Research Article

Performance Analysis of Novel Overload Control with Threshold Mechanism

Doo Il Choi¹ and Dae-Eun Lim²

¹Department of Applied Mathematics, Halla University, 28 Halla University-gil, Wonju-si, Gangwon-do 26404, Republic of Korea
²Department of System & Management Engineering, Kangwon National University, 1 Kangwondaehak-gil, Chuncheon-si, Gangwon-do 24341, Republic of Korea

Correspondence should be addressed to Dae-Eun Lim; del@kangwon.ac.kr

Received 30 September 2015; Accepted 30 December 2015

Academic Editor: Jian Guo Zhou

We propose a novel overload control method with hysteresis property; that is, we analyze the $M/G/1/K$ queueing system where the service and arrival rates are varied depending on the queue-length. We use two threshold values: $L_1(\leq L_2)$ and $L_2(\leq K)$. When the queue-length increases by an amount between $L_1$ and $L_2$, we apply one of the following two strategies to reduce the queue-length, either we decrease the mean service time or we decrease the arrival rate. If the queue-length exceeds $L_2$ with one strategy, we apply the other; thus, there are two models that depend on the method that was applied first. We derive the queue-length distribution at departure and at arbitrary epochs using the embedded Markov chain method and the supplementary variable method. We investigate performance measures including the loss probability and mean waiting time using various numerical examples.

1. Introduction

In systems with queues, various overload control mechanisms to alleviate congestion caused by an excessive number of customers have been investigated. Here, we investigate queue-length-dependent policies based on single or multiple threshold value(s) that depend on the system size. For example, consider an oscillating queue of Chydzinski [1], also known as a queue with a hysteretic control method. There are two threshold values: $L_1(\geq 0)$ and $L_2(\geq L_1)$. Initially, the system is idle and the service rate is set to $1/\mu_1$. When the system size exceeds $L_2$, the service rate is changed to $1/\mu_2$, and it continues until the system size reduces to $L_1$. The service rate then returns to $1/\mu_1$, and the process is repeated. It is customary to assume that $1/\mu_1 < 1/\mu_2$ in order to reduce the system size more quickly.

Control mechanisms that are based simply on the system size have been applied to many applications, including manufacturing systems, telecommunications, and computer networks. Li and Zhu [2] and Li et al. [3] mentioned that techniques based on the input rate are usually employed due to high throughput demand. There have been several reports of telecommunication networks, including asynchronous transfer mode (ATM) networks, which support various types of traffic, including voice, data, and video. Important properties of such traffic include time correlation and burstiness, which may lead to network congestion due to unpredictable statistical fluctuations in the traffic streams. Overload control mechanisms can be applied to alleviate this congestion to ensure service quality is maintained; for further details, refer to B. D. Choi and D. I. Choi [4], Choi et al. [5, 6], and the references therein. Banik [7] recently reported the BMAP/G/G/1/N oscillating queueing system that can be used to model semiconductor manufacturing processes. Controlling the arrival rate and service rate has been also considered for potential applications in data centers [8]. The service rate is represented by the frequency of the processors, which was adapted to conserve energy by applying voltage control.

Here, we propose two novel overload control methods, in which both the arrival and service rates are adapted depending upon the system size. A detailed description of our
methods is given in Section 3. Most existing models regulate only the service rate in accordance with the system size, whereas the arrival rate remains constant, or vice versa. In contrast, our overload control method changes the service and arrival rates depending upon the system size. In practice, it is common to apply two or more different overload control mechanisms either simultaneously or sequentially. For example, with congested road networks, one or more lanes can be reversed to improve the flow of traffic. Despite this measure, if the area remains congested, entering the section may be prohibited until the waiting times for cars decrease to a certain level. In semiconductor manufacturing systems, adaptation of the service and arrival rates are often considered together. To decrease the input rate to a particular processing tool, the arriving lots can be rerouted (i.e., sent to another machine). For the example of increasing the service rate in semiconductor manufacturing systems, refer to Banik [7].

The remainder of this paper is organized as follows. In the subsequent section, we review the relevant literature; a description and analysis of the model is given in Section 3; numerical results describing the performance of the model are then given in Section 4; Section 5 provides the conclusions of this paper.

2. Literature Review

Queueing models with queue-length-dependent overload control can be classified into two types depending upon the type of rate that is controllable, that is, the service or the arrival rate. The majority of research has assumed Markovian arrival (i.e., an $M/G/1$-type queueing model), where the service rate is adjusted.

Nishimura and Jiang [9] introduced an $M/G/1$ queueing model with an adjustable service speed. The model had only one threshold; that is, when the number of customers exceeds (is less than) the threshold value, faster (ordinary) service is provided. Dudin [10] extended this model (with a single threshold) with one threshold by considering a batch arrival process. B. D. Choi and D. I. Choi [4] and Choi et al. [5] applied adjustable service rate schemes for telecommunications networks, extending Nishimura and Jiang’s model [9] by considering the arrival process as a Markovian modulated Poisson process (MMPP) (i.e., an MMPP/G/1 queue) or a finite $M/G/1/K$ queueing model, respectively. Later, more complicated arrival processes are considered; Choi [11] analyzed a model with Markovian arrival process (MAP) and an $M^{X}$/G/1 queue with multiple threshold values (or service modes) is analyzed by Choi et al. [12]. Chydzinski [1, 13] and Zhernovyi and Zhernovyi [14, 15] analyzed oscillating queues (or queues with a hysteresis strategy). Oscillating queues with two thresholds differ from queues with multiple thresholds. Suppose a system has two threshold values $L_1$ ($\geq 0$) and $L_2$ ($\geq L_1$) and that the number of customers is between $L_1$ and $L_2$. When hysteretic control is applied, the service modes can differ even for the same number of customers, when the number of customers is between two thresholds (refer to the example in Section 1). On the other hand, in queues with multiple thresholds and without a hysteresis control, the same service mode is applied for the same system size. Jain [16] analyzed a heterogeneous $M/M/r$ queueing model where the number of working servers varied depending on the number of customers. On aggregate, the number of working servers can be considered as the service speed. Recently, Banik [7] extended the oscillating queueing model by considering a batch Markovian arrival process (BMAP). Gupta et al. [17] considered a DMAP/G/1/K queue with queue-length-dependent service rates. There are few examples of $G/1$ queueing systems with adjustable service rates, outside of the work of Chydzinski [1].

There have been fewer reports of systems with adjustable arrival rates than of systems with adjustable service rates. Gong et al. [18] introduced the $M/G/1$ queue with a queue-length-dependent arrival rate. Li and Zhu [2] considered vacations and Li et al. [3] considered the setup time. Chydzinski [19] analyzed an $M/G/1$ queueing system with hysteretic control based on the arrival rates using potential method. More recently, Choi et al. [6] extended these models by considering an MMPP as an arrival process and multiple thresholds.

Although there exist models that can adjust the arrival and service rates simultaneously, they are mostly based on Markov decision process (MDP) and the research interests they address differ from those addressed by our model. Thus, we claim that our model differs significantly from existing queueing models because both the arrival and service rates are controllable. However, the derivation is similar to the models of B. D. Choi and D. I. Choi [4] and Choi et al. [6].

3. Analysis

In this section, we describe the model and then analyze it. We derive the queue-length distribution at the departure epoch using the embedded Markov chain method. The queue-length distribution at an arbitrary time is then derived using the supplementary variable method.

3.1. Model Description. We consider an $M/G/1/K$ queueing system with the overload control policies “Arrival First (AF)” and “Service First (SF).” The queueing system has a single server and a buffer with finite capacity $K$, including the customer in service. Customers arrive to the system according to a Poisson process and are served on a first-come-first-served (FCFS) basis. Customers that arrive when the buffer is full are blocked and hence lost. The arrival and service rates are dependent upon the queue-length. We place two thresholds $L_1$ and $L_2$ on the buffer, where $L_1 \leq L_2$. Suppose that the queue-length at the service initiation epoch of a customer is less than the threshold $L_1$. The arrival of customers follows a Poisson process with a rate $\lambda_1$, and the service time of customers is denoted by a random variable $S_j$, which has the distribution function $G_j$. When the queue-length is equal to or greater than $L_1$ at the service initiation epoch, either the arrival rate or the service time distribution changes depending upon the policy that is employed. With the AF policy, the arrival rate for a Poisson process is changed
to $\lambda_2 (\leq \lambda_1)$, while the service time distribution remains unchanged. With the SF policy, the service time of customers is changed and the arrival rate remains constant. The changed service time is denoted by a random variable $S_2$ with the distribution function $G_2$. Finally, assume that the queue-length at the service initiation epoch is equal to or greater than $L_2 (< K)$. In this case, the arrival of customers follows a Poisson process with a rate $\lambda_2$, and the service time of customers is $S_2$. We define $\mu_i$ as the mean and $G'_i(s)$ as the Laplace transform of $S_i$ ($i = 1, 2$).

In this paper, we analyze a system with the AF policy first; a system with the SF policy is then discussed, as the derivations are similar.

### 3.2. Embedded Markov Chain

Let $\tau_n$ ($n \geq 1$) be the epoch of successive departures with $\tau_0 = 0$. The number of customers in a queue (excluding the customer in service) at time $\tau_n$ (immediately after the $n$th departure) is denoted by $N_n$. The process $\{N_n, n \geq 0\}$ forms a Markov chain with a finite state space $\{0, 1, \ldots, K - 1\}$. To derive the stationary probability distribution of the Markov chain $\{N_n, n \geq 0\}$, we define the probabilities $x_k$ as

$$x_k = \lim_{n \to \infty} \Pr \{N_n = k\}, \quad 0 \leq k \leq K - 1,$$

where $x = (x_0, x_1, \ldots, x_{K-1})$.

We introduce the following probabilities to obtain $x_k$.

First, $a'_r$ denotes the probability that there are $n(\geq 0)$ arrivals during the service time $S_r$; that is,

$$a'_r = \int_0^\infty \frac{e^{-\lambda_r x}}{n!} dG_r(x), \quad r = 1, 2.$$  (2)

Let $b_n$ ($n \geq 0$) be the probability that there are $n$ arrivals during the service time $S_1$, which is defined as follows (when the employed policy is AF; for the SF policy, refer to Remark 1):

$$b_n = \int_0^\infty \frac{e^{-\lambda_1 x}}{n!} dG_1(x).$$  (3)

We therefore have

$$\bar{a}'_n = \sum_{k=0}^{\infty} a'_k, \quad r = 1, 2,$$

$$\bar{b}_n = \sum_{k=0}^{\infty} b_k.$$  (4)

For the Markov chain $\{N_n, n \geq 0\}$, we have the one-step transition probability matrix $P$ as follows:

$$P = \begin{pmatrix}
    a_0^1 & a_0^2 & \cdots & a_0^{K-1} & a_1^1 & a_1^2 & \cdots & a_1^{K-1} & a_2^1 & a_2^2 & \cdots & a_2^{K-1} & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    a_0^1 & a_0^2 & \cdots & a_0^{K-1} \cdot & a_1^1 & a_1^2 & \cdots & a_1^{K-1} & a_2^1 & a_2^2 & \cdots & a_2^{K-1} & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    0 & a_0^1 & \cdots & a_0^{K-1} \cdot & a_1^1 & a_1^2 & \cdots & a_1^{K-1} & a_2^1 & a_2^2 & \cdots & a_2^{K-1} & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & a_1^1 & a_2^1 & \cdots & a_2^{K-1} \cdot & a_3^1 & a_3^2 & \cdots & a_3^{K-1} & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    0 & 0 & 0 & \cdots & b_0 & b_1 & b_2 & \cdots & b_{L-1} & b_{L-1} & \cdots & b_{L+1} & \bar{b}_K & \bar{b}_K & \cdots & \bar{b}_K \\
    0 & 0 & 0 & \cdots & 0 & b_0 & b_1 & \cdots & b_{L-1} & b_{L-1} & \cdots & b_{L+1} & \bar{b}_K & \bar{b}_K & \cdots & \bar{b}_K \\
    \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & 0 & 0 & \cdots & b_1 & b_2 & \cdots & b_{K-1} & b_{K-1} & \cdots & \bar{b}_K \\
    0 & 0 & 0 & \cdots & 0 & 0 & \cdots & b_1 & b_2 & \cdots & b_{K-1} & b_{K-1} & \cdots & \bar{b}_K \\
    \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & a_1^1 & a_1^2 & \cdots & a_1^{K-1} & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & a_1^1 & a_1^2 & \cdots & a_1^{K-1} & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & \bar{a}_K^1 & \bar{a}_K^2 & \cdots & \bar{a}_K^{K-1} \\
    \end{pmatrix}.$$  (5)
Remark 1. The probability \(b_n\) for the SF policy is defined similarly, which is the probability that there are \(n\) arrivals by a Poisson process \(\lambda_1\) during the service time \(S_2\); that is,

\[
b_n = \int_0^{\infty} e^{-x} \frac{x^n}{n!} dG_2(x).
\]

To analyze systems, \(b_n\) in the matrix \(P\) should be computed properly.

The stationary probability vector \(x\) of the Markov chain \(\{N_n, n \geq 0\}\) can be obtained by solving the following equations:

\[
xP = x,\]

\[
xe = 1,
\]

where \(e = (1, 1, \ldots, 1)^T\).

3.3. Queue-Length Distribution at an Arbitrary Time. In this section, we derive the stationary queue-length distribution at an arbitrary time. At an arbitrary time \(t\), let \(N(t)\) be the queue-length including a customer in service, and let \(\xi(t)\) be defined by

\[
\xi(t) = \begin{cases} 
1, & \text{if the service time distribution is } S_1 \text{ at time } t, \\
2, & \text{if the service time distribution is } S_2 \text{ at time } t.
\end{cases}
\]

We define the stationary probabilities as follows:

\[
y_n = \lim_{t \to \infty} \Pr \{N(t) = n\}, \quad 0 \leq n \leq K.
\]

Then, from the key renewal theorem, we have

\[
y_0 = \frac{1}{E} \frac{1}{\lambda_1},
\]

where \(E = x_0(1/\lambda_1 + \mu_1) + \sum_{n=1}^{K-1} x_n \mu_1 + \sum_{n=L_1}^{K-1} x_n \mu_2\) is the mean interdeparture time of customers.

Remark 2. For the SF case, we have

\[
E = x_0 \left( \frac{1}{\lambda_1} + \mu_1 \right) + \sum_{n=1}^{L-1} x_n \mu_1 + \sum_{n=L_1}^{K-1} x_n \mu_2.
\]

The Laplace transform of \(\alpha_{n,r}(x)\) \((r = 1, 2)\) is as follows:

\[
\alpha_{n,r}^*(s) = \int_0^{\infty} e^{-sx} \alpha_{n,r}(x) \, dx.
\]

To calculate \(\alpha_{n,r}^*(s)\), we must find the probability for arrivals during the elapsed service time \(\bar{T}\). Thus, we define the joint probabilities \(\beta_r(n,x) dx\) and \(\beta_b(n,x) dx\) as

\[
\beta_r(n,x) dx = \lim_{t \to \infty} \Pr \{n \text{ arrivals during } \bar{T} \text{ by } \lambda_r, x < \bar{T} \leq x + dx, \xi(t) = r\}, \quad r = 1, 2,
\]

\[
\beta_b(n,x) dx = \lim_{t \to \infty} \Pr \{n \text{ arrivals during } \bar{T} \text{ by } \lambda_2, x < \bar{T} \leq x + dx, \xi(t) = 1\}.
\]

Let \(\beta_r^*(n,s)\) and \(\beta_b^*(n,s)\) be the Laplace transform of \(\beta_r(n,x)\) and \(\beta_b(n,x)\), respectively; that is,

\[
\beta_r^*(n,s) = \int_0^{\infty} e^{-sx} \beta_r(n,x) \, dx, \quad r = 1, 2,
\]

\[
\beta_b^*(n,s) = \int_0^{\infty} e^{-sx} \beta_b(n,x) \, dx.
\]

From the basic property of the semi-Markov process, \(\alpha_{n,r}^*(s)\) must satisfy two relations for \(\alpha_{n,1}^*(s)\) and \(\alpha_{n,2}^*(s)\). First, for \(1 \leq n < L_2\), we have

\[
\alpha_{n,1}^*(s) = \frac{\mu_1}{E} \left( x_0 \beta_1^*(n-1, s) + \sum_{k=1}^{\min(n,L_1-1)} x_k \beta_1^*(n-k, s) \right),
\]

\[
\alpha_{n,2}^*(s) = \frac{\mu_2}{E} \sum_{k=L_1}^{n} x_k \beta_2^*(n-k, s).
\]

Remark 3. For the SF case, the definition of \(\beta_b(n,x) dx\) in (15) is replaced by (19); that is,

\[
\beta_b(n,x) dx = \lim_{t \to \infty} \Pr \{n \text{ arrivals during } \bar{T} \text{ by } \lambda_1, x < \bar{T} \leq x + dx, \xi(t) = 2\}.
\]

In addition, (17) and (18) for \(\alpha_{n,1}^*(s)\) and \(\alpha_{n,2}^*(s)\) are replaced by (20) and (21), respectively; that is, for \(1 \leq n < L_1\), we have

\[
\alpha_{n,1}^*(s) = \frac{\mu_1}{E} \left( x_0 \beta_1^*(n-1, s) + \sum_{k=1}^{\min(n,L_1-1)} x_k \beta_1^*(n-k, s) \right),
\]

\[
\alpha_{n,2}^*(s) = \frac{\mu_2}{E} \sum_{k=L_1}^{n} x_k \beta_2^*(n-k, s).
\]
and for \( L_1 \leq n < K \), we have
\[
\alpha_{n,2}^* (s) = \frac{\mu_2}{E} \left\{ \min\{n, L_i-1\} \sum_{k=L_i}^{n} x_k \beta_k^* (n-k, s) + \sum_{k=L_i}^{\min\{n, L_i-1\}} x_k \beta_k^* (n-k, s) I_{[n>L_i]} \right\},
\]
(21)
\[
\beta_r^* (n, s) = \frac{1}{\mu_r} \left\{ \sum_{i=0}^{n} a_i R_i^r (s) - G_r^* (s) R_n^r (s) \right\},
\]
r = 1, 2, \quad (22)
\[
\beta_2^* (n, s) = \frac{1}{\mu_2} \left\{ \sum_{i=0}^{n} b_i R_i^2 (s) - G_2^* (s) R_n^2 (s) \right\},
\]
where \( R_n^r (s) = (s - \lambda_r)^{-1} \left\{ \lambda_r (\lambda_r - s)^{-1} \right\}^n \).

Remark 4. For the SF case, \( \beta_r^* (n, s) = (1/\mu_2) \sum_{i=0}^{n} b_i R_i^r (s) - G^* (s) R_n^r (s) \).

By substituting \( \beta_r^* (n, s) \) (\( r = 1, 2 \)) into (17) and (18) and then setting \( s = 0 \), we obtain the stationary queue-length probabilities at an arbitrary time for \( 1 \leq n < K \); that is,
\[
y_n = \alpha_{n,1}^* (0) + \alpha_{n,2}^* (0) = \frac{1}{\lambda_1 E} \left\{ x_0 \left( 1 - \sum_{i=0}^{n-1} a_i \right) + \sum_{k=L_i}^{\min\{n, L_i-1\}} x_k \left( 1 - \sum_{i=0}^{n-k} a_i \right) \right\} + \sum_{k=L_i}^{\min\{n, L_i-1\}} x_k \left( 1 - \sum_{i=0}^{n-k} a_i \right) I_{[n>L_i]},
\]
(23)
\[
y_n = \frac{1}{\lambda_1 E} \left\{ x_0 \left( 1 - \sum_{i=0}^{n-1} a_i \right) + \sum_{k=L_i}^{\min\{n, L_i-1\}} x_k \left( 1 - \sum_{i=0}^{n-k} a_i \right) \right\} + \sum_{k=L_i}^{\min\{n, L_i-1\}} x_k \left( 1 - \sum_{i=0}^{n-k} a_i \right) I_{[n>L_i]} + \sum_{k=L_i}^{K-1} x_k \left( 1 - \sum_{i=0}^{n-k} a_i \right) I_{[n>K]},
\]
(24)

Remark 5. For the SF case, \( y_n \) (\( 1 \leq n < K \)) in (23) is defined as follows:
\[
\min\{n, L_i-1\} \sum_{k=L_i}^{n} x_k \beta_k^* (n-k, s) I_{[n=L_i]},
\]
(25)

3.4. Performance Measures. We obtain three performance measures using the stationary queue-length distribution \( \{y_n, n \geq 0 \} \) in (23) or (25). First, the loss probability \( P_{\text{loss}} \) (or blocking probability) is defined as
\[
P_{\text{loss}} = y_k.
\]
(26)
Second, the mean queue-length (excluding the customer in service) is
\[
L_q = \sum_{i=1}^{K} y_i.
\]
(27)
Finally, from Little’s law, we obtain the mean waiting time in the system as follows:
\[
W_q = \frac{L_q}{\lambda^* (1 - P_{\text{loss}})},
\]
(28)
where
\[
\lambda^* = \sum_{k=0}^{L_i-1} x_k \lambda_1 + \sum_{k=L_i}^{K-1} x_k \lambda_2.
\]
(29)

Remark 6. For the SF case, we have
\[
\lambda^* = \sum_{k=0}^{L_i-1} x_k \lambda_1 + \sum_{k=L_i}^{K-1} x_k \lambda_2.
\]
(30)

4. Numerical Examples

In this section, we present numerical examples on the performance of our overload control method. We obtained various results, changing sets of the parameters \( \lambda_i, \mu_i, L_i, \) and \( K \), where \( i = 1, 2 \). The input rate, the mean service time, and \( K \) are assumed to, respectively, have a value listed in Table 1. Note that \( L_1 \) and \( L_2 \) are depending on the relation \( 0 \leq L_1 < L_2 < K \). We consider three service time distributions: the exponential and the Erlang distributions with a parameter of 2 or 9. It is assumed that the service time distribution itself is also changeable; for example, for the SF policy, the service time distribution when the system size is less than \( L_1 \) may differ from when the system size is greater than \( L_1 \). The Erlang distribution with a parameter of 2 and exponential distribution are denoted by “ERL2,” “ERL9,” and “EXP,” respectively. Suppose that the service time distribution before adjusting the mean service time is the Erlang distribution with a parameter 9, and the Erlang distribution with a parameter 2 is applied after the adjustment. It is then denoted by “ERL9-ERL2.”
However, the AF policy is not always superior considering the mean waiting time in the queue. Table 2 time distribution is depicted in Figure 1. It is assumed that queue with the AF policy and the same queue with the SF policy have the same threshold. In Figure 1, as expected, the mean system size gets larger as \( L_1 \) and \( L_2 \) values increase. It is also clear that the distribution with smaller coefficient of variation (CV) has the smaller mean system size for the identical policy; for example, in Figure 1(b), the CV values of the exponential distribution and the Erlang distribution with a parameter \( K = 30 \) are 1 and 1/3, respectively. Figure 2 also shows the mean system size when \( K = 50 \), but graphs are represented in a different way. In this case, we assume that \( \lambda_1 = 7, \lambda_2 = 3, \mu_1 = 0.25, \mu_2 = 0.125 \), and \( K = 30 \). Cases are the combinations of distributions and the policies. Note that the mean system size when \( L_1 = L_2 = 0 \) corresponds to that of a standard M/G/1/K queue (without any policies). When \( L_1 = L_2 > 0 \), it corresponds to an M/G/1/K queue with a single threshold. In Figure 1, as expected, the mean system size gets larger as \( L_1 \) and \( L_2 \) values increase. It is also clear that the distribution with smaller coefficient of variation (CV) has the smaller mean system size for the identical policy; for example, in Figure 1(b), the CV values of the exponential distribution and the Erlang distribution with a parameter \( K = 30 \) are 1 and 1/3, respectively. Figure 2 also shows the mean system size when \( K = 50 \), but graphs are represented in a different way. In this case, we assume that \( \lambda_1 = 7, \lambda_2 = 3, \mu_1 = 0.25, \mu_2 = 0.125 \), and \( K = 30 \).

In Figure 3, the blocking probabilities over the policies (the AF or SF and combinations of \( L_1 \) and \( L_2 \)) and the service time distributions are illustrated. In this example, we assume that \( \lambda_1 = 9, \lambda_2 = 5, \mu_1 = 0.25, \mu_2 = 0.167 \), and \( K = 10 \).

Based on Figures 1, 2, and 3, it seems that the AF policy is more efficient in reducing \( L_q \) and \( P_{\text{loss}} \). That is, because the effective arrival rate (\( \lambda^* \)) is smaller than that of the SF policy, referring to (29) and (30), it can then be concluded that \( \lambda^* \) of the AF policy is smaller given that \( \lambda_1 \) is reasonably larger than \( \lambda_2 \). However, the AF policy is not always superior considering the mean waiting time in the queue. Table 2
Table 2: Performance measures of an $M/M/1/10$ queue with the AF and SF policy.

<table>
<thead>
<tr>
<th>$(L_1, L_2)$</th>
<th>AF</th>
<th>$L_q$</th>
<th>SF</th>
<th>AF</th>
<th>$W_q$</th>
<th>SF</th>
<th>AF</th>
<th>$P_{loss}$</th>
<th>SF</th>
<th>AF</th>
<th>SF</th>
<th>$\lambda^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>3.3058</td>
<td>3.3058</td>
<td>0.6827</td>
<td>0.6827</td>
<td>0.0315</td>
<td>0.0315</td>
<td>5.0000</td>
<td>5.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0, 2)</td>
<td>4.0239</td>
<td>4.7729</td>
<td>0.8403</td>
<td>0.8705</td>
<td>0.0422</td>
<td>0.0559</td>
<td>5.0000</td>
<td>5.8078</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0, 4)</td>
<td>4.7822</td>
<td>5.7890</td>
<td>1.0143</td>
<td>1.0158</td>
<td>0.0570</td>
<td>0.0827</td>
<td>5.0000</td>
<td>6.2127</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0, 6)</td>
<td>5.6154</td>
<td>6.7887</td>
<td>1.2253</td>
<td>1.1806</td>
<td>0.0834</td>
<td>0.1294</td>
<td>5.0000</td>
<td>6.6050</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0, 8)</td>
<td>6.4002</td>
<td>7.6390</td>
<td>1.4730</td>
<td>1.3330</td>
<td>0.1310</td>
<td>0.2121</td>
<td>5.0000</td>
<td>7.2734</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2, 2)</td>
<td>5.2464</td>
<td>5.2464</td>
<td>1.0183</td>
<td>1.0183</td>
<td>0.0708</td>
<td>0.0708</td>
<td>5.5452</td>
<td>5.5452</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2, 4)</td>
<td>5.5374</td>
<td>5.9766</td>
<td>1.1994</td>
<td>1.0800</td>
<td>0.0746</td>
<td>0.0889</td>
<td>5.3456</td>
<td>6.0742</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2, 6)</td>
<td>6.1719</td>
<td>6.8886</td>
<td>1.3071</td>
<td>1.2147</td>
<td>0.0972</td>
<td>0.1328</td>
<td>5.2299</td>
<td>6.5395</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2, 8)</td>
<td>6.8586</td>
<td>7.7052</td>
<td>1.5512</td>
<td>1.3545</td>
<td>0.1449</td>
<td>0.2148</td>
<td>5.1709</td>
<td>7.2444</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4, 2)</td>
<td>6.4017</td>
<td>6.4017</td>
<td>1.2475</td>
<td>1.2475</td>
<td>0.1082</td>
<td>0.1082</td>
<td>5.7543</td>
<td>5.7543</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4, 4)</td>
<td>6.7383</td>
<td>7.0971</td>
<td>1.3920</td>
<td>1.2970</td>
<td>0.1191</td>
<td>0.1423</td>
<td>5.4955</td>
<td>6.3803</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4, 6)</td>
<td>7.3040</td>
<td>7.8399</td>
<td>1.6316</td>
<td>1.4040</td>
<td>0.1645</td>
<td>0.2216</td>
<td>5.3579</td>
<td>7.1742</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4, 8)</td>
<td>7.9821</td>
<td>8.7482</td>
<td>1.5095</td>
<td>1.5095</td>
<td>0.1743</td>
<td>0.1743</td>
<td>6.0185</td>
<td>6.0185</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(6, 8)</td>
<td>7.8479</td>
<td>8.0775</td>
<td>1.7289</td>
<td>1.5189</td>
<td>0.2048</td>
<td>0.2409</td>
<td>5.7086</td>
<td>7.0057</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(8, 8)</td>
<td>8.4580</td>
<td>8.4580</td>
<td>1.8258</td>
<td>1.8258</td>
<td>0.3005</td>
<td>0.3005</td>
<td>6.6229</td>
<td>6.6229</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Conclusion

We have analyzed the $M/G/1/K$ queueing system with adjustable service and arrival rates depending on the queue-length. We derive the queue-length distributions at departure and at arbitrary epochs using the embedded Markov chain method and the supplementary variable technique. We presented performance measures including the mean system size and the blocking probabilities. The performance of our overload control method was then verified using numerical examples.

We have identified three areas for further work. First, after defining the proper cost structure, it is possible to investigate the trade-off between maintaining a small sized system by changing the arrival and service rates early and the total cost, which includes the holding cost related to the system size and the operating cost related to the speed of service. Second, there has been little investigation of general input models (i.e., $GI/M/1$-type queues) with variable service or arrival rates; it is possible to analyze the $GI/M/1$ queueing models with various state-dependent policies. Lastly, it is possible to
assume a queueing system adjusts the arrival and service rates immediately after the queue size reaches $L_1$ and $L_2$, not at the service completion epochs.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**References**


Submit your manuscripts at http://www.hindawi.com