bar{l} = \sum_{i=1}^{L} i.l_i$. The probability that a customer arrives in a batch with size $k$ is equal to $k.h_k/\bar{h}$ (high-priority case) and $k.l_k/\bar{l}$ (low-priority case). When the WBA rule is applied, the probability that the customer in batch of size $k$ is dropped is equal to the probability that the batch itself is dropped. Therefore, it follows that

$$P(\text{customer loss}) = \sum_{k=1}^{b_{\max}} \frac{k.r_k}{b}.P(\text{batch of size } k \text{ loss}). \tag{26}$$

The expression above is valid for any class of customers. For the high-priority class, we have to make $b_{\max} = H, b = \bar{h}, r_k = h_k$ and use the Eq. (20). For the two kinds of low-priority loss, $b_{\max} = L, b = \bar{l}, r_k = l_k$ substitutions and Eqs. (22) or (24) are used.

### 2.4 Average queueing and system times

Applying Little's formula, the mean waiting time of a high-priority customer is written as

$$W_{\text{high}} = \frac{E_{\text{high}}}{\bar{h}\lambda_H(1 - P(\text{HP customer loss}))}. \tag{27}$$

For the low-priority customer, the mean system time is calculated again by Little's theorem

$$S_{\text{low}} = \frac{E_{\text{low}}}{\bar{l}\left(\sum_{j=0}^{K_{\text{LP}}} \lambda_{L,j}\mathbf{v}_j\mathbf{e}\right)\left(1 - P^{(2)}(\text{LP customer loss})\right)}. \tag{28}$$

### 2.5 System time distribution

For high-priority class, define the Laplace transform

$$S_i(s) = \begin{cases} \left(\frac{N\mu_H}{s+N\mu_H}\right)^{i-N}\left(\frac{\mu_H}{s+\mu_H}\right) & \text{for } N < i \le N + K_H \\ \frac{\mu_H}{s+\mu_H} & \text{for } i \le N \end{cases} \tag{29}$$

If a high-priority customer arrives in $k$th position of batch of size $r$ and sees $i$ other high-priority customers in the system (provided that a batch is accepted), then the Laplace transform of the system time distribution of that customer is given by $S_{i+k}(s)$. Since the probability of being in position $k$ in batch of size $r$ is $\xi_k = \frac{1}{r}$ and the distribution of batch size is known, the total probability rule yields

$$S_{HP}(s) = \sum_{i=0}^{N+K_H} \pi_i \left(\sum_{r=1}^{H} \Omega_{\{r+i\le N+K_H\}} h_r \left(\sum_{k=1}^{r} \xi_k S_{i+k}(s)\right)\right) \tag{30}$$

The derivation of system time distribution for low-priority customer is done in two steps. First, for the sake of convenient interpretation, we renumber the states of the QBD-M process, i.e. each $(i, j)$ state now corresponds to state with assigned index $j * (\mathcal{N} + 1) + i$. With this "transformation", each state of the process can be accessed not by a couple of index numbers, but by only one. Given a state having index $k$, an original couple of indexes are

$$j = \left[\frac{k}{\mathcal{N}+1}\right] \tag{31}$$

$$i = k - j * (\mathcal{N} + 1) \tag{32}$$

where $[x]$ denotes the greatest integer, which is less than or equal to $x$. Let us denote the steady-state probability vector of the process by $\gamma$, which is in fact a rewritten version of all vectors $\mathbf{v}_j$ calculated in the previous section.

Second, the technique of Markov driven workload process is utilized to determine the system time distribution. Recall that the service capacity for low-priority customers are evenly shared between them and it depends on the number of high-priority customers. In other words, one can consider a workload process controlled by our original Markov process. When a driving Markov process is in state $k$, the workload process accomplishes service at rate $r_k = \frac{\max(N-i,0)}{j}$ where $i$ and $j$ are determined from $k$ as in the expression (31) and (32).

When a low-priority customer arrives, it brings its service requirement $x$. This customer will leaves the system when the workload process defined above accomplishes a work amount of $x$. The fact that the customer arrives alone or in batch with other ones does not matter at this point, because the essence here is upon an arrival a workload process changes its service rate due to the state change of a driving Markov process. That is the information of possible batch arrivals is indeed included in the generator matrix of the driving process.

Since the service requirement $x$ is exponentially distributed with parameter $\mu_L$, the Laplace transform of the density function of system time can be derived and appears to correspond to phase time distribution

$$S_{LP}(s) = \mathbf{p}_R[sI - (Q - \mu_L R)]^{-1} R\mu_L\mathbf{e} \tag{33}$$

where $Q$ is the generator matrix of the driving Markov process, $R$ is a diagonal matrix where the entry $(k, k)$ gives the service rate in state $k$, and $\mathbf{p}_R = \frac{\gamma R}{\gamma R\mathbf{e}}$ is a row initial probability vector seen by the entering customer. The phase type distribution has the initial probability $\mathbf{p}_R$, transient matrix $Q - \mu_L R$ and vector $R\mu_L\mathbf{e}$ of rates to the absorbing state.

## 3 Numerical results

In what follows, unless otherwise stated, the parameter set $N = 10, \mu_{\text{LP}} = 1, \lambda_{\text{LP}} = 2, \mu_{\text{HP}} = 2, \lambda_{\text{HP}} = 4, K_{\text{LP}} = 50, H = L = 2$ is applied. The distribution of batch sizes is assumed to be uniform.

The impact of the dropping probability ($p_{\text{dropp}}$) on performance measures related to the low-priority class is shown in Figs. 3 and 4 for $m_1 = 3, K_{\text{HP}} = 10$. In Fig. 3, both the blocking and the overall loss probability are plotted as a function of the dropping probability. The first observation is that increasing the dropping probability of arriving batches increases slightly the overall and at the same time reduces significantly the blocking probability. The later performance measure is in connection with the mean queue length, which decreases with the dropping probability as shown in Fig. 4.

The second observation from the figures is that the smaller the control level $K_{\text{LP}} - m_2$ is chosen, the better performance is obtained from the aspect of the mean queue length and the blocking probability, at the same time, the worse the performance becomes from the aspect of overall loss probability. A

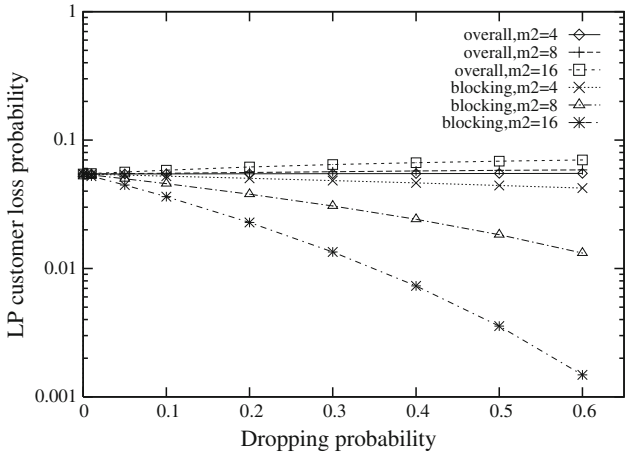


**Fig. 3** LP customer loss probability versus the dropping probability

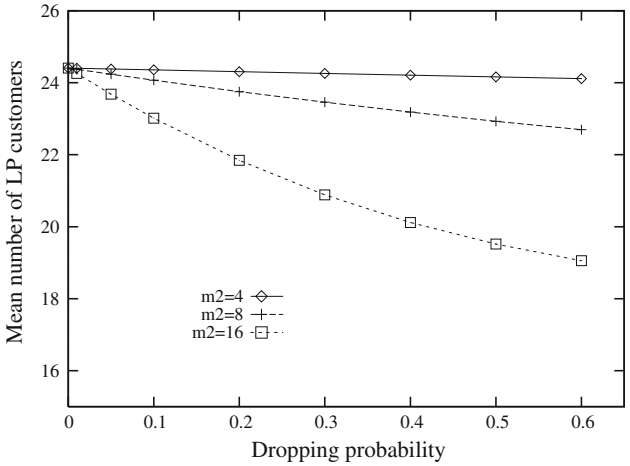


**Fig. 4** Mean queue length of LP customers versus the dropping probability

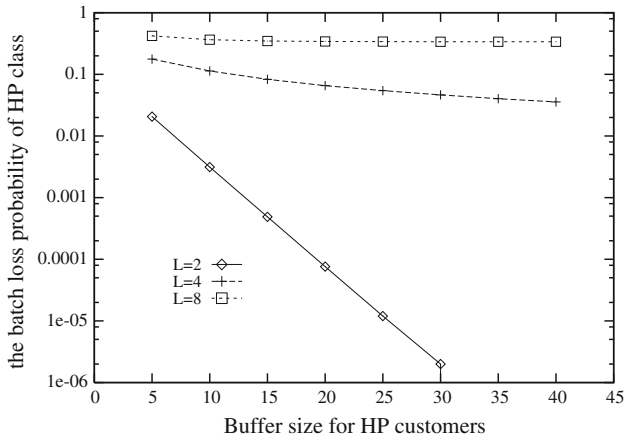


**Fig. 5** HP batch loss probability versus buffer size

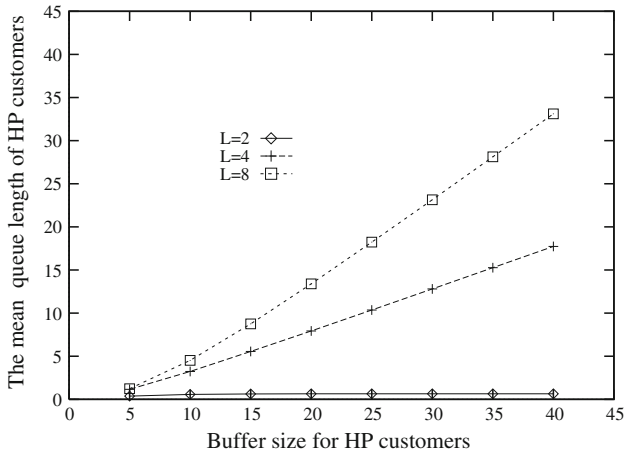


**Fig. 6** Mean queue length of HP customer versus buffer size

trade-off between the increase of overall customer loss probability and the reduction of mean queue length (and through it, the mean system time) should be carefully considered when choosing an appropriate control level $K_{\mathrm{LP}} - m_2$.

Figures 5 and 6 illustrate the effect of buffer size ($K_{\mathrm{HP}}$) for customers of high priority on the queueing performance. The system parameters are $N = 5, m_1 = 5, m_2 = 8, p_{\mathrm{dropp}} = 0.01$. The maximum size of arriving batches is chosen in sequence to be 2, 4 and 8. Obviously, increasing the buffer size reduces the batch loss probability as well as customer loss probability as Fig. 5 shows. The decreasing rate is quite fast when the maximum batch size is small (corresponding to small offered traffic) and seems to slow down when the maximum batch size (so as the offered traffic) becomes greater. Note that if reducing the batch (customer) loss probability is performed by increasing the storage requirement then it also implies the increasing of mean queue length and through it the mean queueing delay, especially in the case of heavy traffic load. This situation is depicted in Fig. 6.

Now we discuss the influence of offered traffic on the queueing performance with respect to interaction between two classes. The system parameters are $m_1 = 5, m_2 = 8$, $K_{\mathrm{HP}} = 10, p_{\mathrm{dropp}} = 0.01, \mu_{\mathrm{LP}} = 1$. The offer traffic is increased by varying $\lambda_{\mathrm{HP}}/\mu_{\mathrm{HP}}$ meanwhile $r_{\mathrm{LP}} = \lambda_{\mathrm{LP}}/\mu_{\mathrm{LP}}$ is fixed. Figures 7 and 8 clearly indicate the impact of the preemptive principle. As high-priority customers have the right to occupy service capacity over low-priority customers, the offered traffic of high priority first forces more customers of low priority to disclaim their service capacity and return to waiting condition. Until a certain value of $\lambda_{\mathrm{HP}}/\mu_{\mathrm{HP}}$, there is no customer of the high-priority class in the queue and no high-priority customer loss at all. Further the increase of the offered traffic of high-priority customers causes the occupancy of buffers and initiates lost events relating to both two classes. However, due to the preemptive service discipline, the mean number of low-priority customers in the queue, as well as the customer loss probability of the low-priority class are increasing at a greater rate than the corresponding ones

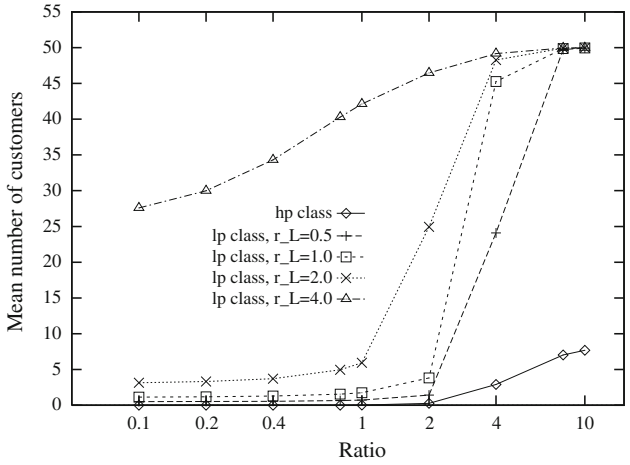


**Fig. 7** Mean queue length versus offered traffic

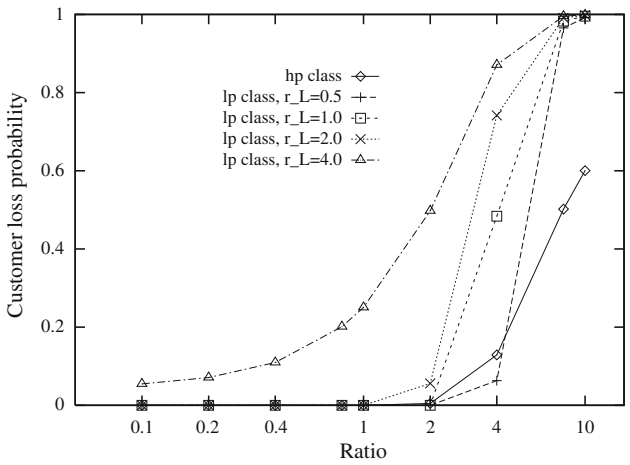


**Fig. 8** Customers loss probability versus offered traffic

for the high-priority class. It is also worth emphasizing that the offered load of the low-priority class (expressed by parameter $r_L$) does not have any effect on the performance of the high-priority class. That is why for different values of $r_L$, the mean number and the loss probability of high-priority customers remains the same as plotted in Figs. 7 and 8.

## 4 Conclusion

In this paper, the steady-state analysis of a queue with two priority classes has been performed. To model some practical problems, additional extensions have been considered, such as finite capacity of buffers, batch arrival condition and level-dependent feedback control.

We have shown that the steady-state probabilities can be computed using the theory of generalized invariant subspace. Using the presented methodology, a performance trade-off can be achieved regarding the application of congestion control for networks.



## References

1. Tran, H.T., Do, T.V.: A new iterative method for systems with batch arrivals and batch departures. In: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, CNDS'00, San Diego, USA (2000)
2. Akar, N., Sohraby, K.: Matrix-geometric Solutions in M/G/1 type Markov Chains with Multiple Boundaries: A Generalized State-space Approach. International Teletraffic Congress (1997)
3. Chakka, R.: Performance and Reliability Modelling of Computing System Using Spectral Expansion. PhD thesis, University of Newcastle upon Tyne (1995)
4. Chakka, R.: Spectral expansion method for QBD and QBD-M processes in performance modeling of computing and communication systems: a review. In: Krishna, P., Babu, M., Ariwa, E. (eds.) Global Trends in Information Systems and Software Applications, Communications in Computer and Information Science, vol. 270, pp. 794–810. Springer, Berlin (2012)
5. Chakka, R., Do, T.V.: Some new Markovian models for traffic and performance evaluation of telecommunication networks. In: Kouvatsos, D. (ed.) Network Performance Engineering, Lecture Notes in Computer Science, vol. 5233, pp. 642–664. Springer, Berlin (2011)
6. Ciardo, G., Smirni, E.: ETAQA: an efficient technique for the analysis of QBD-processes by aggregation. Perform. Eval. **36–37**, 71–93 (1999)
7. Do, T.V., Chakka, R.: Generalized QBD processes, spectral expansion and performance modeling applications. In: Kouvatsos, D. (ed.) Network Performance Engineering, Lecture Notes in Computer Science, vol. 5233, pp. 612–641. Springer, Berlin (2011)
8. Do, T.V., Chakka, R.: On the properties of generalised Markovian queues with heterogeneous servers. In: Nguyen, N.T., Do, T., Thi, H.A. (eds.) Advanced Computational Methods for Knowledge Engineering, Studies in Computational Intelligence, vol. 479, pp. 143–155. Springer, Berlin (2013)
9. Do, T.V., Chakka, R., Sztrik, J.: Spectral expansion solution methodology for QBD-M processes and applications in future internet engineering. In: Nguyen, N.T., Do, T., Thi, H.A. (eds.) Advanced Computational Methods for Knowledge Engineering, Studies in Computational Intelligence, vol. 479, pp. 131–142. Springer, Berlin (2013)
10. Falin, G., Khalil, Z., Stanford, D.A.: Performance analysis of a hybrid switching system where voice messages can be queued. Queue. Syst. **16**, 51–65 (1994)
11. Gail, H.R., Hantler, S.L., Taylor, B.A.: On a preemptive Markovian queue with multiple servers and two priority classes. Math. Oper. Res. **17**, 365–391 (1992)
12. Haverkort, B., Ost, A.: Steady state analysis of infinite stochastic petri nets: a comparing between the spectral expansion and the matrix geometric method. In: Proceedings of the 7th International Workshop on Petri Nets and Performance Models, pp. 335–346 (1997)
13. Stewart, W.J.: Probability, Markov Chains, Queues and Simulation: The Mathematical Basis of Performance Modeling. Princeton University Press, Cambridge (2009)
14. Xiao, X., Ni, L.M.: Internet QoS: the big picture. IEEE Netw. **13**, 1–13 (1999)