The Impact of Buffer Finiteness on the Loss Rate in a Priority Queueing System

J. Van Velthoven, B. Van Houdt* and C. Blondia

University of Antwerp
Dept. Mathematics and Computer Science
PATS Research Group

Abstract. This paper discusses five different ways to approximate the loss rate in a fundamental two class priority system, where each class has its own finite capacity buffer, as well as an exact approach. We identify the type of error one can expect by assuming that one, or both buffers are of infinite size. Furthermore, we investigate whether asymptotic based results can achieve the same level of accuracy as those based on the actual steady state probabilities. Three novel priority queueing models are introduced and efficient algorithms, relying on matrix analytic methods, are developed within this context. A comparative study based on numerical examples is also included.

Keywords: Buffer finiteness, priority queues, loss rate, matrix analytic methods, generating functions.

1 Introduction

The study of priority queues has a long history and is often motivated by their common occurrence in communication networks [16, 17, 3, 4, 8], where they can be used to model Random Access Memory (RAM) buffers and in service part logistics [14, 15]. One of the key performance measures of such a buffer is the loss rate induced by their finite capacity as this strongly affects the network performance. From an analytical point of view, dealing with finite capacity queues is often more troublesome compared to infinite size buffers. Therefore, it is a common practice to analyze the infinite capacity system first and afterward to apply a heuristic method to obtain an estimate of the loss probability for the finite capacity problem (e.g., the probability of having more than C customers in the infinite case is frequently used as an approximation to the loss rate in a finite capacity C setting [6]).

Although this approach has been shown to be fruitful for many queueing systems, more recent results may question such an approach when applied to the (low priority) loss rate in priority queueing system. More specifically, in [2, 10, 16, 17] it is shown that the tail behavior of the low priority buffer occupation might

* B. Van Houdt is a post-doctoral fellow of the FWO-Flanders.

be nongeometric when both the low and high priority buffer is of infinite capacity. Earlier results (e.g., [5]), however, have shown that one typically has geometric tails when the high priority buffer capacity is finite (and arbitrarily large). One does not expect a substantial difference between having an infinite or a very large finite buffer for the high priority traffic (i.e., any simulation run attains some finite maximum queue length). As such, the correspondence between the infinite and finite capacity C system should grow as C increases. However, the tail behavior of both systems, for any finite C, follows a very different regime, implying that blindly trusting upon asymptotic results may lead to substantial errors. The opposite modeling approach, where infinite size queueing systems are studied by truncation to accomplish a numerical evaluation, also exists [3, 4], further motivating our interest in this subject.

The objective of this paper lies in identifying the approaches that may cause poor estimates. To achieve this goal, we will analyze a fundamental discrete-time queueing system with two priority classes, where each priority class has its own waiting room. To study the impact of the buffer finiteness, we introduce three novel discrete time queueing models with batch arrivals: one to analyze the system where both queues (low and high priority) are finite and two models that evaluate the systems where either one of the buffers is finite. The arrival process considered allows correlation between the number of arrivals of each priority class. There is, however, no correlation between the number of arrivals during consecutive time slots. We further assume a deterministic service time of one time slot for all packets. Although this model is a rather restrictive one, it allows us to isolate the impact of assuming one (or two) infinite size buffers on the accuracy of the loss rate obtained.

A variety of matrix analytic techniques are exploited to assess the (estimated) loss rate for each of the three models with at least one finite capacity buffer. Especially useful is the observation that the system with two finite capacity buffers can be captured by the paradigm developed in [7] for an M/G/1-type Markov chain with some regenerative structure, as well as the explicit knowledge of the G matrix appearing in the M/G/1-type Markov chain for the finite capacity high priority buffer. For the setup where both queues are of infinite size we can rely on existing results involving generating functions [17] to obtain numerical results. In case the low priority traffic has an infinite size capacity buffer, we develop two estimates for the loss rate: one based on a numerical evaluation of the steady state probabilities and another that uses an asymptotic description of the tail behavior. This leads to a total of six different approaches to gather the loss rate of a system with two finite buffers (including five approximations).

Notice, although the methods developed in [3] are closely related to the model with an infinite high and finite low capacity buffer, they do not apply directly as batch arrivals are not considered in [3]. Finally, some of the solution techniques can be adapted such that they still apply to a more general setting (i.e., more general service times).

2 System Characteristics

We consider a discrete-time single-server multi-class queueing system with a priority scheduling discipline. We consider a system with two priority classes, denoted as the high (class-1) and the low (class-2) priority class. The arrival process is chosen as in [10, 16, 17] and is characterized by the probabilities

$$a(i_1, i_2) \triangleq Prob[a_1 = i_1, a_2 = i_2], \tag{1}$$

where a_j denotes the number of arriving packets of class-j during a time slot. The corresponding joint probability generating function is given by

$$A(z_1, z_2) \triangleq E\left[z_1^{a_1} z_2^{a_2}\right] = \sum_{i_1=0}^{\infty} \sum_{i_2=0}^{\infty} a(i_1, i_2) z_1^{i_1} z_2^{i_2}. \tag{2}$$

Notice that the number of arrivals from different classes in one slot can be correlated. There is however no correlation between the number of arrivals during consecutive time slots. For further use, let $a_1(i) = \sum_{i_2=0}^{\infty} a(i, i_2)$, $a_2(i) = \sum_{i_1=0}^{\infty} a(i_1, i)$, $a_1^*(i) = \sum_{k=i}^{\infty} a_1(k)$ and $a_2^*(i) = \sum_{k=i}^{\infty} a_2(k)$. The class-$i$ arrival rate λ_i equals $\sum_{k=1}^{\infty} a_i^*(k)$.

We assume a deterministic service time of one time slot for all the packets. Although this assumption is rather strong, it allows us to isolate the impact of assuming one (or two) infinite size buffers on the accuracy of the loss rate obtained. There are two buffers, one for the high and one for the low priority traffic. If an arriving packet finds the server busy, it joins the appropriate buffer. The class-1 packets have priority over these of class-2 and within each class the service discipline is assumed to be First Come First Served. Therefore, when a packet completes its service, the class-1 packet with the longest waiting time will be served. If there are no high priority packets available, the oldest low priority packet is selected for service.

In the next sections, we discuss four different cases, where the buffer size of the two buffers is either finite or infinite. For each situation, we determine the steady state probabilities of the system contents distribution, which can, among others be used to calculate loss probability of the class-2 packets. In each of these models, all events such as arrivals, service completions and packet losses are assumed to occur at instants immediately after the discrete time epochs. We further assume that departures occur before arrivals.

3 Finite High Priority Buffer

Let us first discuss the above-mentioned queueing system provided that the class-1 buffer is finite, with a capacity H, and the class-2 buffer is infinite. We can model this system using an M/G/1-type Markov chain represented by the

following transition matrix:

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \dots \\ A_0 & A_1 & A_2 & A_3 & \dots \\ 0 & A_0 & A_1 & A_2 & \dots \\ 0 & 0 & A_0 & A_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{3}$$

We denote the states of this Markov chain as $\langle i, j \rangle$, where the level $i \geq 0$ denotes the number of low priority packets in the queueing system and $j = 0, \dots, H+1$ reflects the number of high priority packets. An expression for the $(H+2) \times (H+2)$ matrices A_i $(i = 0, 1, \dots)$ is given first. A transition to a lower level can only occur, if there are no high priority packets present in the system, otherwise such a packet is served, preventing any low priority packet from leaving the system. As a consequence only the first row of the matrix A_0 contains non-zero probabilities. A second condition in order to have a transition to a lower level is that no low priority packets arrive during the current time slot. Hence,

$$A_0 = e_1(a(0,0), a(1,0), a(2,0), \dots, a^*(H+1,0)), \tag{4}$$

where $a^*(i,j) = \sum_{k=i}^{\infty} a(k,j)$ and e_1 is a column vector with all its entries equal to zero, except for the first which equals one. The transitions from state $\langle i, j \rangle$ to state $\langle i+k, j' \rangle$ are covered by the matrix A_{k+1}, for $i \geq 1$ and $k \geq 0$. We distinguish two cases: $j = 0$ and $j > 0$. In the first case, a low priority packet is in service; hence, $k+1$ low priority packets need to arrive in order to get a transition to level $i+k$. In the latter case, a class-1 packet occupies the server. A transition to level $i+k$ thus occurs if k class-2 packets arrive. This yields,

$$A_{k+1} = \begin{bmatrix} a(0,k+1) & a(1,k+1) & a(2,k+1) & \dots & a^*(H+1,k+1) \\ a(0,k) & a(1,k) & a(2,k) & \dots & a^*(H+1,k) \\ 0 & a(0,k) & a(1,k) & \dots & a^*(H,k) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a(0,k) & a^*(1,k) \end{bmatrix}. \tag{5}$$

Finally, the matrix B_k contains the probabilities of having a transition from level zero to level k. Level zero corresponds to having zero class-2 packets in the system, implying that k low priority packets must arrive to enter a level k state, for $k \geq 0$,

$$B_k = \begin{bmatrix} a(0,k) & a(1,k) & a(2,k) & \dots & a^*(H+1,k) \\ a(0,k) & a(1,k) & a(2,k) & \dots & a^*(H+1,k) \\ 0 & a(0,k) & a(1,k) & \dots & a^*(H,k) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a(0,k) & a^*(1,k) \end{bmatrix}. \tag{6}$$

To calculate the steady state vector $x = (x_0, x_1, x_2, \dots)$, with x_k a $1 \times (H+2)$ vector for $k \geq 0$, of P, i.e., the joint system contents distribution, Ramaswami's

formula [13, 12, 11] can be used. This formula requires x_0 and a (stochastic) matrix G, being the smallest nonnegative solution of $G = \sum_{k=0}^{\infty} A_k G^k$, as its input. The (j,k)-th entry of this matrix represents the probability that, starting from state $\langle i+1, j\rangle$, the Markov chain visits the set of states $\{\langle i,0\rangle, \ldots, \langle i, H+1\rangle\}$ the first time by entering the state $\langle i,k\rangle$. Finding G is often by far the bottleneck when computing the invariant vector of an M/G/1-type MC. However, in this setup, a transition to a lower level can only occur when there are no high priority packets in the system and there is no arrival of low priority traffic at the current time instant. As a consequence, all the rows of G are identical and can be given explicitly by the vector $\alpha = (a(0,0), a(1,0), a(2,0), \ldots, a^*(H+1,0))/a^*(0,0)$. Notice that $G^k = G = e\alpha$ for $k > 0$ and $g = \alpha$, where g is the unique solution of $gG = g$, with $ge = 1$ and e a column vector of ones. Combining [12, Chapter 3] and the structure of the A_k and B_k matrices with these properties, the following algorithm to compute x can be devised:

Algorithm 3.1: [H/∞]

1. Input: the probabilities $a(i_1, i_2)$ for $0 \le i_1$ and $0 \le i_2$, concerning the arrival process and the capacity H of the buffer for the high priority traffic.
2. Determine the matrices A_k and B_k ($k \ge 0$) using Eqn. (4), (5) and (6).
3. Calculate $\rho = \pi\beta$, where π is the vector representing the stationary distribution of the stochastic matrix $A = \sum_{k=0}^{\infty} A_k$ and $\beta = (1+\lambda_2)e - e_1$.
4. Next, set $\tilde{\kappa}_1 = \psi_2 + (B_1 + a_2^*(2)e\alpha)(I - A_1 - a_2^*(1)e\alpha + a_2(1)e_1\alpha)^{-1}\psi_1$, where I is the identity matrix of the appropriate dimension. The vectors ψ_1 and ψ_2 are given by the following expressions:
$$\begin{aligned}\psi_1 &= (I - A_0 - A_1)(I - e\alpha)(I - A + (e-\beta)\alpha)^{-1} e \\ &\quad + (1-\rho)^{-1} a_2(0) e_1, \\ \psi_2 &= (B - B_0 - B_1)(I - e\alpha)(I - A + (e-\beta)\alpha)^{-1} e \\ &\quad + (1-\rho)^{-1}(\lambda_2 - \rho + a_2(0))e,\end{aligned}$$
where $B = \sum_{k=0}^{\infty} B_k$.
5. The vector x_0 containing the steady state probabilities that there are no low priority packets in the system, is given by $x_0 = (\kappa\tilde{\kappa}_1)^{-1}\kappa$ with κ the invariant probability vector of $K = B_0 + (I - B_0)e\alpha$.
6. Finally, the following recursion is used to calculate the remaining vectors x_i of the steady state distribution:
$$x_i = \left(x_0 \bar{B}_i + \sum_{j=1}^{i-1} x_j \bar{A}_{i+1-j} \right) \left(I - \bar{A}_1\right)^{-1}, \qquad i > 0. \tag{7}$$
In this expression we have $\bar{A}_k = A_k + (a_2^*(k)e - a_2(k)e_1)\alpha$ and $\bar{B}_k = B_k + a_2^*(k+1)e\alpha$, for $k \ge 0$.

Notice, the matrices A_k, B_k, $\bar{A}_k$, $\bar{B}_k$, etc. are fully characterized by their first (or first two) rows; hence, there is no need to store more than one (two) rows for each of these matrices.

In this section, we assumed an infinite size low priority buffer. In practice, buffers are finite and some low priority losses can occur. To estimate the loss probability of the class-2 packets, given the maximum capacity L of the corresponding buffer, we can use the following standard approach in queueing[1]. This approach approximates the packet loss in a finite size L buffer, by the expected value of max(0,number of packets waiting $-L$) in an infinite size system:

$$P_{loss} \approx \sum_{k=L+1}^{\infty} (k-L)x_k e - x_{L+1}(0), \tag{8}$$

where $x_{L+1} = (x_{L+1}(0), x_{L+1}(1), \ldots, x_{L+1}(H))$. The accuracy of this estimate is studied in Section 7. Apart from computing the steady state vector $x = (x_0, x_1, x_2, \ldots)$ in an exact manner via Algorithm 3.1, we can also rely on a theorem by Falkenberg [5, Theorem 3.5], that describes the tail behavior of an M/G/1-type MC, to approximate x_k for k large. This theorem states that the tail will typically decay geometrically, with parameter τ. This parameter is the solution $\tau > 1$ to $\xi\left(\sum_{k=0}^{\infty} A_k z^k\right) = z$, with $\xi(X)$ representing the Perron-Frobenius eigenvalue of the matrix X, and can be computed by a simple bisection algorithm. By plugging the approximated x_k values in (8), we find an alternative estimate for the class-2 loss probability. We will refer to this approach as the H/∞_t approach (as opposed to the H/∞ approach of Algorithm 3.1).

4 Finite Low Priority Buffer

Consider the same system as in Section 3, but with an infinite buffer for the high priority traffic and a finite one of size L for the low priority traffic. As before, we start by setting up an M/G/1-type Markov chain to describe the system. The transition matrix of this Markov chain is given by

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \ldots \\ C_0 & A_1 & A_2 & A_3 & \ldots \\ 0 & A_0 & A_1 & A_2 & \ldots \\ 0 & 0 & A_0 & A_1 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{9}$$

with A_k ($k \geq 0$) an $(L+1) \times (L+1)$ matrix, B_k ($k > 0$) an $(L+2) \times (L+1)$ matrix, B_0 an $(L+2) \times (L+2)$ matrix and C_0 an $(L+1) \times (L+2)$ matrix. The different dimensions originate from the fact that there can be $L+1$ low priority packets in the system only if there are no packets of high priority present. Within a level, the states of this Markov chain correspond to the number of low priority packets; thus, level zero contains one additional state. Arguments similar to the

[1] High priority buffers are usually dimensioned such that hardly any losses occur, therefore, we focus on the low priority packets.

one presented in Section 3 yield the following expressions:

$$A_k = \begin{bmatrix} a(k,0) & a(k,1) & \dots & \bar{a}(k,L) \\ 0 & a(k,0) & \dots & \bar{a}(k,L-1) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \bar{a}(k,0) \end{bmatrix}, \quad k \geq 0, \tag{10}$$

$$B_0 = \begin{bmatrix} a(0,0) & a(0,1) & a(0,2) & \dots & \bar{a}(0,L+1) \\ a(0,0) & a(0,1) & a(0,2) & \dots & \bar{a}(0,L+1) \\ 0 & a(0,0) & a(0,1) & \dots & \bar{a}(0,L) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a(0,0) & \bar{a}(0,1) \end{bmatrix}, \tag{11}$$

$$B_k = \begin{bmatrix} a(k,0) & a(k,1) & \dots & \bar{a}(k,L) \\ a(k,0) & a(k,1) & \dots & \bar{a}(k,L) \\ 0 & a(k,0) & \dots & \bar{a}(k,L-1) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \bar{a}(k,0) \end{bmatrix}, \quad k > 0 \tag{12}$$

and

$$C_0 = \begin{bmatrix} a(0,0) & a(0,1) & a(0,2) & \dots & \bar{a}(0,L+1) \\ 0 & a(0,0) & a(0,1) & \dots & \bar{a}(0,L) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a(0,0) & \bar{a}(0,1) \end{bmatrix}, \tag{13}$$

where $\bar{a}(i,j) = \sum_{k=j}^{\infty} a(i,k)$. Given these expressions, we only need to find x_0 and the matrix G before we can apply Ramaswami's formula to compute $x = (x_0, x_1, \ldots)$. For this setup, there is no explicit expression for G. However, various iterative algorithms can be used to compute G. A low memory implementation can be achieved using the following basic scheme: $G_0 = I, G_n = \sum_{k=0}^{\infty} A_k G_{n-1}^k$. The time needed to execute one iteration can be reduced by observing that only the first row has to be calculated for the entire matrix to be known. That is, the matrix G_n is a triangular matrix with the following structure (due to the probabilistic interpretation of G) :

$$G_n = \begin{bmatrix} G(0) & G(1) & \dots & G(L) \\ 0 & G(0) & \ddots & G^*(L-1) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & G^*(0) \end{bmatrix},$$

where $G^*(i) = \sum_{k=i}^{L} G(k)$. Hence, the steady state vector of the stochastic matrix G is given by $g = (0, 0, \ldots, 1)$. Similarly, as $A = \sum_k A_k$ is also triangular, its invariant vector $\pi = (0, 0, \ldots, 1)$ as well. Furthermore, the matrices A_k, B_k and $C_0 (k \geq 0)$ can be represented by their first row and both $A_k e$ and $B_k e$ equal $a_1(k)e$ (for $k \geq 0$). This leads to the following simplifications: $\beta = \lambda_1 e$,

$\rho = \lambda_1$, $\psi_1 = \psi_2 = a_1(0)(1-\lambda_1)^{-1}e$ and $\tilde{\kappa}_1 = (1-\lambda_1)^{-1}e$. These expression can be obtained from [12, Chapter 3] by noticing that $(I - A + (e-\beta)g)^{-1}e = \sum_{k=0}^{\infty}(A-(e-\beta)g)^k e = \sum_{k=0}^{\infty}\lambda_1^k e = (1-\lambda_1)^{-1}e$. Therefore, the following algorithm can be used to compute $x = (x_0, x_1, x_2, \ldots)$:

Algorithm 4.1: [∞/L]

1. Input: the probabilities $a(i_1, i_2)$ for $0 \leq i_1$ and $0 \leq i_2$, concerning the arrival process and the capacity L of the buffer for the class-2 traffic.
2. Determine the matrices A_k, B_k $(k \geq 0)$ and C_0 using the Eqn. (10), (11), (12) and (13).
3. Set $x_0 = (\kappa\tilde{\kappa}_1)^{-1}\kappa = (1-\lambda_1)\kappa$ with κ the invariant probability vector of the matrix K:
$$K = B_0 + \left(\sum_{k=1}^{\infty} B_k G^{k-1}\right)\left(I - \sum_{k=1}^{\infty} A_k G^{k-1}\right)^{-1} C_0.$$
4. Finally, we can use the following iteration to calculate the other vectors of the steady state distribution:
$$x_i = \left(x_0\bar{B}_i + \sum_{j=1}^{i-1} x_j \bar{A}_{i+1-j}\right)\left(I - \bar{A}_1\right)^{-1}, \qquad i > 0, \tag{14}$$
where $\bar{A}_k = \sum_{i=k}^{\infty} A_i G^{i-k}$ and $\bar{B}_k = \sum_{i=k}^{\infty} B_i G^{i-k}$, for $k \geq 0$.

As A_k, B_k and G are fully characterized by their first row, so are the $\bar{A}_k$ and $\bar{B}_k$ matrices, allowing a significant reduction in the computing time and storage space needed to implement Ramaswami's formula (i.e., (14)). Having found the steady state probabilities, $x_j(k)$ denotes the steady state probability of having j high and k low priority packets in the system. Define $\bar{a}^*(i,j) = \sum_{k=i}^{\infty}\sum_{l=j}^{\infty} a(k,l)$.

Let us now take a look at the calculation of the loss rate of class-2 packets. Low priority packets are lost when the buffer has reached its maximum capacity upon their arrival. This happens in the following two cases:

– The system contains $j = 0, 1$ class-1 packets, i class-2 packets (for $0 \leq i \leq L+1-j$) and (a) at least one high and $L+1-[i-\bar{j}]^+$ low priority packets arrive (where $[x]^+ = \max(0,x)$ and $\bar{j} = j+1 \mod 2$) or (b) no high and at least $L+2-[i-\bar{j}]^+$ low priority packets arrive. Notice, $[i-\bar{j}]^+$ represents the number of class-2 packets left behind by the possible departure and seen by the new arrivals. The expected number of losses due to these cases corresponds to
$$\sum_{j=0}^{1}\sum_{i=0}^{L+1-j} x_j(i)\left(\sum_{k=L+1-[i-\bar{j}]^+}^{\infty}\bar{a}^*(1,k) + \sum_{k=L+2-[i-\bar{j}]^+}^{\infty}\bar{a}(0,k)\right).$$

– There are j ($j > 1$) class-1, i ($0 \leq i \leq L$) class-2 packets and more than $L - i$ low priority packets arrive. The expected number of losses caused by these cases equals $\sum_{j=2}^{\infty} \sum_{i=0}^{L} x_j(i) \left(\sum_{k=L+1-i}^{\infty} \bar{a}_2^*(k) \right)$.

The loss rate of the class-2 traffic can now be calculated by taking the sum of these two expressions. We expect that this approach provides us with a more accurate estimation than the one presented in the previous section, keeping in mind that the high priority queue is typically dimensioned sufficiently large such that hardly any losses occur. In Section 7 we will give some numerical examples in which both approaches are compared.

5 Two Finite Buffers

This section focuses on the system with both a finite, size L low and finite, size H high priority traffic buffer. In practice, all buffers are finite, thus the results obtained in this section are the most relevant. The system state, captured by the number of low and high priority customers in the queue, can be described by a Markov chain with the following transition matrix P:

$$P = \begin{bmatrix} B_0 & B_1 & \dots & B_{L-1} & D_L & C_L \\ A_0 & A_1 & \dots & A_{L-1} & D_{L-1} & C_{L-1} \\ 0 & A_0 & \ddots & A_{L-2} & D_{L-2} & C_{L-2} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & A_0 & D_0 & C_0 \\ 0 & \dots & \dots & 0 & F & E \end{bmatrix}. \tag{15}$$

As in Section 3, the states are labeled as $\langle i, j \rangle$, with i and j reflecting the number of low and high priority customers, respectively. Notice that the states $\langle L+1, j \rangle$ can only be reached if $j = 0$. Otherwise, a high priority customer will occupy the server, leaving only L buffer places available for the low priority traffic. As a consequence C_i ($0 \leq i \leq L$) are column vectors, F is a row vector, and E is a scalar.

In many applications, the dimension of the buffer for the class-1 traffic is significantly smaller than the class-2 buffer. Keeping this in mind, choosing the representation above allows us to work with relatively smaller matrices then would be the case when the order of both variables would be switched. Moreover, this choice also causes P to have a useful regenerative structure. The expressions for the matrices A_k and B_k ($0 \leq k < L$) are identical to those given in Section 3 and as a consequence the matrix $G = e\alpha$, being the smallest nonnegative solution to $G = \sum_{i=0}^{\infty} A_i G^i$, is again known explicitly.

Let us now determine the expressions for the matrices C_k, D_k, E and F. First, the matrix C_k ($0 \leq k \leq L$) contains the probabilities of having a transition to level $L+1$, which can only occur when there are no high priority packets in the

system during the next time slot. Meaning, C_k is a column vector, the first two entries of which only differ form zero:

$$C_L = \begin{bmatrix} \bar{a}(0, L+1) \\ \bar{a}(0, L+1) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ and } C_k = \begin{bmatrix} \bar{a}(0, k+2) \\ \bar{a}(0, k+1) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ 0 \leq k < L. \tag{16}$$

A similar argument can be used to find

$$E = \bar{a}(0, 1) \tag{17}$$

The transitions to level L are described by D_k $(0 \leq k \leq L)$ and F, and can be written as:

$$D_L = \begin{bmatrix} a(0,L) & \bar{a}(1,L) & \bar{a}(2,L) & \dots & \bar{a}^*(H+1,L) \\ a(0,L) & \bar{a}(1,L) & \bar{a}(2,L) & \dots & \bar{a}^*(H+1,L) \\ 0 & \bar{a}(0,L) & \bar{a}(1,L) & \dots & \bar{a}^*(H,L) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \bar{a}(0,L) & \bar{a}^*(1,L) \end{bmatrix}, \tag{18}$$

$$D_k = \begin{bmatrix} a(0,k+1) & \bar{a}(1,k+1) & \bar{a}(2,k+1) & \dots & \bar{a}^*(H+1,k+1) \\ a(0,k) & \bar{a}(1,k) & \bar{a}(2,k) & \dots & \bar{a}^*(H+1,k) \\ 0 & \bar{a}(0,k) & \bar{a}(1,k) & \dots & \bar{a}^*(H,k) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \bar{a}(0,k) & \bar{a}^*(1,k) \end{bmatrix} \tag{19}$$

and

$$F = (a(0,0), \bar{a}(1,0), \bar{a}(2,0), \dots, \bar{a}^*(H+1,0)), \tag{20}$$

Now that we have derived an expression for the building blocks of the transition matrix P, we are in a position to calculate its steady state distribution $x = (x_0, x_1, \dots, x_{L+1})$. P is a downward skip-free finite transition matrix with a special regenerative structure, in [7, Theorem 4.1] Ishizaki introduced an efficient algorithm (similar to Ramaswami's formula) to compute the steady state vector of such a matrix P. Applying this algorithm to our setting and using the same notations as in Section 3, we can calculate the steady state probabilities by means of the following set of equations:

Algorithm 5.1: [H/L]

1. Input: the probabilities $a(i_1, i_2)$ for $0 \leq i_1$ and $0 \leq i_2$, concerning the arrival process and both buffer capacities L and H.
2. Determine the matrices A_k, B_k $(0 \leq k \leq L-1)$, C_k, D_k $(0 \leq k \leq L)$, E and F using Eqns. (4–6) and (16–20).
3. Let x_0 be the stochastic solution of $x_0 = x_0 K$, where $K = B_0 + (I - B_0)e\alpha$.

4. Set $x_i = \left(x_0\bar{B}_i + \sum_{k=1}^{i-1} x_k\bar{A}_{i-k+1}\right)\left(I-\bar{A}_1\right)^{-1}$ for $i = 1,\ldots,L-1$, where the matrices $\bar{A}_k$ and $\bar{B}_k$ were defined in step 6 of Algorithm 3.1.
5. Let $x_L = \left(\sum_{k=0}^{L-1} x_k(D_{L-k} + C_{L-k}F^*)\right)(I-\bar{D}_0)^{-1}$, where $F^* = F/(Fe)$ and $\bar{D}_0 = D_0 + C_0F^*$.
6. Compute $x_{L+1} = \left(\sum_{i=0}^{L} x_iC_{L-i}\right)(1-E)^{-1}$.
7. Normalize $x = (x_0, x_1, \ldots, x_{L+1})$ such that $\sum_{i=0}^{L+1} x_ie = 1$.

Observe that we compute $(x_0,\ldots,x_{L-1})$ in exactly the same way as in Section 3, except that x_0 is not normalized. Normalization occurs after computing x_L and x_{L+1}. Thus, obtaining results for the system with two finite buffers is almost computationally equivalent to solving the finite/infinite system. This is exceptional as finite buffer systems typically demand more computational power. Using these steady state probabilities, the loss probability of the class-2 packets can be calculated in the same way as in Section 4.

6 Two Infinite Buffers

To analyze the system where both buffers are of infinite size, we can rely on some existing results in the literature. From [17], it follows that the probability generating function $Q_2(z)$ of the number of class-2 packets waiting in the queue can be written as

$$Q_2(z) = (1-\lambda)\frac{(z-1)(Y(z)-1)}{(z-Y(z))(A(1,z)-1)}, \tag{21}$$

where $Y(z)$ is implicitly defined by $Y(z) = A(Y(z),z)$. From Rouché's theorem, it can be seen that there is exactly one solution for $Y(z)$, with $|Y(z)| \leq 1$ for $|z| < 1$. There are two approaches to retrieve an estimate for the class-2 loss probability from (21). The first involves a numerical inversion of the generating function to obtain an approximation for the distribution of the number of class-2 packets present in the buffer. The inversion is realized using a discrete Fourier transform method (DFT), where a damping parameter $0 < r < 1$ is used [1]. We make use of a damping parameter such that when evaluating $Q_2(z)$ at $r\omega_N^s$, where ω_N^s for $s = 0,\ldots,N-1$ are the N-th roots of unity, $Y(z)$ is uniquely defined by Rouché's theorem as $|r\omega_N^s| < 1$. This leads to the following algorithm:

Algorithm 6.1: [∞/∞]

1. Input: the probabilities $a(i_1,i_2)$ for $0 \leq i_1$ and $0 \leq i_2$, concerning the arrival process.
2. Evaluate $Q_2(z)$ at $r\omega_N^s$, where ω_N^s for $s = 0,\ldots,N-1$ are the N-th roots of unity (where N is a power of 2 sufficiently large). This entails that we have to determine the unique solution of $Y(z) = A(Y(z),z)$, with $|Y(z)| < 1$, for each $z = r\omega_N^s$.

3. Compute q_k, for $k = 0, \ldots, N-1$, via the inverse DFT. The values q_k can be used as an approximation to the probability of having k buffered class-2 packets.

In [9], it is argued that as long as enough numerical precision is used, the desired probabilities can be obtained to any given accuracy. Therefore, it is advised to use a software package that supports high numerical precision when implementing this algorithm (e.g., Maple or Mathematica). The class-2 loss probability can be estimated as $P_{loss} \approx \sum_{k=L+1}^{\infty}(k-L)q_k$.

A second approach is to rely on the tail behavior of (21) to get an alternate approximation q'_k for the probability of having k class-2 packets buffered. A description of the tail behavior of interest can be found easily from [17, Eqn. (21)]. The key in generating numerical results from these expressions is the computation of the real numbers $z_T > 1$ and $z_B > 1$: these numbers are the solutions to $A(z,z) = z$ and $A^{(1)}(Y(z),z) = 1$ (where $A^{(1)}(z_1,z_2)$ is the first partial derivative of $A(z_1,z_2)$), respectively. As $A(z,z)$ is a convex function with $A(1,1) = 1$ and $\left.\frac{dA(z,z)}{dz}\right|_{z=1} < 1$ (otherwise the system would be unstable), we can apply a simple bisection algorithm to find z_T. For z_B we can use the following algorithm:

Algorithm 6.2: [∞/∞_t]

1. Set $z_{2:min} = 1$ and $z_{2:max} = 2$. Determine $z_1 > 1$ via a bisection algorithm such that $A^{(1)}(z_1, z_{2:max}) = 1$. As long as $A(z_1, z_{2:max})$ is less than z_1, increase $z_{2:min}$ and $z_{2:max}$ by one.
2. Let $z_{2:new} = (z_{2:min} + z_{2:max})/2$. Determine $z_1 > 1$ via a bisection algorithm such that $A^{(1)}(z_1, z_{2:new}) = 1$. If $z_1 < A(z_1, z_{2:new})$, assign $z_{2:new}$ to $z_{2:max}$, else $z_{2:min} = z_{2:new}$. Repeat step 2 until $z_{2:max} - z_{2:min} < 10^{-14}$.

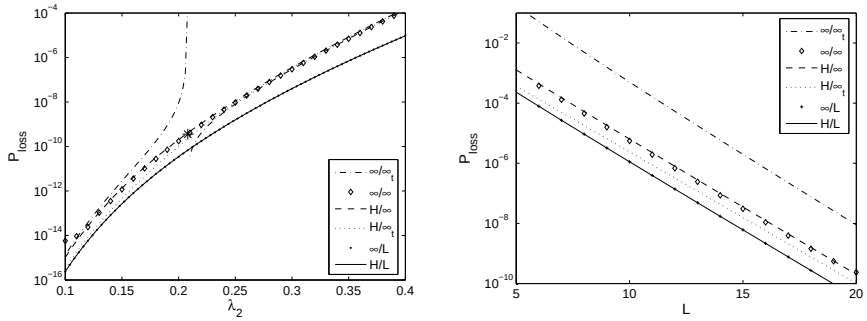
For details on how to compute an approximation q'_k given z_T and z_B we refer to [17].

7 Numerical Examples

In this section we will compare the discussed approaches to estimate the loss probability of the low priority traffic. Let us first describe the arrival process under consideration. The number of arrivals during one time slot is bounded by N and is generated by a Bernoulli process with rate λ_T/N, where an arriving packet belongs to class-j ($j = 1, 2$), with a probability λ_j/λ_T (with $\lambda_1 + \lambda_2 = \lambda_T$). This arrival process is characterized by the joint probability generating function

$$A(z_1, z_2) = \left(1 + \sum_{j=1}^{2} \frac{\lambda_j}{N}(z_j - 1)\right)^N . \tag{22}$$

It was also used in [16] where a non-blocking output-queueing switch with N inlets and N outlets was given as a possible application.



(a) $\lambda_1 = 0.4, H = 25, L = 20$ (b) $\lambda_1 = 0.4, \lambda_2 = 0.2, H = 25$

Fig. 1. *Comparison of the loss rate of low priority packets for each of the six approaches*

More specifically, we assume the maximum number of simultaneously arriving packets to be 16. The probability that a class-1 packet arrives is fixed throughout this section at $\lambda_1 = 0.4$, while the buffer for the high priority traffic has a size $H = 25$ packets. By dimensioning the high priority buffer like this, the probability that a class-1 packet is dropped due to buffer overflow is in the order of 10^{-20}. Figure 1(a) represents, for each of the discussed approaches, the loss rate of the class-2 packets where the corresponding buffer has a size $L = 20$ packets and $\lambda_2 = 0.1, \ldots, 0.4$.

The exact results obtained via the system with two finite buffers, is denoted by the full line. It can be seen that the ∞/L results are very accurate, meaning there is no harm in assuming an infinite size high priority buffer. The other four approximation approaches give rise to higher loss probabilities. This difference is caused by the heuristic calculation used to estimate the loss probability. Whenever an infinite buffer is used for the low priority traffic, the estimate for the loss probability is based on the probability that the number of packets in the buffer exceeds L. In general, this causes an overestimation as packets that would be dropped earlier by the finite capacity system may still reside in the infinite buffer setup when the next arrival(s) occur.

In case both queues are assumed to be infinite, we observe some poor results around $\lambda_2 = 0.21$ for the ∞/∞_t approach, which relies on the asymptotic tail behavior of the class-2 queue. This is caused by the fact that the tail transition point is situated at $p_t = 0.208060765$: for $\lambda_2 \leq p_t$ the tail is nongeometric, whereas for $\lambda_2 > p_t$, we have a geometric tail. When $\lambda_2 < p_t$, the asymptotic regime is dominated by a branch point, whereas for $\lambda_2 > p_t$ there exists a dominant pole. When we approach the transition point, the domination becomes less severe and significant errors occur as shown in this example. The loss rate obtained for $\lambda_2 = p_t$ is quite accurate as indicated by the star on the plot.

Figure 1(b) illustrates the influence of the buffer capacity for the low priority traffic on the loss probability of this traffic in the case where $\lambda_2 = 0.2$. As

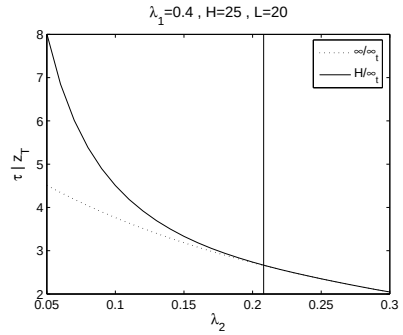


Fig. 2. *Comparison of the coefficient for the tail approaches*

could be expected, the loss probability of the class-2 packets decreases when this buffer becomes larger. If we compare the different approaches, we notice the same behavior as in Figure 1(a). Because taking $\lambda_2 = 0.2$ brings us relatively close to the transition point, a significant error is introduced by assuming both queues infinite and relying on the tail behavior. That is, the loss probability obtained by this approach overestimates the actual loss rate by a factor of 100 to 1000. If, for example, we would use the ∞/∞_t approximation to dimension the class-2 buffer such that the loss probability is less than 10^{-5}, we would need a buffer of 14 packets, whereas a buffer of only 8 packets suffices if we consider the H/L approach.

In Figure 2 we compare the two approaches based on the tail behavior of the low priority queue. In fact, the full line represents the geometric decay parameter in function of the arrival rate λ_2 of the low priority traffic. The dotted line represents the parameter z_T, described in algorithm 6.2. On the figure, the transition point is indicated by the vertical line. As mentioned before, on the left of this line the tail for the class-2 queue obtained by algorithm 6.2 is nongeometric, whereas on the right of the transition point the tails are geometric. It can be noticed that the values indicated by the two curves, converge to the same value as λ_2 reaches the transition point.

8 Conclusions

In this paper we have studied the influence of buffer finiteness on the low priority loss probability in a queueing system with two priority classes. Three novel discrete time queueing models with at least one finite capacity buffer were introduced, together with efficient solution techniques that rely on matrix analytic methods. Six different approaches to estimate the low priority loss rate were discussed and compared.

The most accurate approximation results were generated by the approach in which only the high priority traffic is considered as infinite. Moreover, given that

the size of the high priority buffer is chosen sufficiently large, the distinction with exact results is negligible. When the low priority queue was assumed to be infinite, we observed an overestimated loss rate. Relying on the actual steady state probabilities or the asymptotic tail behavior seemed to make little difference if the high priority queue was finite. However, in case both queues are infinite very inaccurate loss probability were observed when we made use of the asymptotic tail behavior, especially in the area near the transition point.

References

1. J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10:5–88, 1992.
2. J. Abate and W. Whitt. Asymptotics for M/G/1 low-priority waiting-time tail probabilities. *Queueing Systems*, 25:173–233, 1997.
3. A.S. Alfa. Matrix-geometric solution of discrete time MAP/PH/1 priority queue. *Naval Research Logistics*, 45:23–50, 1998.
4. A.S. Alfa, B. Liu, and Q.M. HE. Discrete-time analysis of MAP/PH/1 multiclass general preemptive priority queue. *Naval Research Logistics*, 50:662–682, 2003.
5. E. Falkenberg. On the asymptotic behaviour of the stationary distribution of Markov chains of M/G/1-type. *Stochastic Models*, 10(1):75–97, 1994.
6. A. György and T. Borsos. Estimates on the packet loss ratio via queue tail probabilities. In *IEEE Globecom*, San Antonio, TX, USA, Nov 2001.
7. F. Ishizaki. Numerical method for discrete-time finite-buffer queues with some regenerative structure. *Stochastic Models*, 18(1):25–39, 2002.
8. K.P. Sapna Isotupa and David A. Stanford. An infinite-phase quasi-birth-and-death model for the non-preemptive priority M/PH/1 queue. *Stochastic Models*, 18(3):387–424, 2002.
9. N.K. Kim and M.L. Chaudry. Numerical inversion of generating functions: a computational experience. Manuscript, 2005.
10. K. Laevens and H. Bruneel. Discrete-time multiserver queues with priorities. *Performance Evaluation*, 33(4):249–275, 1998.
11. B. Meini. An improved FFT-based version of Ramaswami's formula. *Stochastic Models*, 13:223–238, 1997.
12. M.F. Neuts. *Structured Stochastic Matrices of M/G/1 type and their applications*. Marcel Dekker, Inc., New York and Basel, 1989.
13. V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Commun. Statist.-Stochastic Models*, 4:183–188, 1988.
14. A. Sleptchenko, A. van Harten, and M.C. van der Heijden. An exact analysis of the multi-class M/M/k priority queue with partial blocking. *Stochastic Models*, 19(4):527–548, 2003.
15. A. Sleptchenko, A. van Harten, and M.C. van der Heijden. An exact solution for the state probabilities of the multi-class, multi-server queue with preemptive priorities. *Queueing Systems*, 50(1):81–107, 2005.
16. J. Walraevens, B. Steyaert, and H. Bruneel. Performance analysis of a single-server ATM queue with a priority scheduling. *Computers & Operations Research*, 30(12):1807–1829, 2003.
17. J. Walraevens, B. Steyaert, and H. Bruneel. A packet switch with a priority scheduling discipline: Performance analysis. *Telecommunication Systems*, 28(1):53–77, 2005.