Research Article

Finite-Buffer M/G/1 Queues with Time and Space Priorities

Kilhwan Kim

Department of Management Engineering, Sangmyung University, Seoul, Republic of Korea

Correspondence should be addressed to Kilhwan Kim; khkim@smu.ac.kr

Received 17 December 2021; Accepted 25 March 2022; Published 20 May 2022

Academic Editor: Xindong Peng

Copyright © 2022 Kilhwan Kim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many communication systems have finite buffers and service delay-sensitive and loss-sensitive types of traffic simultaneously. To meet the diverse QoS requirements of these heterogeneous types of traffic, it is desirable to offer delay-sensitive traffic time priority over loss-sensitive traffic, and loss-sensitive traffic space priority over delay-sensitive traffic. To analyze the performance of such systems, we study a finite-buffer M/G/1 priority queueing model where nonpreemptive time priority is given to delay-sensitive traffic and push-out space priority is given to loss-sensitive traffic. Compared to the previous study on finite-buffer M/M/1 priority queues with time and space priority, where service times are identical and exponentially distributed for both types of traffic, in our model we assume that service times are different and are generally distributed for different types of traffic. As a result, our model is more suitable for the performance analysis of communication systems accommodating multiple types of traffic with different service-time distributions. For the proposed queueing model, we derive the queue-length distributions, loss probabilities, and mean waiting times of both types of traffic, as well as the push-out probability of delay-sensitive traffic. With numerical examples, we also explore how the performance measures are affected by system parameters such as the buffer size, and the arrival rates and mean service times of both types of traffic for different service-time distributions.

1. Introduction

In this paper, we study a finite-buffer M/G/1 queueing model with two priority classes, where one class is given time priority and the other space priority, to meet the diverse QoS (Quality of Service) requirements of both classes.

Priority queues have been extensively studied for the performance analysis of communication systems that serve heterogeneous types of traffic with different QoS requirements. Various studies have been conducted on priority queues. One line of these studies focuses on priority queues with an infinite buffer, where packet loss never occurs, so the delays of different classes of traffic are a main concern. In these priority queues, one class of traffic is given time priority over the other class so that high-priority packets can be served before low-priority packets, even though the low-priority packets arrive at the system earlier than the high-priority packets. A variety of priority disciplines, such as nonpreemptive, preemptive, and mixed types, and related queueing models have been proposed to meet the different delay requirements of high- and low-priority classes of traffic. An analysis of the classical time-priority queueing models can be found in [11], Ch. 3, and [12], Ch. 3, and the analysis of more sophisticated time-priority queueing models can be found in [3, 4] and the references therein. Another line of research is focused on priority queueing models with a finite buffer, where packet loss can occur, so the loss probabilities of different classes of traffic are more important than their delays. In these priority queues, one class of traffic is given space priority over the other class so that high-priority packets can access the buffer even when low-priority packets are not allowed to access the buffer and are lost. Several space-priority mechanisms for buffer management, such as partial buffer-sharing and push-out mechanisms, have been proposed to meet the diverse loss requirements of high- and low-priority classes of traffic [5–8].

However, these two types of priority queues have their own limitations, because they only deal with the delay or loss requirements of traffic solely with priority disciplines. Communication systems generally have finite buffers and serve delay- and loss-sensitive types of traffic...
simultaneously. In this case, it can be more desirable to employ both time and space priorities and give time priority to delay-sensitive traffic and space priority to loss-sensitive traffic. For example, Internet traffic can be grouped into streaming and elastic types of traffic. While streaming traffic served on UDP, such as audio and video streaming packets, is delay sensitive, elastic traffic served on TCP, such as file transfer packets, is loss sensitive [9]. Streaming traffic requires real-time transmission, so packets that exceed a certain tight delay bound will simply be discarded at the destination end. Thus, packets that are too delayed to be discarded at the destination end do not have more value than packets lost during their transmission along communication links. Instead, they may be harmful, as valuable communication links are consumed in transmitting useless packets. Therefore, the user of streaming traffic may have an interest in reducing packet delay when competing for transmission links with other types of traffic at the cost of an increase in packet loss when traffic congestion occurs. On the other hand, elastic traffic on top of TCP generally tolerates variation of packet delay, but packet loss has a significant negative impact on the performance of elastic traffic because this causes the transmission rate to drop considerably due to the operation characteristic of TCP [10]. Therefore, the user of elastic traffic may also be interested in reducing packet loss during periods of traffic congestion at the cost of an increase in packet delay when competing for transmission links with other types of traffic. Offering time priority to streaming traffic and space priority to elastic traffic results in an exchange of packet delay and loss between the two types of traffic that may improve the aggregate utility of the communication system that the users of both types of traffic receive.

Several studies have dealt with priority queues with both time and space priorities [6, 9, 11–17]. However, many of them [6, 11, 13, 14] consider priority queues where both time and space priorities are provided to the same class so that they are not appropriate to address the different diverse requirements of both delay- and loss-sensitive types of traffic. Lee and Choi [12] analyzed priority queues with a hierarchical priority structure where traffic was grouped into high- and low-time-priority classes, and then each time-priority class was divided into high- and low-space-priority subclasses. Although the priority queue in [12] was more sophisticated, it was also not appropriate to address the different diverse requirements of delay- and loss-sensitive types of traffic, because both time and space priorities were provided to the same subclass of traffic (high-time-priority and high-space-priority subclass). There is another approach for dealing with both time and space priorities, where the system provides delay-sensitive traffic time priority and a finite buffer, and loss-sensitive traffic space priority with a virtually infinite buffer [15–17]. However, assuming a virtually infinite buffer for one class is not realistic in communication systems. Thus, the applicability of this approach is also limited.

Recently, [9] analyzed a finite-buffer M/M/1 priority queue where delay-sensitive traffic was given nonpreemptive time priority and loss-sensitive traffic was given push-out space priority. To the best of the author's knowledge, this study is the only work dealing with priority queues with a finite buffer where time and space priorities are given to different classes. However, the queueing model in [9] also has a limitation. In [9], the service times of delay-sensitive and loss-sensitive packets are assumed to be exponentially distributed and have the same distribution for the tractability of the analysis. However, this is a substantial limitation because the maximum packet sizes of delay- and loss-sensitive types of traffic can be set to different values and the packet sizes of the traffic can have an arbitrary distribution.

In this paper, we analyze a finite-buffer M/G/1 priority queue where time and space priorities are given to different classes of traffic. This work can be viewed as a generalization of the queueing model in [9] in the sense that the service times of different classes can be assumed to have different arbitrary distributions. As a result, using this queueing model, we can explore the impact of the service-time distributions on system performance. As will be seen in Section 6, the service-time distribution of loss-sensitive traffic has different effects on the performance measures, depending on whether the buffer is nearly monopolized by loss-sensitive traffic or not.

As in [9], our model also employs a push-out space-priority mechanism. Here, we provide a rationale for employing push-out mechanism in the proposed space-time-priority queueing model. As mentioned earlier, there are several space-priority mechanisms to differentiate packet loss probabilities between classes of traffic. The simplest way to implement space priority is to provide each class of traffic with a separate region in the buffer, and the higher the space priority a class of traffic has, the more the portion of the buffer it is given to experience a lower packet loss probability [18]. However, compared to when the whole buffer is shared by all classes of traffic, this mechanism may bring about a significantly underutilized buffer because an arriving packet cannot enter the buffer even when the buffer has many vacancies if the buffer region for the class of the arrival is already full. One of the remedies for this underutilized buffer problem is the partial buffer-sharing (PBS) mechanism, where the buffer is divided into two regions. One region is shared by all packets, regardless of their classes, and the other region is used only for high-space-priority packets [11, 19]. Thus, while the shared region promotes buffer utilization, the dedicated region differentiates packet loss probabilities between classes. However, one difficulty with this mechanism is that it is difficult to determine an appropriate portion of the shared region, especially when the stochastic characteristics of the classes of traffic fluctuate.

The push-out mechanism is different from the above two mechanisms in respect of the way of differentiating loss probabilities. While the above two mechanisms utilize buffer partitioning to differentiate loss probabilities between classes, the push-out mechanism uses selective loss of arriving packets when the buffer is full. If a low-space-priority packet arrives when the buffer is full, then the arriving packet is always lost. However, if a high-space-priority packet arrives when the buffer is full, then a packet of the lower class in the buffer (if any) is lost instead of the arriving packet...
follows a nondegenerate general distribution \( L \) choose one of class-packets of the same class. The other is delay sensitive. The loss-sensitive and delay-sensitive classes of packets will be called loss sensitive, and the other is delay sensitive. The FCFS service order is assumed between the buffer as their traffic load soars. However, this drawback can be alleviated if high-space-priority packets are such elastic as TCP packets that adaptively reduces its traffic load when congestion occurs [10] (even though this behavior of elastic traffic is not considered in the proposed queueing model in this study).

This paper is organized as follows. In Section 2, we define the priority queueing model that is analyzed in this paper. In Section 3, using an embedded Markov chain, we obtain the joint distribution of loss- and delay-sensitive queue lengths at departure epochs. In Section 5, we derive the joint distribution of queue lengths and remaining service time at an arbitrary time using the results of the embedded Markov chain. In Section 6, we derive the loss probabilities and mean waiting times of both classes. In Section 7, we present some numerical examples to investigate the impact of the system parameters, such as the buffer size and the arrival rates and the mean service times of both classes, on the system performance for different service-time distributions.

2. Model

We consider the following finite-buffer M/G/1 queue: There are two classes of packets that arrive at the system. One is loss sensitive, and the other is delay sensitive. The loss-sensitive and delay-sensitive classes of packets will be called classes \( L \) and \( D \), respectively. Class- \( g \) packets, \( g \in \{ L, D \} \), arrive at the system according to a Poisson process with rate \( \lambda_g \), \( \lambda_L > 0 \). Thus, the total arrival rate \( \lambda \) is given by \( \lambda = \lambda_L + \lambda_D \). The service time \( S_h \) of a class- \( h \), \( h \in \{ L, D \} \), packet follows a nondegenerate general distribution \( S_h(t) \), \( t \geq 0 \), of finite mean; i.e., \( 0 < E[S_h] < \infty \).

Since class- \( D \) packets are delay sensitive, we give class- \( D \) packets nonpreemptive time priority over class- \( L \) packets. That is, when the server is available for the next service, we choose one of class- \( D \) packets (if any) in the buffer, even though there are class- \( L \) packets which arrive earlier than the class- \( D \) packets. The FCFS service order is assumed between packets of the same class.

On the other hand, we give class- \( L \) packets push-out space priority over class- \( D \) packets because class- \( L \) packets are loss-sensitive. We assume that the size of the buffer is \( K \), and a packet of either class in the buffer occupies one buffer slot in the buffer. Thus, there are at most \( K + 1 \) packets in the system (one is in service and at most \( K \) packets in the buffer). If the buffer is not full on arrival of a packet, then the packet joins the buffer, regardless of its class. However, if the buffer is full, whether to join the buffer depends on the class of the packet: A class- \( D \) packet is always lost if the buffer is full on its arrival, while a class- \( L \) packet can push out a class- \( D \) packet in the buffer if the buffer is full but there is a class- \( D \) packet in the buffer on its arrival. If the buffer is full and there is more than one class- \( D \) packet in the buffer on arrival of a class- \( L \) packet, then the last class- \( D \) packet in order of arrival is pushed out. Neither class- \( L \) nor class- \( D \) packet is allowed to push out packets of the same class in the buffer or a packet in service.

3. The Embedded Markov Chain at Departure Epochs

We now consider an embedded Markov chain (EMC) at packet departure epochs. Let \( N^d_g(n), g \in \{ L, D \}, n = 1, 2, \ldots \), denote the number of class- \( g \) packets in the system just after the \( n \)-th service completion. Then, \( \{ (N^d_L(n), N^d_D(n)) | n = 1, 2, \ldots \} \) constitutes an EMC. Observe that \( N^d_L(n) \), \( N^d_D(n) \geq 0 \) and \( N^d_L(n) + N^d_D(n) \leq K \) due to the finite-buffer size \( K \). Define the limiting probabilities as

\[
\pi_{(i,j)} = \lim_{n \to \infty} \Pr \{ N^d_L(n) = i, N^d_D(n) = j \}, \quad i, j \geq 0, i + j \leq K.
\]

We will call all the states that satisfy \( N^d_L(n) = i, (i, 0), (i, 1), \ldots, (i, K - i) \), level \( i \), \( 0 \leq i \leq K \). We also define the corresponding partial limiting probability vectors of level \( i \) and the whole limiting probability vector as

\[
\pi_i = [\pi_{(i,0)}, \pi_{(i,1)}, \ldots, \pi_{(i,K-i)}]_1 \times (K+1)/2, \quad 0 \leq i \leq K.
\]

Let \( p_{(i,j), (l,m)} \) denote the transition probability of the EMC from state \( (i, j) \) to state \( (l, m) \), \( i, j, l, m \geq 0 \), \( i + j, l + m \leq K \). We define the partial transition matrix \( P_{il} \) from level \( i \) to level \( l \) as follows:

\[
P_{il} = \begin{bmatrix}
    p_{(i,0), (l,0)} & p_{(i,0), (l,1)} & \cdots & p_{(i,0), (l,K-l)} \\
p_{(i,1), (l,0)} & p_{(i,1), (l,1)} & \cdots & p_{(i,1), (l,K-l)} \\
    \vdots & \vdots & \ddots & \vdots \\
p_{(i,K-l), (l,0)} & p_{(i,K-l), (l,1)} & \cdots & p_{(i,K-l), (l,K-l)}
\end{bmatrix}_{(K+1) \times (K+1)}
\]

(3)

Notice that, since class- \( L \) packets in the buffer are not pushed out, at most one class- \( L \) packet can depart from the system between two consecutive embedded points. Thus, \( P_{il} = 0 \) for \( 0 \leq i \leq K, \ 0 \leq l < i - 1 \). Therefore, the whole transition matrix \( P \) has the left-skip-free structure for levels:
We will exploit this M/G/1-type structure when deriving the limiting probability vector $\pi$. If the EMC is irreducible and ergodic, then the limiting probability vector $\pi$ satisfies the following relationship (see [25], Sec. 7.6):
\[
\pi = \pi P,
\]
\[
\pi e = 1,
\tag{5}
\]
where $e$ is a $(K + 1)(K + 2)/2 \times 1$ column vector whose elements are all 1.

We now derive the transition probabilities $P_{i,j,m,n}$. To do this, let $A_g(S_h)$, $g, h = L, D$, denote the number of class-$g$ arrivals during the service time of a class-$h$ packet. We further define the following probabilities associated with $A_g(S_h)$:
\[
a^h_{i,j} = \Pr\{A_L(S_h) = i, A_D(S_h) = j\} = \int_0^\infty \frac{(\lambda_i t)^j}{j!} \lambda_d e^{-\lambda_d t} dS_h(t) > 0,
\]
\[
a^h_{i,j} = \Pr\{A_L(S_h) = i\} = \int_0^\infty \frac{(\lambda_i t)^j}{j!} e^{-\lambda_i t} dS_h(t) > 0.
\tag{6}
\]

for $i, m \geq 0$ and $h = L, D$. The inequalities above hold because $\lambda_g > 0$ and $S_h$ is a nondegenerate nonnegative random variable with finite mean.

Since we assume that class-$D$ packets have time priority over class-$L$ packets, if $N_D^L(n) \geq 1$, then a class-$D$ packet gets serviced during the transition from the $n$-th to the $(n + 1)$-th embedded points. Furthermore, if $N_D^L(n) = 0$ and $N_D^L(n) \geq 1$, then a class-$L$ packet gets serviced during the transition. Finally, if $N_D^L(n) = N_D^L(n) = 0$, then an idle period begins and a packet which arrives during the idle period gets serviced during the transition. Hence, we can divide the state space $\{(i, j): i, j \geq 0, i + j \leq K\}$ of the EMC into the following disjoint subsets, each of which corresponds to the type of the transition time between the $n$-th and $(n + 1)$-th embedded points:

1. $i \geq 0, j \geq 1$, and $i + j \leq K$, where the transition time is $S_D$.
2. $i \leq K$ and $j = 0$, where the transition time is $S_L$.
3. $i = j = 0$, where the transition time is the idle period plus either $S_L$ with probability $\lambda_i/\lambda$ or $S_D$ with probability $\lambda_D/\lambda$.

We now consider $P_{i,j,m,n}$ for each of the above disjoint subsets.
Hence, for \( N_L^d (n) \geq 0 \), \( N_D^d (n) \geq 1 \), and \( N_L^d (n) + N_D^d (n) \leq K \), we have the following relationship:

\[
\begin{align*}
\begin{cases}
(K,0), & \text{if } A_L (S_D) \geq K - N_L^d (n), \\
(N_L^d (n) + A_L (S_D), K - N_L^d (n) - A_L (S_D)), & \text{if } K - N_L^d (n) - N_D^d (n) + 1 \leq A_L (S_D), \\
(N_L^d (n) + A_L (S_D), K - N_L^d (n) - A_L (S_D)), & \text{if } 0 \leq A_L (S_D) \leq K - N_L^d (n) - N_D^d (n), \\
(N_L^d (n) + A_L (S_D), N_D^d (n) - 1 + A_D (S_D)), & \text{if } 0 \leq A_D (S_D) \leq K - N_L^d (n) - N_D^d (n), \\
\end{cases}
\end{align*}
\]

where each case corresponds to the following events: The first case occurs when the buffer is full and is monopolized by class-L packets at the end of the transition. The second and third cases occur when the buffer is full but is not monopolized by class-L packets at the end of the transition, and all the \( A_L (S_D) \) class-L arrivals during the transition join the buffer. The main difference between the second and third cases is whether \( N_L^d (n) + N_D^d (n) - 1 + A_L (S_D) \geq K \) or not. If \( N_L^d (n) + N_D^d (n) - 1 + A_L (S_D) \geq K \), then none of the class-D arrivals can remain in the buffer by the end of the transition. Hence, we can only consider the marginal probabilities of \( A_L (S_D) \) to derive the transition probability. However, if \( N_L^d (n) + N_D^d (n) - 1 + A_L (S_D) < K \), some of the \( A_D (S_D) \) class-D arrivals can remain in the buffer at the end of the transition. Hence, we need to consider the joint probabilities of \( A_L (S_D) \) and \( A_D (S_D) \) to derive the transition probability. The last case occurs when the buffer is not full at the end of this transition, and all arrivals of both classes during the transition successfully join the buffer. Therefore, from the above relationship of \( (N_L^d (n), N_D^d (n)) \) and \( (N_L^d (n+1), N_D^d (n+1)) \), the transition probabilities \( P_{(i,j),(l,m)} \), \( i \geq 0 \), \( j \geq 1 \), \( i + j \leq K \), are given by

\[
P_{(i,j),(l,m)} = \begin{cases}
\sum_{n=K-i}^{\infty} a_{D_{(n)}}^l > 0, & l = K, m = 0, \\
0, & \text{otherwise.}
\end{cases}
\]

3.2. \( P_{(i,j),(l,m)} \) When \( 1 \leq i \leq K \) and \( j = 0 \). If \( N_L^d (n) \geq 1 \) and \( N_D^d (n) = 0 \), then the transition time between the \( n \)-th and \( (n+1) \)-th embedded points is \( S_L \). Hence, one class-L packet gets serviced and departs from the system, and \( A_L (S_L) \) class-L packets and \( A_D (S_L) \) class-D packets arrive at the system during this transition. Notice that there are no class-D packets and \( (N_L^d (n) - 1) \) class-L packets in the buffer at the beginning of the transition because one class-L packet enters its service then. Since class-L packets can push out class-D packets in the buffer, the effective buffer size for class L is \( K \) and that for class D becomes \( K - N_L^d (n+1) \) by the end of this transition. Hence, the state \( (N_L^d (n+1), N_D^d (n+1)) \) at the end of this transition is determined as
\[
\begin{align*}
(N_L^d(n + 1), N_D^d(n + 1)) \\
= & \left\{ \min\{N_L^d(n) - 1 + A_L(S_L), K\}, \min\{A_D(S_L), K - N_L^d(n + 1)\} \right\} \\
= & \begin{cases} 
(K, 0), & A_L(S_L) \geq K - N_L^d(n) + 1, \\
\left( N_L^d(n) - 1 + A_L(S_L), K - N_L^d(n) + 1 - A_L(S_L) \right), & 0 \leq A_L(S_L) \leq K - N_L^d(n), \\
\left( N_L^d(n) - 1 + A_L(S_L), A_D(S_L) \right), & 0 \leq A_L(S_L) \leq K - N_L^d(n) - A_L(S_L), 
\end{cases}
\end{align*}
\]

Therefore, for \(1 \leq i \leq K\), we have

\[
P(i,0),(l,m) = \begin{cases} 
\sum_{n=K+1}^{\infty} a_{(l,m)}^L > 0, & l = K, m = 0, \\
\sum_{n=K-l}^{\infty} a_{(l+1,n)}^L > 0, & 0 \leq l \leq K - 1, l + m = K, \\
\sum_{n=K-l}^{\infty} a_{(l+1,m)}^L > 0, & l \geq i - 1, m \geq 0, l + m \leq K - 1, \\
0 & \text{otherwise.}
\end{cases}
\]

3.3. \(P(i,j),(l,m)\) When \(i = j = 0\). In this case, the transition time consists of the idle period plus either \(S_L\) with probability \(\lambda_L/\lambda\) or \(S_D\) with probability \(\lambda_D/\lambda\), and the one packet which arrives during the idle period is serviced and departs from the system on this transition. Since there are neither class-\(L\) packets nor class-\(D\) packets in the buffer at the beginning of the service of the packet which arrives during the idle period, if the class of packet which arrives in the idle period is \(h\), then the following relationship holds for \(h = L, D\):

\[
\begin{align*}
\left( N_L^d(n + 1), N_D^d(n + 1) \right) \\
= & \left\{ \min\{A_L(S_h), K\}, \min\{A_D(S_h), K - N_L^d(n + 1)\} \right\} \\
= & \begin{cases} 
(K, 0), & A_L(S_h) \geq K, \\
A_L(S_h), & 0 \leq A_L(S_h) \leq K - 1, A_D(S_h) \geq K - A_L(S_h), \\
\left( A_L(S_h), A_D(S_h) \right), & 0 \leq A_L(S_h) \leq K - 1, 0 \leq A_D(S_h) \leq K - A_L(S_h) - 1.
\end{cases}
\end{align*}
\]

Therefore, we have

\[
P(0,0),(l,m) = \begin{cases} 
\sum_{h=L,D} \frac{\lambda_h}{\lambda} \sum_{n=K}^{\infty} a_{(l,m)}^h > 0, & l = K, m = 0, \\
\sum_{h=L,D} \frac{\lambda_h}{\lambda} \sum_{n=K-l}^{\infty} a_{(l,m)}^h > 0, & 0 \leq l \leq K - 1, l + m = K, \\
\sum_{h=L,D} \frac{\lambda_h}{\lambda} a_{(l,m)}^h > 0, & l \geq 0, m \geq l, l + m \leq K - 1, \\
0 & \text{otherwise.}
\end{cases}
\]
Finally, from (10), (12), and (14), we can construct all the partial transition matrices $P_{ij}$, $0 \leq i \leq K$, max $(0, i - 1) \leq l \leq K$, in the whole transition probability matrix $P$, and their special structures are presented in Section A in the appendix. From the special structures of the partial transition matrices and the whole transition matrix, we have the following lemma.

**Lemma 1.** The EMC with transition probability $P$ is an irreducible ergodic Markov chain.

The rigorous proof of Lemma 1 is given in Section A in the appendix. Here, a brief explanation of the irreducibility and ergodicity of the EMC is given. In order for the EMC to be irreducible, all the states can reach each other. From the M/G/1-type structure of $P$ in (4) and $P_{ij}$ in (A.3), it is easy to see that all the levels can reach each other and that all the states within the same level can also reach each other. Therefore, the EMC is irreducible. Notice that if an irreducible finite-state Markov chain is aperiodic, then it is an ergodic Markov chain, whose limiting probabilities converge to a positive value (see [[25], Sec. 7.5]). Notice also that if, starting from a state, the Markov chain can return back to that state in one-step transition, then it is aperiodic at the state (see [[26], Sec. 4.2]). From the structure of $P$ in (A.3), for each state, the one-step transition probability to itself is $p_{(i,j),(i,j)} > 0$. Thus, the EMC is aperiodic and, therefore, ergodic.

Since, from Lemma 1, the EMC is an irreducible ergodic Markov chain, there exists a positive limiting probability vector $\pi$, which is also a unique stationary probability vector that satisfies (5) (see [[25], Secs. 7.5-7.6]). Therefore, if we compute the stationary probability vector that satisfies (5), then the limiting probability vector $\pi$ of the EMC is obtained.

For brevity, let $G_i$ denote the hitting probability matrix $G_{ij}$ from level $i$ to the next lower level $i - 1$.

**Lemma 2.** The hitting probability matrices $G_i$, $1 \leq i \leq K$, are given by the recursive formula

$$G_i = \left( I_{(K-i-1) \times (K-i-1)} - \sum_{l=i+1}^{K} P_{ij} G_{ij} - P_{ij} \right)^{-1} P_{ij},$$

where $I_{ixi}$ is the $i \times i$ identity matrix and

$$G_{ij} = G_i G_{i-1} \cdots G_1, \quad 1 \leq i \leq l \leq K.$$  

In addition, all the elements of $G_i$ are positive, and $G_i$ is a stochastic matrix; i.e.,

$$G_i > 0, \quad G_i e_{(K-i+2) \times 1} = e_{(K-i+1) \times 1}, \quad 1 \leq i \leq K,$$

where $e_{x \times 1}$ is an $i$-dimensional column vector whose elements are all one.

The proof of Lemma 2 is given in Section B in the appendix.

We now derive the partial probability vectors $\pi_j$, $0 \leq i \leq K$. To this end, we consider the following censored Markov chain $H(k)$, $0 \leq k \leq K$, from the original EMC, where we only observe the original EMC when the state is within levels 0, 1, \ldots, $k$. (Censored Markov chains are also known as restricted or watched Markov chains and have been used for the analysis of Markov chains which have partitioned structure. See [[28], 29] and the references therein.) We first define $h_{(i,j),(m)}(k)$ to be the transition probability of the censored Markov chain $H(k)$ from state

4. An Algorithm to Compute the Stationary Probability Vector of the EMC

In general, $\pi$ that satisfies (5) can be obtained by one of numerical methods for computing stationary probability vectors such as those described in [[27]. However, if the buffer size $K$ is large, then the dimension of the matrix $P$ becomes also large, $(K + 1)(K + 2)/2 \times (K + 1)(K + 2)/2$, so the computation of $\pi$ can be computationally expensive. For instance, if we use the LU decomposition method, one of the standard numerical methods for stationary probability vectors, to compute $\pi$, then the numerical method requires $O(K^6)$ floating-point operations because, for the LU decomposition of an $n \times n$ matrix, $n^2/3$, floating-point additions and multiplications are needed. In addition, as seen in (4), the transition probability matrix $P$ has a special structure. Hence, if we exploit this probability structure, then we can not only develop a more efficient algorithm to obtain the stationary probability vector $\pi$, but also gain insight into the behavior of the proposed queueing model.

Observe first that, from the M/G/1-type structure of $P$ in (4), the EMC is left-skip-free for levels. That is, any path from level $i$ to level $l$, $0 \leq i \leq l - 1$, must go through every intermediate level $l', l < l' < j$, at least once on this path. We now consider the hitting probability from level $i$ to level $l$, $0 \leq i \leq l \leq K$. The hitting probability $g_{(i,j),(l,m)}$ is defined to be the probability that, starting from state $(i, j)$, the first state visited at level $l$ is the state $(l, m)$ for $0 \leq l < i \leq K$, $0 \leq j \leq K - i$, $0 \leq m \leq K - l$. We define also the hitting probability matrix $G_{ij}$ from level $i$ to level $l$ to be
Lemma 4. The partial transition matrix \( H_{i,j}(k) \), \( 0 \leq i, j \leq k \), \( 0 \leq j \leq K - i \), \( 0 \leq m \leq K - i \). We also define the partial transition probability matrix \( H_{i,j}(k) \) from level \( i \) to level \( l \) to be

\[
H_{i,l}(k) = \begin{bmatrix}
    h_{(i,0),(l,0)}(k) & h_{(i,0),(l,1)}(k) & \cdots & h_{(i,0),(l,K-l)}(k) \\
    h_{(i,1),(l,0)}(k) & h_{(i,1),(l,1)}(k) & \cdots & h_{(i,1),(l,K-l)}(k) \\
    \vdots & \vdots & \ddots & \vdots \\
    h_{(i,K-1),(l,0)}(k) & h_{(i,K-1),(l,1)}(k) & \cdots & h_{(i,K-1),(l,K-l)}(k)
\end{bmatrix}, \quad 0 \leq i, l \leq k,
\]

and the whole transition probability matrix \( H(k) \) to be

\[
H(k) = \begin{bmatrix}
    H_{0,0}(k) & H_{0,1}(k) & \cdots & H_{0,k}(k) \\
    H_{1,0}(k) & H_{1,1}(k) & \cdots & H_{1,k}(k) \\
    \vdots & \vdots & \ddots & \vdots \\
    H_{k,0}(k) & H_{k,1}(k) & \cdots & H_{k,k}(k)
\end{bmatrix}_{(k+1)(2(K+1)-k)/2}
\]

(19)

Lemma 3. The censored Markov chain \( H(k) \), \( 0 \leq k \leq K \), is an irreducible ergodic Markov chain, and the transition probability matrix is given by

\[
H(k) = \begin{bmatrix}
    P_{0,0} & P_{0,1} & P_{0,2} & \cdots & P_{0,k-2} & P_{0,k-1} & H_{0,k}(k) \\
    P_{1,0} & P_{1,1} & P_{1,2} & \cdots & P_{1,k-2} & P_{1,k-1} & H_{1,k}(k) \\
    0 & P_{2,1} & P_{2,2} & \cdots & P_{2,k-2} & P_{2,k-1} & H_{2,k}(k) \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & P_{k-1,k-2} & P_{k-1,k-1} & H_{k-1,k}(k) \\
    0 & 0 & 0 & \cdots & 0 & P_{k,k-1} & H_{k,k}(k)
\end{bmatrix}_{(k+1)(2(K+1)-k)/2}
\]

(20)

where

\[
H_{i,k}(k) = P_{i,k} + \sum_{l=k+1}^{K} P_{i,l} G_{l,k}, \quad 0 \leq i \leq k.
\]

(22)

The proof of Lemma 3 is given in Section C in the appendix.

From the structure of the censored Markov chain \( H(k) \) in (21), if we obtain \( H_{i,k}(k) \), \( 0 \leq i \leq k \), then we can construct the whole transition matrix \( H(k) \). Thus, the transition probability matrices \( H(k) \), \( H(K-1), \ldots, H(0) \) can be constructed from the following recursive formula in Lemma 4.

Lemma 4. The partial transition matrix \( H_{i,k}(k) \), \( 0 \leq i \leq k \leq K \), can be calculated from the following recursive equations

\[
H_{i,k}(k) = P_{i,k}, \quad 0 \leq i \leq K,
\]

\[
H_{i,k}(k) = P_{i,k} + H_{i,k+1}(k+1)G_{k+1}, \quad 0 \leq k \leq K - 1.
\]

(23)

Furthermore, \( G_k \) can be calculated from \( H_{k,k}(k) \) as

\[
G_k = (I - H_{k,k}(k))^{-1} P_{k,k-1}, \quad 1 \leq k \leq K.
\]

(24)

The proof of Lemma 4 is given in Section D in the appendix.

Since the censored Markov chain \( H(k) \) is an irreducible ergodic Markov chain from Lemma 3, there exists the unique stationary probability vector \( \pi(k) \) that satisfies the following equations:

\[
\pi(k) = \pi(k)H(k), \quad \pi(k)e = 1.
\]

(25)

The stationary probability vector \( \pi(k) \) for the censored Markov chain \( H(k) \) and a portion of the stationary probability vector \( \pi \) of the original EMC \( P \) have a proportional relationship, described in Lemma 5.

Lemma 5. If we define the partial stationary probability vector \( \pi(k) \) associated with the states from level 0 to level \( k \) in the original EMC to be

\[ \pi(k) = [\pi_0, \pi_1, \ldots, \pi_k]_{1 \leq k+1} = 0 \leq k \leq K, \quad (26) \]

then the vector \( \pi(k) \) satisfies the following equation:

\[ \pi(k) = \pi(k)H(k), \quad 0 \leq k \leq K, \quad (27) \]

which implies the following proportional relationship:

\[ h(k) = \frac{\pi(k)}{\pi(k)e_{(k+1)(2(k+1)-k)/2}}, \quad 0 \leq k \leq K. \quad (28) \]

The proof of Lemma 5 is given in Section E in the appendix.

Finally, if we use the proportional relationship of the stationary probability vectors in Lemma 5, then we can recursively compute the vectors \( h_k \), \( 0 \leq k \leq K \), which differ from the partial probability vectors \( \pi_k \) by a constant multiplier, as described in Theorem 1.

**Theorem 1.** The partial probability vector \( \pi_k \) is given by

\[ \pi_k = \beta h_k, \quad 0 \leq k \leq K, \quad (29) \]

where \( h_k \) is the \( 1 \times (K + 1) \) unique stationary probability vector that satisfies

\[ h_0 = h_0H_{h0}(0), \]

\[ h_0e_{(K+1)x1} = 1. \quad (30) \]

\( h_k \) are \( 1 \times (K - k + 1) \) vectors that are given by the recursion formula:

\[ h_k = \sum_{l=0}^{k-1} h_kH_{hl}(k)(1 - H_{hl}(k))^{-1}, \quad 1 \leq k \leq K, \quad (31) \]

and \( \beta \) is the following positive constant:

\[ \beta = \frac{1}{\sum_{k=0}^{K} h_k e_{(K-k+1)x1}}. \quad (32) \]

**Proof.** From (21), for \( k = 0 \), we have \( H(0) = H_{h0}(0) \). Thus, from Lemma 3, \( H_{h0}(0) \) is the transition probability matrix of the irreducible ergodic Markov chain \( H(0) \). Therefore, there exists a unique vector \( h_0 \) that satisfies (30), which is also identical to \( h(0) \). From (28), we also have

\[ \pi_0 = \pi(0) = \beta h(0) = \beta h_0, \quad (33) \]

where \( \beta = \pi_0 e_{(K+1)x1} \), which is an unknown multiplier so far. Moreover, from (21) and (27), we have \( \pi_l = \sum_{i=0}^{l} \pi_iH_{il}(l), \quad 1 \leq l \leq K \), which leads to

\[ \pi_l = \sum_{i=0}^{l-1} \pi_iH_{il}(l)(1 - H_{il}(l))^{-1}, \quad 1 \leq l \leq K. \quad (34) \]

Suppose now that (29) and (31) hold for all \( k \) less than a certain level \( l, \quad 1 \leq l \leq K \). In addition, define \( h_k \) as (31) for \( k = l \). Then, since we assume that (29) holds for \( k = 0, 1, \ldots, l - 1 \), from the above equation, we have

\[ \pi_l = \beta \sum_{i=0}^{l-1} h_iH_{il}(l)(1 - H_{il}(l))^{-1} = \beta h_l. \quad (35) \]

Therefore, by induction, (29) and (31) hold for \( 1 \leq k \leq K \). In addition, from the normalization condition of \( \pi \), we have

\[ \sum_{k=0}^{K} \pi_k e_{(K-k+1)x1} = \beta \sum_{k=0}^{K} h_k e_{(K-k+1)x1} = 1, \quad (36) \]

which leads to (32).

When \( K \geq 1 \), from Lemmas 2–5 and Theorem 1, we can compute \( \pi \) from the following algorithm.

Compared to the methods of directly computing \( \pi \) from (5), which has a computational complexity of \( O(K^6) \), the proposed algorithm has a computational complexity of \( O(K^3) \) (see Section F in the appendix). Hence, the proposed algorithm is about \( K \times \) times more computationally efficient, and this can be significant for performance analysis, especially when \( K \) is large and we need to obtain performance measures for multiple combinations of system parameters. Additionally, in the course of developing the algorithm, we can gain insight into the behavior of the proposed queueing model. For example, (16) and (17) for the hitting probability, even though \( h \) is given by the recursive relationship, \( h \) can gain insight into the behavior of the proposed queueing model, even if the system is in a steady state. Let also \( \xi \) denote the class of a packet in service when the server is busy. Given \( \xi = h, \quad h \in \{L, D\} \), let \( S_h^e \) and \( S_h^r \) denote the elapsed and remaining service times in a steady state, respectively. Since there can be at most \( K \) packets in the buffer and at most one packet in the server, \( N_L + N_D \leq K + 1 \). Define the joint distributions of \( N_L, N_D, \xi, \) and \( S_h^e \) as

\[ Q_{(i,j)}(t) = \Pr[N_L = i, N_D = j, \xi = h, S_h^e \leq t], \quad i, j \geq 0, 1 \leq i + j \leq K + 1. \quad (37) \]
(1) Compute $G_i$, $1 \leq i \leq K$.

1-1. Set $i = K$.
1-2. From (16), compute $G_i$.
1-3. For all $i$, $i + 1 \leq T \leq K$, compute $G_{i+1} = G_i \times G_i$. Moreover, set $G_{i+1} = G_i$.
1-4. Set $i = i - 1$.
1-5. If $i \geq 1$, then go to step 1-2. Otherwise, go to step 2.

(2) Compute $H_{i,j}$, $0 \leq j \leq K$.
2-1. For $0 \leq j \leq K$, set $H_{i,j}(K) = P_{i,j}$.
2-2. Set $k = K$.
2-3. For $0 \leq k \leq k - 1$, compute $H_{i,j-k}(k - 1)$ from (23).
2-4. Set $k = k - 1$.
2-5. If $k \geq 1$, then go to step 2-3. Otherwise, go to step 3.

(3) Compute $h_k$, $0 \leq k \leq K$.
3-1. Obtain a vector $h_k$ that satisfies (30).
3-2. Set $k = 1$.
3-3. Compute $h_k$ from (31).
3-4. $K = k + 1$.
3-5. If $k \leq K$, then go to step 3-3. Otherwise, go to step 4.

(4) Normalize $\pi$.
4-1. Compute $\beta = 1/\sum_{k=1}^{K} h_k \cdot e^{(k-1)}$.
4-2. For $0 \leq k \leq K$, set $\pi_k = \beta h_k$.

\textbf{Algorithm 1:} An algorithm to compute the stationary probability vector of the EMC.

Define also the corresponding joint probabilities as

\[ q_{(i,j,h)}(t) dt = dQ_{(i,j,h)}(t) \]
\[ = \Pr[N_L = i, N_D = j, \xi = h, t \leq S_t < t + dt]. \]

(38)

\[ Q_{(i,0)} = Q_{(i,0,L)}(\infty) = \int_{t=0}^{\infty} q_{(i,0,L)}(t) dt, 1 \leq i \leq K + 1, \]
\[ Q_{(0,j)} = Q_{(0,j,D)}(\infty) = \int_{t=0}^{\infty} q_{(0,j,D)}(t) dt, 1 \leq j \leq K + 1, \]
\[ Q_{(i,j)} = Q_{(i,j,L)}(\infty) + Q_{(i,j,D)}(\infty) = \int_{t=0}^{\infty} q_{(i,j,L)}(t) dt + \int_{t=0}^{\infty} q_{(i,j,D)}(t) dt, i, j \geq 1, i + j \leq K + 1. \]

(39)

Therefore, if we have $Q_{(0,0)}$ and $Q_{(i,j,h)}(t)$, $i, j \geq 0$, $1 \leq i + j \leq K + 1$, $h \in \{L, D\}$, then all $Q_{(i,j)}$, $i, j \geq 0$, $i + j \leq K + 1$, can be obtained.

To derive $Q_{(0,0)}$ and $Q_{(i,j,h)}(t)$, we consider the semi-Markov process $((N_L^d(n), N_D^d(n)), T(n))$ associated with the EMC $(N_L^d(n), N_D^d(n))$, where $T(n)$ is the transition time from the $n$-th embedded point until the $(n+1)$-th embedded point. Due to the time priority of class-$D$ packets over class-$L$ packets, we have

\[ T(n) = \begin{cases} 
S_D, & \text{if } N_D^d(n) \geq 1, \\
S_L, & \text{if } N_D^d(n) \geq 1, N_L^d(n) = 0, \\
I + S_D, & \text{if } N_D^d(n) = N_D^d(n) = 0, (A_L(I), A_D(I)) = (0, 1), \\
I + S_L, & \text{if } N_L^d(n) = N_D^d(n) = 0, (A_L(I), A_D(I)) = (1, 0), 
\end{cases} \]

(40)

where $I$ is the length of the idle period which follows an exponential distribution with rate $\lambda$, and $A_g(I)$, $g \in \{L, D\}$.
is the number of class-$g$ packets which arrive during the idle period. Thus, if we let $T = \lim_{n \to \infty} T(n)$, then from the semi-Markov process theory we have (see [[25], Sec. 9.1])

$$E[T] = E[S_D]\left(\sum_{i=0}^{K} \sum_{j=1}^{K-i} \pi(i,j) + \frac{\lambda_D}{\lambda} \cdot \pi(0,0)\right)$$

$$+ E[S_L]\left(\sum_{i=1}^{K} \frac{\lambda_D}{\lambda} \cdot \pi(0,0)\right) + \frac{1}{\lambda} \cdot \pi(0,0).$$

(41)

Furthermore, since the system is empty only during the idle period, we have

$$Q(0,0) = \frac{(1/\lambda)\pi(0,0)}{E[T]}.$$  \hspace{1cm} (42)

To derive $g_{(i,j)}(t)$, we now consider the relationship between the state $(N^d_L, N^d_D, \xi, S^d)$ at an arbitrary time when the system is busy and the state $(N^d_L, N^d_D)$ at the last embedded point before the arbitrary time. Notice that if $N^d_L + N^d_D \geq 1$ at the last embedded point, then the time between the last embedded point and the arbitrary time is the elapsed service time $S^d$. If $N^d_L = N^d_D = 0$, then an idle period starts at the last embedded point, a packet which arrives in the idle period starts a busy period following the idle period, and the time between the beginning of the service of the packet and the arbitrary time is $S^d$. Hence, if we let $A_g(S^d)$, $g \in \{L, D\}$, denote the number of class-$g$ packets which arrive during the elapsed service time $S^d$, then from arguments analogous to those used to derive (10), (12), and (14) in Section 3, we can derive the relationship between the state $(N^d_L, N^d_D, \xi, S^d)$ at the arbitrary time and the state $(N^d_L, N^d_D)$ at the last embedded point as follows.

Suppose $N^d_L \geq 0$, $N^d_D \geq 1$, and $N^d_L + N^d_D \leq K$. Then, a class-$D$ packet enters its service at the last embedded point. Additionally, the effective buffer size for class-$L$ packets is $K$, and that for class-$D$ packets becomes $K - N^d_L$ at the arbitrary time because $N^d_L$ class-$D$ packets are in the buffer at that time. Considering one class-$D$ packet in service and the effective buffer size of each class, we deduce that the maximum numbers of class-$L$ and $D$-packets in the system at the arbitrary time are $K$ and $K - N^d_L + 1$, respectively. Thus, we have

$$N^d_L = \min\{N^d_L + A_L(S^d), K\},$$

$$N^d_D = \min\{N^d_D + A_D(S^d), K - N^d_L + 1\},$$

$$\xi = D.$$  \hspace{1cm} (43)

Suppose next $1 \leq N^d_L \leq K$ and $N^d_D = 0$. Then, a class-$L$ packet enters its service at the last embedded point. Furthermore, the effective buffer size for class-$L$ packets is $K$, and that for class-$D$ packets becomes $K - N^d_L + 1$ at the arbitrary time because $(N^d_L - 1)$ class-$L$ packets are in the buffer at that time. Considering one class-$L$ packet in service and the effective buffer size of each class, we have

$$N^d_L = \min\{N^d_L + A_L(S^d), K + 1\},$$

$$N^d_D = \min\{1 + A_L(S^d), K - A_L(S^d) + 1\},$$

$$\xi = L.$$  \hspace{1cm} (44)

Similarly, if $(A_L(I), A_D(I)) = (0, 1)$, then a class-$D$ packet enters its service at the end of the idle period. Since there are one class-$D$ packet in service and no packets in the buffer at that point, we have

$$N^d_L = \min\{1 + A_L(S^d), K + 1\},$$

$$N^d_D = \min\{A_D(S^d), K - A_L(S^d)\},$$

$$\xi = D.$$  \hspace{1cm} (45)

Therefore, the relationship between the state $(N^d_L, N^d_D, \xi, S^d)$ at the arbitrary time during the busy period and the state $(N^d_L, N^d_D)$ at the last embedded point can be summarized by Table 1.

In addition, from the renewal theory, the joint distribution of $S^d_L$ and $S^d_D$ is given by

$$\Pr\{x \leq S^d_L < x + dx, t \leq S^d_D < t + dt\} = \frac{dS^d_L(x + t)dt}{E[S^d_L]},$$

$$x, t \geq 0, h \in \{L, D\}.$$  \hspace{1cm} (47)

$$\frac{dS^d_L(x + t)dt}{E[S^d_L]}.$$

Thus, the joint distributions of $A_L(S^d_L)$, $A_D(S^d_D)$, and $S^d$ are given by

$$\tilde{a}_h^{(i,j)}(t)dt = \Pr\{A_L(S^d_L) = i, A_D(S^d_D) = j, t \leq S^d < t + dt\}$$

$$= \frac{1}{E[S^d]} \int_{x=0}^{\infty} \frac{\lambda_L x^i}{i!} (\lambda_D x)^j e^{-(\lambda_L x + \lambda_D x)} dS^d L(x + t)dt, i, j \geq 0, t \geq 0, h \in \{L, D\}$$

$$\tilde{a}_h^{(i)}(t)dt = \Pr\{A_L(S^d_L) = i, t \leq S^d < t + dt\}$$

$$= \frac{1}{E[S^d]} \int_{x=0}^{\infty} \frac{\lambda_L x^i}{i!} e^{-(\lambda_L x + \lambda_D x)} dS^d L(x + t)dt, i \geq 0, t \geq 0, h \in \{L, D\}.\hspace{1cm} (48)$$
Using the semi-Markov process theory, from the relationships in the first four rows and the eighth, ninth, and tenth rows in Table 1, we have

\[
\Pr \{N_L = i, N_D = j, \xi = D, t \leq S_D' < t + dt\} = \sum_{l=0}^{K} \sum_{m=0}^{K} \Pr \{N_L^d = l, N_D^d = m, \xi = D\} \\
\times \Pr \{N_L = i, N_D = j, t \leq S_D' < t + dt; N_L^d = l, N_D^d = m, \xi = D\} \\
= \sum_{l=0}^{K} \sum_{m=0}^{K} \pi_{(l,m)} E[S_D] / E[T] \cdot \Pr \{\min[l + A_L(S_D'), K] = i, \min[m + A_D(S_D'), K - i + 1] = j, t \leq S_D' < t + dt\} \\
+ \pi_{(0,0)} A_D / E[T] \cdot \Pr \{\max[A_L(S_D'), K - A_L(S_D') + 1] = j, t \leq S_D' < t + dt\},
\]

which leads to

<table>
<thead>
<tr>
<th>Possible states at the embedded point</th>
<th>Arrivals between the embedded point and the arbitrary time</th>
<th>((N_L^d, N_D^d, \xi, S_D'))</th>
<th>Possible states at the arbitrary time</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_L^d \geq 0, N_D^d \geq 1, N_L^d + N_D^d \leq K)</td>
<td>(0 \leq A_L(S_D') + A_D(S_D') \leq K - N_L^d - N_D^d)</td>
<td>((N_L^d + A_L(S_D'), N_D^d + A_D(S_D'), D, S_D'))</td>
<td>(N_L \geq N_L^d \geq 0, N_D \geq N_D^d \geq 1, N_L + N_D \leq K, \xi = D)</td>
</tr>
<tr>
<td>(0 \leq A_L(S_D') + A_D(S_D') \leq K - N_L^d - N_D^d)</td>
<td>((N_L^d + A_L(S_D'), K + 1 - N_L^d - A_D(S_D'), D, S_D'))</td>
<td></td>
<td>(0 \leq N_L^d \leq N_L \leq K - N_D^d)</td>
</tr>
<tr>
<td>(K + 1 - N_L^d - N_D^d \leq A_L(S_D') \leq K - 1 - N_L^d)</td>
<td>((N_L^d + A_L(S_D'), K + 1 - N_L^d - A_D(S_D'), D, S_D'))</td>
<td></td>
<td>(N_L + N_D = K + 1, \xi = D)</td>
</tr>
<tr>
<td>(A_L(S_D') \geq K - N_L^d)</td>
<td>((K, 1, D, S_D'))</td>
<td></td>
<td>(N_L = K, N_D = 1, \xi = D)</td>
</tr>
<tr>
<td>(1 \leq N_L^d \leq K, N_D^d = 0)</td>
<td>(0 \leq A_L(S_D') + A_D(S_D') \leq K - N_L^d)</td>
<td>((N_L^d + A_L(S_D'), A_D(S_D'), L, S_D'))</td>
<td>(N_L \geq N_L^d \geq 1, N_D \geq N_D^d = 0, N_L + N_D \leq K, \xi = L)</td>
</tr>
<tr>
<td>(0 \leq A_L(S_D') \leq K - 1 - N_L^d, A_L(S_D') \geq K + 1 - N_L^d)</td>
<td>((N_L^d + A_L(S_D'), K + 1 - N_L^d - A_D(S_D'), L, S_D'))</td>
<td></td>
<td>(1 \leq N_D^d \leq N_D = K, \xi = L)</td>
</tr>
<tr>
<td>(A_L(S_D') \geq K + 1 - N_L^d)</td>
<td>((K + 1, 0, L, S_D'))</td>
<td></td>
<td>(N_L = K + 1, N_D = 0, \xi = L)</td>
</tr>
<tr>
<td>(N_L^d = N_D^d = 0)</td>
<td>((A_L(I), A_D(I)) = (0, 1))</td>
<td>((A_L(S_D'), 1 + A_D(S_D'), D, S_D'))</td>
<td>(N_L \geq 0, N_D \geq 1, N_L + N_D \leq K, \xi = D)</td>
</tr>
<tr>
<td>(0 \leq A_L(S_D') \leq K - 1, A_L(S_D') \geq K + 1)</td>
<td>((A_L(S_D'), K + 1 - A_D(S_D'), D, S_D'))</td>
<td></td>
<td>(0 \leq N_L \leq K - 1)</td>
</tr>
<tr>
<td>(A_L(S_D') + A_D(S_D') \geq K)</td>
<td>((K, 1, D, S_D'))</td>
<td></td>
<td>(N_L = K, N_D = 1, \xi = D)</td>
</tr>
<tr>
<td>((A_L(I), A_D(I)) = (0, 1))</td>
<td>((K + 1, 0, L, S_D'))</td>
<td></td>
<td>(N_L \geq 1, N_D \geq 0)</td>
</tr>
<tr>
<td>(A_L(S_D') \geq K + 1 - N_L^d)</td>
<td>((1 + A_L(S_D'), A_D(S_D'), L, S_D'))</td>
<td></td>
<td>(1 \leq N_D \leq K)</td>
</tr>
<tr>
<td>(A_L(S_D') + A_D(S_D') \leq K - 1)</td>
<td>((1 + A_L(S_D'), K - A_D(S_D') + 1, L, S_D'))</td>
<td></td>
<td>(N_L = K + 1, N_D = 0, \xi = L)</td>
</tr>
<tr>
<td>((A_L(I), A_D(I)) = (1, 0))</td>
<td>((K + 1, 0, L, S_D'))</td>
<td></td>
<td>(N_L = K + 1, N_D = 0, \xi = L)</td>
</tr>
</tbody>
</table>
The joint probabilities $Q(i,j,i,D)$ of $N_L$ and $N_D$, $i,j \geq 0$, $i+j \leq K+1$, which will be utilized to derive the loss probabilities and mean queue lengths of both classes in Section 5. Moreover, the joint probabilities $q(i,j,D)(t)dt$ of the queue lengths and the remaining service time will be used to derive the waiting time distribution of a class-$D$ packet.

### 6. Loss Probabilities and Mean Waiting Times

There are two types of packet loss under the push-out priority discipline. One occurs when a packet is blocked and lost on its arrival, and the other occurs when a class-$L$ packet pushes out a class-$D$ packet in the buffer. Let $p_{(g)}^b$, $g \in \{L,D\}$, denote the probability that a class-$g$ packet is blocked and lost on its arrival because the buffer is full of packets which have space priority higher than or the same as that of the class-$g$ packet. Let $P_{(g)}^b$ also denote the probability that a class-$L$ packet pushes out a class-$D$ packet in the buffer on arrival of the class-$L$ packet.

Under the push-out priority discipline, class-$D$ arrivals are always blocked if the buffer is full on their arrivals. Hence, from PASTA, the class-$D$ blocking probability $P_D^b$ is given by

$$P_D^b = \Pr\{N_L + N_D = K + 1\} = \sum_{i=0}^{K} Q(i,K+1-i). \quad (52)$$

However, class-$L$ arrivals can push out class-$D$ packets in the buffer even though the buffer is full. Thus, $P_L^b$ is not identical to $P_D^b$. Note that class-$L$ packets are blocked only when the buffer is full of $K$ class-$L$ packets and that, when
\[ \xi = L, \] there is one more class-L packet in service other than class-L packets in the buffer. Thus, from PASTA, the class-L blocking probability \( P_L^b \) is given by
\[
P_L^b = \Pr[N_L = K + 1, \xi = L] + \Pr[N_L = K, \xi = D] = Q_{(K+1,0,L)} + Q_{(K,1,D)}. \tag{53}
\]

Note also that a class-L packet pushes out a class-D packet in the buffer if the buffer is full and there are one or more class-D packets in the buffer on arrival of the class-L packet, and that, when \( \xi = D \), there is one more class-D packet in service other than class-D packets in the buffer. Thus, from PASTA, we have
\[
P_L^{po} = \Pr[N_L + N_D = K + 1, N_D \geq 1, \xi = L] + \Pr[N_L + N_D = K + 1, N_D \geq 2, \xi = D] = \sum_{i=1}^{K-1} Q_{(i,K+1-i,L)} + \sum_{i=0}^{K-1} Q_{(i,K+1-i,D)}. \tag{54}
\]

From (52)-(54), we also have
\[
P_D^b = P_D^b + P_D^{po}. \tag{55}
\]

Let \( \lambda_g^b, \lambda_g^{po} \), and \( \lambda_g^s \), \( g \in \{L, D\} \), denote the blocked rate, push-out rate, and served rate of class-\( g \) packets. Then, since all class-L arrivals which are not blocked are served and all class-D arrivals which are neither blocked nor pushed out are served, we have
\[
\begin{align*}
\lambda_g^b &= \lambda_g P_g^b, g \in \{L, D\}, \\
\lambda_g^{po} &= \lambda_g P_g^{po}, \\
\lambda_g^s &= \lambda_g - \lambda_g^b = \lambda_g (1 - P_g^b), \\
\lambda_D - \lambda_D^{po} &= \lambda_D (1 - P_D^b) - \lambda_D P_D^{po}.
\end{align*}
\tag{56}
\]

In addition, since class-D packets arrive at rate \( \lambda_D \) and get pushed out at rate \( \lambda_D^{po} \), the probability \( P_D^{po} \), that a class-D arrival is pushed out in a steady state is given by
\[
P_D^{po} = \frac{\lambda_D - \lambda_D^{po}}{\lambda_D} = \frac{\lambda_D^s}{\lambda_D} \tag{57}
\]
Furthermore, if we let \( L_g \) and \( L_D^g, g \in \{L, D\} \), denote the mean number of class-\( g \) packets in the system and in the buffer in a steady state, respectively, then we have
\[
\begin{align*}
L_g &= \sum_{i=1}^{K+1-i} i \cdot Q_{(i,j)}, \\
L_D^g &= \sum_{i=1}^{K+1-i} (i - 1) \cdot Q_{(i,j,L)} + \sum_{i=1}^{K+1-i} i \cdot Q_{(i,j,D)}, \\
L_D &= \sum_{i=1}^{K+1-i} i \cdot Q_{(i,j)}, \\
L_D^g &= \sum_{i=1}^{K+1-i} j \cdot Q_{(i,j,L)} + \sum_{j=2}^{K+1-i} (j - 1) \cdot Q_{(i,j,D)}.
\end{align*}
\tag{58}
\]

Let \( W_g \) and \( W_Q^g \) denote the mean sojourn and queue waiting times of a served class-\( g \) packet in a steady state. Note that, when considering sojourn and queue waiting times, we exclude packets that are pushed out as well as being blocked. Thus, from Little's law, we have
\[
W_L = \frac{L_L}{\lambda_L}, \quad W_Q^L = \frac{L_Q^L}{\lambda_L}. \tag{59}
\]

However, the mean sojourn time and waiting time of a class-D packet cannot be obtained from Little's law because class-D packets in the buffer can be either served or pushed out, and \( L_D \) and \( L_Q^D \) take into account not only contribution by served class-D packets but also that by pushed-out class-D packets. Therefore, we employ a tagged customer approach to derive \( W_D \) and \( W_Q^D \) instead.

Suppose that we observe a tagged class-D packet in the buffer just after the \( n \)-th embedded point. Let \( X_d(n) \) denote the remaining queue waiting time of the tagged class-D packet after the \( n \)-th embedded point. Let \( N_D^{before}(n) \) also denote the number of class-D packets in the system at the \( n \)-th embedded point which arrive before the tagged packet. Notice that the tagged class-D packet is finally either served or pushed out. Observe also that class-D packets which arrive after the tagged packet affect neither the waiting time of the tagged packet nor whether the tagged packet is finally served or pushed out because a push-out happens to the last arrived class-D packet in the buffer when the buffer is full.

Since there are at most \( K \) packets in the system just after the \( n \)-th embedded point, \( 0 \leq N_D^{before}(n) \leq K - 1 \) because one slot of the buffer is occupied by the tagged packet. We now define the partial LST (Laplace–Stieltjes Transform) \( X_d^{(s)}(i, n) \) of \( X_d(n) \) as
\[
X_d^{(s)}(i, n) = \Pr\{served|N_D^{before}(n) = i, N_D^{before}(n) = j\} \times E\left[e^{-sX_d(n)}|N_D^{before}(n) = i, N_D^{before}(n) = j, served\right],
\]
\[
i, j \geq 0, i + j \leq K - 1, s \geq 0, \tag{60}
\]
where “served” is the event that the tagged packet is finally served.

Suppose that \( N_D^{before}(n) = 0 \). Then, the tagged packet enters its own service just after the embedded point, due to time priority. Hence, we have
\[
X_d^{(n)}(i, n) = 1, \quad 0 \leq i \leq K - 1. \tag{61}
\]

Suppose now that \( N_D^{before}(n) \geq 1 \), then one class-D packet which arrived before the tagged packet enters its service, and there are \( (N_D^{before} - 1) \) class-D packets in the buffer which arrive before the tagged packet just after the \( n \)-th embedded point. Thus, there are \( N_D^{before}(n) - 1 \) packets in the buffer which have effective space priority over the tagged packet, in the sense that they are not pushed out before the tagged packet is pushed out. Observe that if
more than $K - N_L^d(n) - N_D^d$ class-$L$ packets arrive between the $n$-th and $(n+1)$-th embedded points, i.e., $A_L(S_D) > K - N_L^d(n) - N_D^d$, then the tagged packet will be pushed out by the $(n+1)$-th embedded point because the buffer will become full of $K$ packets which will have effective space priority over the tagged packet by then. Thus, we have

$$\Pr\{\text{served}|N_L^d(n) = i, N_D^d(n) = j, A_L(S_D) > K - i - j\} = 0.$$  

(62)

If $A_L(S_D) \leq K - N_L^d(n) - N_D^d$, then the tagged packet is not pushed out between the $n$-th and $(n+1)$-th embedded points. Hence, the tagged packet needs to wait in the buffer during this transition of $S_D$ plus the remaining waiting time $X^d(n+1)$ after the $(n+1)$-th embedded point, when there are $N_L + A_L(S_D)$ class-$L$ packets and $(N_D^d - 1)$ class-$D$ packets which arrive earlier than the tagged packet in the buffer. Thus, from the Markovian property at embedded points, we have, for $0 \leq i \leq K - i - j$,

$$\Pr\{\text{served}|N_L^d(n) = i, N_D^d(n) = j, A_L(S_D) = l, \text{served}\} = \Pr\{\text{served}|N_L^d(n) = i, N_D^d(n) = j - 1\} \times E\left[e^{-X^d(n+1)}|N_L^d(n) = i, N_D^d(n) = j, A_L(S_D) = l, \text{served}\right].$$  

(63)

Therefore, conditioning on the numbers of class-$L$ arrivals between the $n$-th and $(n+1)$-th embedded points leads to the following recursive formula:

$$X^d_{(i,j)}(s) = \sum_{l=0}^{\infty} \Pr\{A_L(S_D) = l\} \cdot \Pr\{\text{served}|N_L^d(n) = i, N_D^d(n) = j, A_L(S_D) = l, \text{served}\} \times E\left[e^{-X^d(n)}|N_L^d(n) = i, N_D^d(n) = j, A_L(S_D) = l, \text{served}\right]$$

$$= \sum_{l=0}^{K-i-j} a^{D_s}_{i,j}(s) \cdot X^d_{(i+1,j-1)}(s; n+1), \quad i \geq 0, j \leq 1, i + j \leq K - 1,$$

where

$$a^{D_s}_{i,j}(s) = \Pr\{A_L(S_h) = l\} \cdot E\left[e^{-X^d_h}|A_L(S_h) = l\right] = \int_{r=0}^{\infty} \frac{(\lambda t)^r}{r!} e^{-(\lambda + \mu t)} dS_h(t), \quad h \in \{L, D\}, \Re(s) \geq 0.$$  

(65)

If we let $X^d = \lim_{n \to \infty} X^d(n)$ and $X^d_{(i,j)}(s) = \lim_{n \to \infty} X^d_{(i,j)}(s; n)$, then from (61) and (64), we have

$$X^d_{(i,j)}(s) = 1, 0 \leq i \leq K - 1$$

(66)

$$X^d_{(i,j)}(s) = \sum_{l=0}^{K-i-j} a^{D_s}_{i,j}(s) \cdot X^d_{(i+1,j-1)}(s), \quad i \geq 0, j \leq 1, i + j \leq K - 1.$$  

(67)

Notice that all $X^d_{(i,j)}(s)$, $i, j \geq 0, i + j \leq K - 1$, can be obtained from the recursive formula (67) with the initial condition (66).

Let $X$ denote the queue waiting time of a tagged class-$D$ packet which arrives at an arbitrary time. Define the partial LST $X^*(s)$ of the queue waiting time of the tagged class-$D$ packet as

$$X^*(s) = \Pr\{\text{served}\} \cdot E\left[e^{-sX}\text{\text{served}\right].$$  

(68)

Consider first the case where the tagged packet arrives when the buffer is full. Then, the tagged packet is lost, not served. Thus, we have

$$\Pr\{\text{served}|N_L + N_D = K + 1\} = 0.$$  

(69)

Consider next the case where the tagged packet arrives when the system is empty, i.e., $N_L = N_D = 0$. Then, the tagged
packet immediately enters its service so that \( X = 0 \). Thus, we have

\[
\Pr \{ \text{served} | N_L = N_D = 0 \} \cdot \mathcal{E} \left[ e^{-sX} | \text{served} \right] = 1. 
\] (70)

Consider now the case where the tagged packet arrives when the system is not empty and the buffer is not full, i.e., \( 1 \leq N_L + N_D \leq K \). Then, the tagged packet joins the buffer immediately and observes the remaining service time \( S'_{\xi} \) of a class-\( \xi \) packet in service. Notice that the number of packets in the buffer just after the tagged packet joins the buffer is \( N_L + N_D \) because the tagged packet occupies one slot in the buffer. Notice also that, given that the tagged packet observes \( S'_{\xi} \), if the number of class-\( L \) packets which arrive during the remaining service time \( S'_{\xi} \) is more than \( K - N_L - N_D \), i.e., \( A_L (S') > K - N_L - N_D \), then the tagged packet will be pushed out during \( S'_{\xi} \). Thus, we have

\[
\Pr \{ \text{served} | 1 \leq N_L + N_D \leq K, A_L (S') > K - N_L - N_D \} = 0. 
\] (71)

If \( A_L (S') \leq K - N_L - N_D \), then the tagged packet will wait the remaining service time \( S' \) until the first embedded point after the tagged packet’s arrival. After the embedded point, the remaining waiting time of the tagged packet after the embedded point is \( X^d (N_L, N_D, t) \) where \( N_d^d \) is the number of packets in the buffer and \( N_D^d \) is the number of packets in the buffer which arrive earlier than the tagged packet at the embedded point. Furthermore, given \( A_L (S') \leq K - N_L - N_D \), we have

\[
\left( N_L^d, N_D^d \right) = \begin{cases} (N_L + A_L (S'), N_D - 1), & \text{if } \xi = D, \\ (N_L + A_L (S'), N_D - 1), & \text{if } \xi = L, \end{cases} 
\] (72)

because the packet in service on arrival of the tagged customer departs from the system at the embedded point. Thus, for \( i \geq 1, j \geq 0, i + j \leq K, 0 \leq n \leq K - i - j \), we have

\[
\Pr \{ \text{served} \} = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \sum_{n=0}^{\min(K-i-j, N_D)} q_{(i,j,D,n)}^*(s) \cdot \mathcal{E} \left[ X_{(i+n+1,j)}^d | \text{served} \right] 
\]

and for \( i \geq 0, j \geq 1, i + j \leq K, 0 \leq n \leq K - i - j \), we have

\[
\Pr \{ \text{served} \} = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \sum_{n=0}^{\min(K-i-j, N_D)} q_{(i,j,D,n)}^*(s) \cdot \mathcal{E} \left[ X_{(i+n,j+1)}^d | \text{served} \right] 
\]

Therefore, conditioning on \( N_L, N_D, \xi \), and \( A_L (S') \) results in

\[
X^* (s) = Q_{(0,0)} + \sum_{i=0}^{K-1} \sum_{j=0}^{K-1-j} \sum_{n=0}^{\min(K-i-j, N_D)} q_{(i,j,D,n)}^* (s) \cdot \mathcal{E} \left[ X_{(i+n+1,j)}^d | \text{served} \right] 
\]

where

\[
q_{(i,j,h,n)}^* (s) = \Pr \{ N_L = i, N_D = j, \xi = h, A_L (S') = n \} \cdot \mathcal{E} \left[ e^{-sX} | \text{served} \right] \]
for \( i \geq 1(\xi = L), \ j \geq 1(\xi = D), \ i + j \leq K + 1, \ n \geq 0, \ h \in \{L, D\}. \)

Thus, the LST of the queue waiting times of a class-\( D \) packet which is finally served is given by

\[
E[e^{-sX}]_{\text{served}} = \frac{X^*_\xi(s)}{X^*_\xi(0)}.
\]

(77)

Therefore, we finally have

\[
W^Q_D = -\frac{X^*_\xi(0)}{X^*_\xi(0)},
\]

(78)

\[
W_D = W^Q_D + E[S_D].
\]

7. Numerical Examples

In this section, we explore the impact of the buffer size \( K \), the arrival rates, and the service-time distributions on the performance measures with numerical examples.

We first examine the impact of the buffer size \( K \) on the performance measures with the following numerical example: The arrival rates \( \lambda_L \) and \( \lambda_D \) are set to 1, and the mean service times \( E[S_L] \) and \( E[S_D] \) are set to 1/3. To see the effect of the service-time distributions of both classes, we use three distributions with the same mean of 1/3 for the service-time distributions: an exponential distribution with rate 3, an Erlang distribution with rate 6 and shape parameter 2, and a hyperexponential distribution with rates 6 and 3/2 and
weights $2/3$ and $1/3$. The coefficients of variation (CVs) of the Erlang, exponential, and hyperexponential distributions are $1/\sqrt{2}$, 1, and $\sqrt{2}$, respectively. To explore the effect of $K$, the buffer size $K$ varies from 1 to 7.

Figure 1 shows the blocking probabilities $P^b_L$ and $P^b_D$, the push-out probability $P_{D}^{po}$ of class-D packets, and the mean sojourn times $W_L$ and $W_D$ as a function of the buffer size $K$. As seen in Figure 1, the blocking probabilities $P^b_L$ (top-left) and $P^b_D$ (top-center) and the push-out probability $P_{D}^{po}$ (top-right) all decrease as $K$ increases for all the combinations of the service-time distributions. (The labels in the legend are the distribution names of $S_L$ (before the hyphen) and $S_D$ (after the hyphen).) However, the mean sojourn times $W_L$ (bottom-left) and $W_D$ (bottom-right) increase as the buffer size $K$ increases. This is because, for a smaller value of $K$, arrivals are more prone to loss, and a long queue cannot build. Additionally, the smaller the CVs of $S_L$ and $S_D$, the smaller the blocking/push-out probabilities and the mean sojourn times. This is because large CVs of $S_L$ and $S_D$ lead to a high level of the buffer occupancy, which, in turn, results in high blocking and push-out probabilities and mean waiting times. In addition, we can observe higher blocking probabilities and lower mean sojourn times of class-$D$ packets than those of class-$L$ packets because we offer space priority to class-$L$ packets but time priority to class-$D$ packets.

We now examine the impact of the arrival rate $\lambda_L$ on the performance measures with the same numerical example as for the buffer size $K$, except that, to see the effect of $\lambda_L$, $\lambda_L$ varies from 0.1 to 3.6, and the buffer size $K$ is fixed to be 3. As seen in Figure 2, the blocking probabilities $P^b_L$ and $P^b_D$ and the push-out probability $P_{D}^{po}$, as well as the mean sojourn time $W_L$ of class-$L$ packets, increase as the arrival rate $\lambda_L$ increases. (For brevity of presentation, the exponential cases are omitted in Figure 2.) This is because a large arrival rate $\lambda_L$
leads to a high level of the buffer occupancy by class-L packets (see the top-left panel of Figure 3), which, in turn, results in high levels of the blocking probabilities of both classes and the mean sojourn time $W_L$. However, the mean sojourn time $W_D$ of class-D packets does not follow this pattern (see the bottom-right panel of Figure 2). $W_D$ initially increases and then decreases as $\lambda_L$ increases, except when $S_L$ has the Erlang distribution and $S_D$ the hyperexponential distribution. (For the Erlang-Hyper combination, $W_D$ decreases monotonically.) This is because there are positive and negative effects of $\lambda_L$ on $W_D$. As $\lambda_L$ increases, the probability that the server is busy for class-L packets also increases (see the bottom-left panel of Figure 3 where $\rho_L = \lambda_L E[S_L]$). This negative effect of increasing $\lambda_L$ makes the waiting time of class-D packets longer. However, this negative effect on $W_D$ is limited because class-L packets in the buffer do not affect $W_D$ due to the time priority of class-D packets. Only class-L packets in service have a negative effect on $W_D$. On the other hand, as $\lambda_L$ increases, the push-out probability of class-D packets rises (see the top-right panel of Figure 2). This effect makes the queue level of class-D packets low (see the top-right panel of Figure 3). For large values of $\lambda_L$, this positive effect of high push-out probabilities on $W_D$ dominates the negative effect of high server utilization. Hence, the mean sojourn time $W_D$ of class-D packets initially increases and then decreases.

In addition, for small values of $\lambda_L$ (e.g., $\lambda_L = 0.1$), the larger the CV of $S_L$, the larger the $W_D$ for the same distribution of $S_D$ (see the bottom-right panel of Figure 2). However, as $\lambda_L$ increases, $W_D$ with a larger CV of $S_L$ rises and then falls more rapidly than that with a smaller CV of $S_L$ for the same distribution of $S_D$. As a result, for large values of $\lambda_L$ (e.g., $\lambda_L = 3.6$), the larger the CV of $S_L$, the smaller the $W_D$. This phenomenon can also be explained by the negative and positive effects of $\lambda_L$ on $W_D$. When $\lambda_L$ is small, the queue level of class-L packets is low and push-out rarely occurs (see the top-right panel of Figure 2). Hence, the negative effect of $\lambda_L$ on $W_D$ dominates its positive effect. The negative effect of $\lambda_L$ on $W_D$ is due to the remaining service time of class-L packets in service. Hence, a larger CV of $S_L$ results in a larger $W_D$. On the other hand, when $\lambda_L$ is large (e.g., $\lambda_L = 3$), the buffer is nearly full of class-L packets due to the space priority of class-L packets, and the server is mostly occupied by class-L packets (see the top-left and bottom-left panels of Figure 3).

**Figure 3:** Various factors over various values of $\lambda_L$. 

![Figure 3](image-url)
Hence, it is highly likely that most class-D packets arrive during the service time of a class-L packet, and there is only one vacancy in the buffer at the beginning of the service of that class-L packet because the buffer is nearly full of class-L packets and the class-L packet enters its service from the buffer then. In this situation, a class-D packet can successfully join and be served mainly when there are no arrivals during the elapsed service time of the class-L packet in service (for joining the buffer) and no class-L arrivals during the remaining service time of the class-L packet in service (for no push-out before entering service). In this case, the mean queue waiting time of a class-D packet is $E[S_L^d | A_L (S_L^d) = A_D (S_L^d) = A_L (S_L^d) = 0]$. The bottom-right panel of Figure 3 shows $E[S_L^d | A_L (S_L^d) = A_D (S_L^d) = A_L (S_L^d) = 0]$ as a function of $\lambda_L$. (Since the distribution of $S_D$ does not affect this quantity, the lines of the different distributions of $S_D$ for the same $S_L$ distribution overlap perfectly.) As seen in the bottom-right panel of Figure 3, for small values of $\lambda_L$, the larger the CV of $S_L$, the larger the $E[S_L^d | A_L (S_L^d) = A_D (S_L^d) = A_L (S_L^d) = 0]$, but for large values of $\lambda_L$, the larger the CV of $S_L$, the smaller the $E[S_L^d | A_L (S_L^d) = A_D (S_L^d) = A_L (S_L^d) = 0]$, which, in turn, results in a larger $W_D$. In other words, when $\lambda_L$ is so large that class-L packets monopolize the buffer, the queue level of class-D packets is nearly zero so that the waiting time of a class-D packet which successfully joins and is served converges to the remaining service time of a class-L packet on its arrival, provided there are no more class-L arrivals during the service time of the class-L packet. In this case, the mean of this remaining service time tends to be smaller for a larger value of CV of $S_L$.

We now examine the impact of the arrival rate $\lambda_D$ on the performance measures with the same numerical example as for the buffer size $K$, except that, to see the effect of $\lambda_D$, $\lambda_D$ varies from 0.1 to 3.6, and the buffer size $K$ is fixed to be 3. As seen in Figure 4, the blocking probabilities $P_L^b$ and $P_D^b$ and
the push-out probability $P_{po}^D$, as well as the mean sojourn times $W_L$ and $W_D$, all increase as the arrival rate $\lambda_D$ increases. (For brevity of presentation, the exponential cases are omitted in Figure 4.) In contrast with the impact of $\lambda_L$ on $W_D$, $W_D$ keeps increasing as $\lambda_D$ increases. This is because, as $\lambda_D$ increases, the queue level of class-$D$ packets increases, which results in a negative effect on $W_D$, but the positive effect of push-out on $W_D$ is limited when $\lambda_L$ is fixed.

We now examine the impact of the mean service time $E[SL]$ on the performance measures with the same numerical example as for the buffer size $K$, except that, to see the effect of $E[SL]$, $E[S_L]$ varies from 0.2 to 2, and the buffer size $K$ is fixed to be 3. As seen in Figure 5, the blocking probabilities and mean sojourn times of both classes increase as the mean service time $E[S_L]$ increases. (For brevity of presentation, the exponential cases are omitted in Figure 5.) However, the push-out probability $P_{po}^D$ of class-$D$ packets initially increases and then decreases as $E[S_L]$ increases. This is because there are positive and negative effects of $E[S_L]$ on $P_{po}^D$. As $E[S_L]$ increases, the mean queue length of class-$L$ packets also increases (see the top-left panel of Figure 6). This negative effect makes the push-out probability of class-$D$ packets high. On the other hand, as $E[S_L]$ increases, the server is nearly always occupied by class-$L$ packets (see the bottom-left panel of Figure 6), and the buffer is nearly full of class-$L$ packets, due to the space priority of class-$L$ packets. In this situation, a class-$D$ packet can successfully join and be pushed out by a class-$L$ arrival mostly when there are no arrivals during the elapsed service time of the class-$L$ packet.
in service (for joining the buffer) and one or more class-L arrivals during the remaining service time of the class-L packet in service (for push-out before the class-D packet enters service). In this case, the push-out probability is $Pr\{A_L(S'_L) = A_D(S'_L) = 0, A_L(S'_L) > 0\}$. As seen in the bottom-right panel of Figure 6, $Pr\{A_L(S'_L) = A_D(S'_L) = 0, A_L(S'_L) > 0\}$ initially rises and then falls, as $E[S_L]$ increases. (Since the distribution of $S_D$ does not affect this quantity, the lines of the different distributions of $S_D$ for the same $S_L$ distribution overlap perfectly.) This is because, as $E[S_L]$ increases, the expected number of class-D arrivals during $S_L$ linear increases, but when class-L packets nearly monopolize the buffer, only the first class-D arrival during $S_L$ has a chance to join and be pushed out, and the other class-D arrivals are simply blocked. Furthermore, as seen in the bottom-right panel of Figure 6, the lines of $Pr\{A_L(S'_L) = A_D(S'_L) = 0, A_L(S'_L) > 0\}$ for two different CVs of $S_L$ cross each other, which explains why the lines of the push-out probabilities $P_{po,D}^L$ of two different CVs of $S_L$ for the same distribution of $S_D$ also cross each other.

We finally examine the impact of the mean service time $E[S_D]$ on the performance measures with the same numerical example as for the buffer size $K$, except that, to see the effect of $E[S_D]$, $E[S_D]$ varies from 0.2 to 2, and the buffer size $K$ is fixed to be 3. As seen in Figure 7, the blocking probabilities $P_{po}^L$ and $P_{po}^D$, the push-out probability $P_{po,D}^L$, and the mean sojourn times $W_L$ and $W_D$ all increase as the mean service time $E[S_D]$ increases. (For brevity of presentation, the exponential cases are omitted in Figure 7.) In contrast with the impact of $E[S_L]$ on $P_{po,D}^L$, as $E[S_D]$ increases, the slope of $P_{po,D}^L$ becomes flat, but does not fall to negative. This is because the effect of $E[S_D]$ is limited by the space priority of class-L packets. While, as $E[S_L]$ increases, the server utilization $\rho_L = \lambda_L E[S_L]$ by class-L packets approaches 1 (see the bottom-left panel of Figure 6), as $E[S_D]$ increases, the server utilization $\rho_D = \lambda_D E[S_D]$ by class-D packets does not approach 1 (see the bottom-right panel of Figure 7).

In summary, the buffer size, the arrival rates, and the mean service times all have significant effects on the performance measures. In general, for higher CVs of the service-time distributions, the system performance measures become worse because the buffer level grows higher. However, this can be not the case when the buffer is nearly monopolized by packets of the space-priority class. As seen in the bottom-right panel of Figure 2 and the top-right panel of Figure 5, the service-time distribution of space-priority packets has different effects on the performance measures, depending on whether the buffer is nearly monopolized by space-priority packets or not.
8. Conclusions

In this paper, we analyzed a finite-buffer M/G/1 queueing model with two priority classes, where delay-sensitive traffic was given time priority and loss-sensitive traffic was given space priority, to meet the diverse QoS requirements of both types of traffic. To do this, we first derived the algorithm for computing the joint queue-length distribution at departure epochs using the embedded Markov chain; then, we obtained the joint distributions of queue lengths and remaining service time at an arbitrary time using the semi-Markov process. Using these distributions, we derived the performance measures, such as loss probabilities, push-out probabilities, and mean waiting times for the proposed queueing models. With several numerical examples, we finally demonstrated the applicability of the proposed analysis and explored the effect of system parameters on the performance measures.

Compared to the previous study [9], which tackled a queueing model having identically and exponentially distributed service times for both types of traffic, this study dealt with a queueing model having differently and generally distributed service times for different types of traffic. As a result, the performance measures could be explored not only as functions of the buffer size $K$, the arrival rates, and the mean service time, but also for different service-time distributions. As seen in Section 6, the service-time distributions with different CVs had different effects on the performance measures, especially when the arrival rate and mean service time of the space-priority packets changed. Therefore, it is expected that we can predict the performance measures of the system more accurately, if the proposed finite-buffer M/G/1 model with time and space priorities is employed, instead of the finite-buffer M/M/1 model, to analyze communication systems accommodating multiple types of traffic with different service-time distributions.
We finally conclude this paper with future research directions. In this study, we assumed that the time-priority discipline was nonpreemptive and the space-priority discipline was the push-out mechanism. Our study can be extended to time-space-priority queueing models with other time- and space-priority disciplines. For example, the space-priority discipline can be the partial buffer-sharing mechanism, and/or the time-priority discipline can be preemptive. If this research is achieved, then we can analyze the effect of different combinations of space- and time-priority disciplines on system performance and could design an optimal priority policy to balance different service requirements of loss- and delay-sensitive types of traffic.

In addition, the proposed method can be extended to the analysis of the corresponding batch-arrival model and the corresponding discrete-time model. For the batch-arrival model, there are two different batch acceptance policies regarding how to accept a batch whose arrival makes the buffer exceed the capacity. While for the partial-acceptance policy, a part of the arriving batch joins the buffer until the buffer becomes full, for the total rejection policy, the whole batch is blocked (see [30], Sec. 5.6). For the partial-acceptance policy, it is expected that the embedded Markov chain and the semi-Markov process have a stochastic structure so similar to those in this study that the proposed method can be applied to the batch-arrival model. However, for the total rejection policy, constituting the embedded Markov chain is very difficult (see [30], Sec. 5.6), so an approach other than the proposed method will be needed to analyze the batch-arrival model. Similarly, the corresponding finite Geo/G/1 model and the corresponding partial-acceptance finite Geo\(^X\)/G/1 model can be tackled with the proposed method because the embedded Markov chain and semi-Markov process have a similar structure.

**Appendix**

**A. Proof of Lemma 1**

To prove Lemma 1, we explore special structures of the partial transition matrices \( P_{ijm} \) whose elements are either zero or a positive value. Let \( I \) denote the set of integer tuples \( \{(i,j,l,m): P_{(i,j),(l,m)} > 0\} \). Then, from (10), (12), and (14), \( I \) is the union of the disjoint sets \( I_1, I_2, \) and \( I_3 \), each of which corresponds to the set of \( (i,j,l,m) \) where \( P_{(i,j),(l,m)} > 0 \) in (10), (12), and (14).

\[
I_1 = \{(i,j,l,m): i \geq 0, j \geq 1, i + j \leq K, l \geq i, m \geq j - 1, l + m \leq K\}
\]

\[
\cup \{(i,j,l,m): i \geq 0, j \geq 1, i + j \leq K, K - j + 2 \leq l \leq K, l + m = K\}
\]

\[
= \{(i,j,l,m): 0 \leq i \leq K - 1, i \leq l \leq K, 1 \leq j \leq \min(K - i, K - l + 1), j - 1 \leq m \leq K - l\}
\]

\[
\cup \{(i,j,l,m): 0 \leq i \leq K - 1, i + 2 \leq l \leq K, K - l + 2 \leq j \leq K - i, m = K - l\},
\]

(A.1)

\[
I_2 = \{(i,j,l,m): 1 \leq i \leq K, j = 0, l \geq i - 1, m \geq 0, l + m \leq K\}
\]

\[
= \{(i,j,l,m): 1 \leq i \leq K, i - 1 \leq l \leq K, j = 0, 0 \leq m \leq K - l\},
\]

\[
I_3 = \{(i,j,l,m): i = 0, j = 0, l \geq 0, m \geq 0, l + m \leq K\}
\]

\[
= \{(i,j,l,m): i = 0, 0 \leq l \leq K, j = 0, 0 \leq m \leq K - l\}.
\]

Therefore, the partial transition matrices \( P_{ijm} \) have the following special structure where \( P_{(i,j),(l,m)} > 0 \).

\[
P_{i,j-1} = \begin{bmatrix}
P_{(i,0),(i-1,0)} & P_{(i,0),(i-1,1)} & \cdots & P_{(i,0),(i-1,K-1)} & P_{(i,0),(i-1,K)} & P_{(i,0),(i-1,K-i+1)} \\
0 & 0 & \cdots & 0 & 0 & \\
: & : & \vdots & : & : & : \\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}.
\]

(A.2)
For $1 \leq i \leq K$, we also have

\[
\begin{bmatrix}
P_{(i,0),(i,0)} & P_{(i,0),(i,1)} & \cdots & P_{(i,0),(i,K-1)} & P_{(i,0),(i,K)} \\
P_{(i,1),(i,0)} & P_{(i,1),(i,1)} & \cdots & P_{(i,1),(i,K-1)} & P_{(i,1),(i,K)} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & P_{(i,K-1),(i,K-1)} & P_{(i,K-1),(i,K)} \\
0 & 0 & \cdots & 0 & P_{(i,K),(i,K)}
\end{bmatrix}
\] \hspace{1cm} (A.3)

For $0 \leq i \leq K-1$,

\[
P_{i,i} = \begin{bmatrix} P_{(i,0),(i,0)} \end{bmatrix}.
\] \hspace{1cm} (A.4)

For $0 \leq i \leq K-1$ and $i+1 \leq l \leq K$, we have

\[
P_{i,l} = \begin{bmatrix} P_{(i,0),(l,0)} \end{bmatrix}.
\] \hspace{1cm} (A.5)

where $P_{i,l}$ and $P_{i,l}$ are the following matrices:

\[
P_{i,l} = \begin{bmatrix}
0 & 0 & \cdots & P_{(i,0),(i+1,0)} \\
0 & 0 & \cdots & P_{(i,0),(i+2,0)} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0
\end{bmatrix}
\] \hspace{1cm} \text{(i+1)(i+2)K-(i+1)K+1)

In addition,

\[
P_{i,l} = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0
\end{bmatrix}
\] \hspace{1cm} \text{(i+1)(i+2)K-(i+1)K+1)

From the structure of $P_{i,l}$, $0 \leq i \leq K$, in (A.3) and (A.4), the states $(i, j)$, $0 \leq j \leq K-i$, can reach each other within the same level $i$. Moreover, from (A.2) and (A.5), $P_{i,-1}, 1 \leq i \leq K$, and $P_{i,i}, 0 \leq i \leq K$, $i+1 \leq i \leq K$, have at least one nonzero element, so the levels $i, 0 \leq i \leq K$, can reach each other. Thus, the EMC is irreducible. Furthermore, from (A.2)–(A.4), $0 < P_{(i,j),(i,j)} < 1$ for $i, j \geq 0, i + j \leq K$ because there is at least one positive element in the row of the state $(i, j)$. Thus, the EMC is aperiodic. Since the EMC is a finite-state Markov chain, the EMC is an irreducible ergodic Markov chain.

**B. Proof of Lemma 2**

We first derive $G_K$. Note that from level $K$, the system visits either level $K-1$ or level $K$ on the first transition. Hence, conditioning on the first transition, $G_K$ is expressed as

\[
G_K = P_{K,K-1} + P_{K,K} G_K.
\] \hspace{1cm} (B.1)

This leads to (16) for $i = K$, where the inverse matrix of $I - P_{K,K}$ exists because $P_{K,K}$ is a $1 \times 1$ substochastic matrix. From the definition of $G_K$, $G_{K,K-1} = G_K$. Hence, (17) holds for $i = K$. In addition, from (A.4), we have

\[
P_{K,K} = \left[ \begin{bmatrix} P_{(K,0),(K,0)} \end{bmatrix} \right],
\] \hspace{1cm} (B.2)

where $P_{(K,0),(K,0)} > 0$. Since $P_{K,K}$ is a substochastic matrix, $P_{K,K} < 1$, which leads to

\[
\begin{aligned}
&\left( I_{1 \times 1} - P_{K,K} \right)^{-1} = (1 - P_{(K,0),(K,0)})^{-1} > 0. \\
&\text{Hence, from (16) and (A.2) for } i = K, \text{ we have}
\end{aligned}
\] \hspace{1cm} (B.3)

\[
G_K = \left( I_{1 \times 1} - P_{K,K} \right)^{-1} P_{K,K-1}
\] \hspace{1cm} (B.4)

Furthermore, since $P$ is a stochastic matrix, from (4) we have

\[
P_{K,K-1} \cdot e_{2 \times 1} + P_{K,K} \cdot e_{1 \times 1} = I_{1 \times 1} \cdot e_{1 \times 1}.
\] \hspace{1cm} (B.5)

This leads to
\( G_k \cdot e_{2x1} = (I_{1x1} - P_{k,k})^{-1} P_{K,k-1} \cdot e_{2x1} \)
\( = (I_{1x1} - P_{k,k})^{-1} (I_{1x1} - P_{k,k}) \cdot e_{1x1} = 1. \)  \( (B.6) \)

Therefore, (18) holds for \( i = K. \)

Suppose now that (16), (17), and (18) hold for all \( i, 2 \leq k + 1 \leq i \leq K. \) Observe that from level \( k, \) the system visits either level \( k - 1 \) or level \( l, k \leq l \leq K, \) on one-step transition. Hence, conditioning on the first transition, \( G_k \) is given by
\[ G_k = P_{k,k-1} + \sum_{l=k}^{K} P_{k,l} G_{l,k-1}. \] \( (B.7) \)

Note also that if the first transition is to level \( l, k \leq l \leq K, \) then from the left-skip-free property for levels, the system visits level \( l', k \leq l' < m, \) before the system goes into level \( k - 1. \) Thus, we have
\[ G_{l,k-1} = G_{l,l} G_k. \] \( (B.8) \)

This implies that if (17) holds for \( k + 1 \leq i \leq K, \) then (17) also holds for \( i = k. \)

On the other hand, from (B.7), we have
\[ \left( I_{(K-k+1)\times(K-k+1)} - \sum_{l=k}^{K} P_{k,l} G_{l,k} - P_{k,k} \right) G_k = P_{k,k-1}. \] \( (B.9) \)

Since we assume that (17) and (18) hold for \( k + 1 \leq i \leq K, \) we have
\[ \sum_{l=k}^{K} P_{k,l} G_{l,k} + P_{k,k} \geq \sum_{l=k}^{K} P_{k,l} G_{l-1} \cdots G_{k+1} > 0, \] \( (B.10) \)

where the last inequality holds because \( G_i > 0 \) for \( k + 1 \leq i \leq K \) and \( P_{k,l} \) has at least one positive element on each row from the special structure in (A.5). Moreover, from (17) and (18) for \( k + 1 \leq i \leq K, \) we have
\[ \left( \sum_{l=k}^{K} P_{k,l} G_{l,k} + P_{k,k} \right) \cdot e_{(K-k+1)\times1} = \sum_{l=k}^{K} P_{k,l} G_{l-1} \cdots G_{k+1} \cdot e_{(K-k+1)\times1} + P_{k,k} \cdot e_{(K-k+1)\times1} = \sum_{l=k}^{K} P_{k,l} \cdot e_{(K-l+1)\times1}. \] \( (B.11) \)

Since \( P \) is a stochastic matrix, from (4), we have
\[ P_{k,k-1} \cdot e_{(K-k+2)\times1} + \sum_{l=k}^{K} P_{k,l} \cdot e_{(K-l+1)\times1} = e_{(K-k+1)\times1}. \] \( (B.12) \)

In addition, since from (A.2) \( P_{k,k-1} \) has at least one positive element, from (B.11) and (B.12) we have
\[ \left( \sum_{l=k}^{K} P_{k,l} G_{l,k} + P_{k,k} \right) \cdot e_{(K-k+1)\times1} = e_{(K-k+1)\times1} - P_{k,k-1} \cdot e_{(K-k+2)\times1} \leq e_{(K-k+1)\times1}. \] \( (B.13) \)

where there exists at least one row that satisfies the strict inequality. Hence, \( \sum_{l=k}^{K} P_{k,l} G_{l,k} + P_{k,k} \) is a substochastic matrix. Therefore, there exists the inverse of \( I - \sum_{l=k}^{K} P_{k,l} G_{l,k} - P_{k,k}, \) and (B.9) leads to (16) for \( i = k. \)

In addition, we have
\[ \left( I_{(K-k+1)\times(K-k+1)} - \sum_{l=k}^{K} P_{k,l} G_{l,k} - P_{k,k} \right)^{-1} = \sum_{n=0}^{\infty} \left( \sum_{l=k}^{K} P_{k,l} G_{l,k} + P_{k,k} \right)^n > 0. \] \( (B.14) \)

Thus, we have
\[ G_k = \left( I_{(K-k+1)\times(K-k+1)} - \sum_{l=k}^{K} P_{k,l} G_{l,k} - P_{k,k} \right)^{-1} P_{k,k-1} > 0, \] \( (B.15) \)

because from (A.2) all the columns of \( P_{k,k-1} \) have one positive element and the others are zero elements. Furthermore, from (B.11) and (B.12), we have
\[ P_{k,k-1} \cdot e_{(K-k+2)\times1} = \left( I_{(K-k+1)\times(K-k+1)} - \sum_{l=k}^{K} P_{k,l} G_{l,k} - P_{k,k} \right)^{-1} \cdot e_{(K-k+1)\times1}. \] \( (B.16) \)

Thus, from (16) for \( i = k, \) we have
\[ G_k \cdot e_{(K-k+2)\times1} = \left( I_{(K-j+1)\times(K-j+1)} - \sum_{l=k}^{K} P_{k,l} G_{l,k} - P_{k,k} \right)^{-1} P_{k,k-1} \cdot e_{(K-k+2)\times1} = e_{(K-k+1)\times1}. \] \( (B.17) \)

Therefore, (18) holds for \( i = k. \)

Finally, by induction, we can prove that (16), (17), and (18) hold for all \( i, 1 \leq i \leq K. \)

### C. Proof of Lemma 3

Observe that, due to the left-skip-free property for levels, there are no transitions from levels \( k + 1, k + 2, \ldots, K \) to levels \( 0, 1, \ldots, k - 1 \) in the original EMC. Thus, any transition probability from level \( i, 0 \leq i \leq k, \) to level \( l, 0 \leq l \leq k - 1, \) in the censored Markov chain \( H(k) \) is the same as the corresponding transition probability in the original EMC. That is,
\[ H_{ij}(k) = P_{ij}, \quad 0 \leq i \leq k, 0 \leq l \leq k - 1. \] \( (C.1) \)

Observe also that any transition from level \( i, 0 \leq i \leq k, \) to levels \( k + 1, k + 2, \ldots, K \) in the original EMC is considered as a transition from level \( i \) to level \( k \) in the censored Markov chain because the system must visit level \( k \) first from levels \( k + 1, k + 2, \ldots, K \) before it visits any of levels \( 0, 1, \ldots, k - 1, \) due to the left-skip-free property for levels. Hence, conditioning on the first transition from level \( i, 0 \leq i \leq k, \) to level \( l, \)
This gives the partial transition probability matrix $H_{i,k}(k)$ from level $i$ to level $k$ in the censored Markov chain as (22). Therefore, the whole transition probability matrix $H(k)$ is given by (21).

We now prove that the censored Markov chain $H(k)$ is irreducible and ergodic. From (22), we have

$$H_{i,k}(k) \geq P_{i,k}, \quad 0 \leq i \leq k \leq K,$$

because $G_{i,k}$ are positive matrices from Lemma 2. Furthermore, $P_{i,i}$, $1 \leq i \leq k$, and $P_{i,i}$, $0 \leq i \leq k$, are nonnegative matrices that have at least one positive element. Thus, from the structure of $H(k)$ all the levels can reach each other. In addition, from the structure of $P_{i,i}$, $0 \leq i \leq k - 1$, all the states at level $i$ can reach each other within the same level, and so do all the states at level $k$ because $H_{i,k}(k) \geq P_{k,k}$. Therefore, the censored Markov chain is also irreducible. Furthermore, from (21), (A.3), and $H_{i,k}(k) \geq P_{k,k}$, we have

$$0 < p_{(i,j),(i,j)} \leq h_{(i,j),(i,j)} < 1.$$

Thus, the censored Markov chain $H(k)$ is aperiodic, so it is ergodic.

### D. Proof of Lemma 4

The censored Markov chain $H(K)$ has the same transition probability matrix as the original EMC because we observe all the levels $l$, $0 \leq l \leq K$. Hence, we have (23) for $k = K$. Furthermore, from (22) we have

$$H_{i,k}(k) = P_{i,k} + P_{i,k+1}G_{k+1,k} + \sum_{l=k+2}^{K} P_{i,l}G_{l,k+1}G_{k+1,k}$$

$$= P_{i,k} + \left(P_{i,k+1} + \sum_{l=k+2}^{K} P_{i,l}G_{l,k+1}\right)G_{k+1}$$

$$= P_{i,k} + H_{i,k}(k+1)G_{k+1},$$

which leads to (23).

In addition, from (18) and (22), we have, for $1 \leq k \leq K$ and $0 \leq j \leq k$,

$$H_{i,k}(k)e_{(K-k+1)\times 1} = P_{i,k}e_{(K-k+1)\times 1} + \sum_{l=k+1}^{K} P_{i,l}G_{l,k}e_{(K-k+1)\times 1}$$

$$= \sum_{l=k}^{K} P_{i,l}e_{(K-l+1)\times 1} = e_{(K-k+1)\times 1} - P_{i,k-1}e_{(K-k+2)\times 1} \leq e_{(K-j+1)\times 1}.$$

### E. Proof of Lemma 1

From (21), $H(K) = P$. From the definition of $\pi(k)$, we also have $\pi = \pi(K)$. Hence, we have

$$\pi = \pi = \pi P = \pi(K)H(K), \quad \pi e = \pi e = 1.$$

Therefore, $\pi(K)$ is the unique stationary probability vector of $H(K)$, and (27) and (28) hold for $k = K$.

Suppose that (27) and (28) hold for a certain $k$, $1 \leq k \leq K$. Then, from (27) for $k$ and the structure of $H(k)$ described in (21), we have

$$\pi_k = \sum_{i=0}^{k-1} \pi_i H_{i,k}(k) = \sum_{i=0}^{k-1} \pi_i H_{i,k}(k) + \pi_k H_{k,k}(k).$$
This leads to
\[ \sum_{i=0}^{k-1} \pi_i H_{ik}(k) = \pi_i (I - H_{ik}(k)). \quad (E.3) \]

We now prove that (27) holds for \( k - 1 \); i.e.,
\[ \pi(k - 1) = \pi(k - 1)H(k - 1). \] Since \( H(k - 1) \) has the structure described in (21), \( \pi(k - 1) = \pi(k - 1)H(k - 1) \) holds if the following equations are satisfied:
\[ \pi_l = \sum_{i=0}^{l-1} \pi_i P_{ij} = 0 \leq l \leq k - 2, \quad (E.4) \]
\[ \pi_{k-1} = \sum_{i=0}^{k-1} \pi_i H_{ik}(k - 1). \quad (E.5) \]

(E.4) holds because \( \pi \) is the stationary probability vector of the original EMC. Furthermore, from (23), (24), and (E.3), we have
\[ \sum_{i=0}^{k-1} \pi_i H_{ik}(k - 1) = \sum_{i=0}^{k-1} \pi_i \left( P_{ik} + H_{ik}(k)G_k \right) \]
\[ = \sum_{i=0}^{k-1} \pi_i P_{ik} + \sum_{i=0}^{k-1} \pi_i H_{ik}(k)G_k \]
\[ = \sum_{i=0}^{k-1} \pi_i P_{ik} + \pi_k (I - H_{ik}(k))G_k \quad (E.6) \]
\[ = \sum_{i=0}^{k-1} \pi_i P_{ik} + \pi_k P_{kk,1} \]
\[ = \pi_{k-1}. \]

The final inequality above holds because \( \pi = \pi P \). Thus, (E.5) holds. Therefore, (27) holds for \( k - 1 \), and (28) also holds for \( k = 1 \) because \( h(k) \) is the unique probability vector that satisfies (25). Finally, by induction (27) and (28) hold for all \( k, 0 \leq k \leq K \).

### F. Computation Complexity of Algorithm 1

We now calculate the computation complexity of Algorithm 1. Let \( CP \) denote the computational complexity of Step 5 in Algorithm 1.

Step 1-2: To obtain \( P_{ij}G_{ij} \) in (16), about \( 2(K - I + 1)(K - i + 1)^2 \) floating-point addition or multiplication operations are required. Hence, computing \( I_{(K-1)^2} - \sum_{i=0}^{K-1} P_{ij}G_{ij} \) requires \((K - i + 1)^3\) plus about \( 2K^2 \sum_{i=0}^{K-1} I = I^2 (K + i + 1)(K - i)\) addition or multiplication operations. Furthermore, if we use the Gauss–Jordan elimination method, then the inversion of the matrix \( I_{(K-1)^2} - \sum_{i=0}^{K-1} P_{ij}G_{ij} - P_{ij} \) requires about \( 2(K - i + 1)^3/3 \) operations. Since these operations are performed for \( 1 \leq i \leq K \), we have

\[ CP_{1-2} = \frac{1}{30} K(K + 1)(6K^3 + 29K^2 + 41K + 9) = O(K^5). \quad (F.1) \]

Step 1-3: Computing \( G_{ij} = G_{ij} \times G_i \) requires about \( 2(K - I + 1)(K - i + 1)(K - i + 2) \) operations. For all \( i \) and \( l, 1 \leq i \leq K, i + 1 \leq l \leq K \), we need to perform the computation. Hence, we have

\[ CP_{1-3} = \sum_{i=1}^{K} \sum_{l=1}^{K} (K - l + 1)(K - i + 1)(K - i + 2) \]
\[ = \frac{1}{10} (K - 1)K(K + 1)(K + 2)(2K + 1) = O(K^5). \quad (F.2) \]

Step 2-3: Computing \( H_{i,k+1} + G_{k+1} \) requires about \( 2(K - i + 1)(K - k)(K - k + 1) \) operations. For all \( i \) and \( k, 0 \leq i \leq k \leq K \), we need to perform the computation. Hence, we have

\[ CP_{2-3} = \sum_{k=0}^{K-1} \sum_{i=0}^{K-1} (K - i + 1)(K - k)(K - k + 1) \]
\[ = \frac{1}{15} K(K + 1)(K + 2)(K + 3)(2K + 2) = O(K^5). \quad (F.3) \]

Step 3-1: If we use the LU decomposition method to obtain the stationary probability vector \( h_0 \), then we have (see [23])

\[ CP_{3-1} = O(K^3). \quad (F.4) \]

Step 3-3: To obtain \( (I - H_{kk}(k))^{-1} \), about \( 2(K - k + 1)^3/3 + (K - k + 1)^2 \) floating-point addition or multiplication operations are required. The summation of \( \sum_{i=0}^{K-1} h_i H_{ik}(k) \) requires \( \sum_{i=0}^{K-1} 2(K - i + 1)(K - k + 1) + (K - k + 1)(K - k + 1) \) operations. In addition, the multiplication of \( \sum_{i=0}^{K-1} h_i H_{ik}(k) \) \( (I - H_{kk}(k))^{-1} \) requires \( 2(K - k + 1)^2 \) operations. Since these operations are performed for \( 1 \leq k \leq K \), we have

\[ CP_{3-3} = \sum_{k=1}^{K} \left[ \frac{2}{3} (K - k + 1)^3 + \sum_{i=0}^{k-1} 2(K - i + 1)(K - k + 1) + (K - i)(K - k + 1) + 2(K - k + 1)^2 \right] \]
\[ = \frac{1}{12} K(K + 1)(K + 4)(5K + 3) = O(K^4). \quad (F.5) \]
Step 4-1: Since we only need to sum all the elements of the vectors $h_k$, $0 \leq k \leq K$, we have

\[ CP_{4-1} = \frac{(K - k + 1)(K - k + 2)}{2} = O(K^2). \]  

(F.6)

Step 4-2: Since we only need to multiply all the elements of the vectors $h_k$, $0 \leq k \leq K$, we have

\[ CP_{4-2} = \frac{(K - k + 1)(K - k + 2)}{2} = O(K^2). \]  

(F.7)

Therefore, the proposed algorithm has a computational complexity of $O(K^3)$.

### Data Availability

The parameter input values used in the numerical study are described in Section 6 of the article.

### Conflicts of Interest

The author declares no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

This research was funded by a 2021 Research Grant from Sangmyung University.

### References


