Comm. Korean Math. Soc. 10 (1995), No. 3, pp. 715–733

TRANSIENT DIFFUSION APPROXIMATION FOR $M/G/m/N$ QUEUE WITH STATE DEPENDENT ARRIVAL RATES

YANG WOO SHIN

ABSTRACT. We present a transient queue size distribution for $M/G/m/N$ queue with state dependent arrival rates, using the diffusion process with piecewise constant diffusion parameters, with state space $[0, N]$ and elementary return boundaries at $x = 0$ and at $x = N$. The model considered here contains not only many basic models but the practical models such as as two-node cyclic queue, repairmen model and overload control in communication system with finite storage buffer. For the accuracy check, we compare the approximation results with the exact and simulation results.

1. Introduction

We consider an $M/G/m/N$ queue with state dependent arrival rates where the interarrival times of customers have exponential distribution with parameter λ_k, $(k = 0, 1, \cdots, N)$, when there are k customers in the system. The service time of each customer has general distribution with mean $1/\mu$ and variance σ^2, which are independent of the interarrival times and the number of customers in the system.

The $M/G/m/N$ queue with state dependent arrival rates generalizes many practical queueing models such as two-node cyclic queue, $M/G/m$ queue with finite source, $M/G/m$ queue with balking and can be applied to overload control in communication system with finite storage buffer.

Time dependent queue size distribution for $M/M/m$ queue with finite source has been obtained by Sharma and Dass [21]. Except for special cases as above, there is no analytic or computationally tractable approach for the time dependent queue size distribution in the $M/G/m/N$

Received March 4, 1995. Revised May 11, 1995.
1991 AMS Subject Classification: 60J60, 60K25, 90B22.
Key words: State dependent arrival rates, transient diffusion approximation.

queue with state dependent arrival rates. There have been various diffusion models developed for ordinary multi-server queues such as $GI/G/m$ queue and $M/G/m$ queue : for stationary distribution, see Kimura [14,16], Halachmi and Franta [12], Yao [24], for transient distribution, see Choi and Shin [4,5]. For more general or extended multi-server queues, many diffusion models also have been developed by Choi and Shin [7], Kimura and Ohsone [15]and Ohsone [18] for group arrivals; by Choi and Shin [6, 8] and Yao and Buzacott [25] for queues with finite waiting spaces; and by Biswas and Sunaga [2], Sivazlian and Wang [19] for queues with state dependency.

The purpose of this paper is to propose a diffusion approximation of time dependent queue size distribution for $M/G/m/N$ queue with state dependent arrival rates. In this paper we use a diffusion process with elementary return boundaries at $x = 0$ and $x = N$ and with piecewise continuous infinitesimal mean and infinitesimal variance. In section 2, we give examples of our model. In section 3, we formulate a diffusion process and derive the solution (in Laplace transform form) of diffusion equation. In section 4, we approximate the time dependent queue size distribution for the $M/G/m/N$ system with state dependent arrival rates and check the accuracy of approximation by numerical comparison with simulation results.

2. Examples

(1) $M/G/m$ queue with finite source.

A typical application of the queue with finite source is the machine repairmen model where the calling population is the machines and an arrival corresponds to a machine breakdown. The repairmen are the servers. We assume m servers are available and that the service times have general distribution with mean $1/\mu$ and variance σ^2. The life time of each machine is exponential with parameter λ and is independent of that of other machines. If we concern the number of breakdowns, this system can be modeled as $M/G/m/N$ queue with state dependent arrival rates $\lambda_k = (N - k)\lambda,\ k = 0, 1, \cdots, N$.

(2) Two-node cyclic queue.

The flow of mechanism of the two-node cyclic queue is such that each job passes through each node in a cyclic manner, repeating the process indefinitely. Two node cyclic model can be applied to the analysis of multiprogramming computer system of control processing unit (CPU) and input output device (IOD). We assume that there are m servers at node 1 and the service times at node 1 are independent and identically distributed (*iid*) and general distribution with mean $1/\mu$ and variance σ^2. We assume that there are c servers at node 2 and the service times at node 2 are *iid* and have exponential distribution with mean $1/\lambda$ and are independent of the service time at node 1. We assume that there are N jobs in the system, each of which passes through the node 1 and node 2 cyclically, repeat indefinitely. We denote this model as $(M/c||G/m)/N$. If we concern the number of customers at node 1, this system can be modeled as $M/G/m/N$ queue with state dependent arrival rates $\lambda_k = \min(N-k, c)\lambda,\ k = 0, 1, \cdots, N$.

(3) $M/G/m$ queue with balking.

We consider $M/G/m$ queue where arrivals become discouraged when the queue is long and may not wish to wait. We assume that if there are k customers in the system then the arrival rates are λ_k and

$$\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1} \geq \lambda_N = 0.$$

This model is the typical $M/G/m/N$ queue with state dependent arrival rates.

(4) Admission control.

Suppose that there are s arrival sources, marked by $\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_s$, respectively, each with arrival rate $\lambda_1^0, \lambda_2^0, \cdots, \lambda_s^0$ and there are n threshold valus $K_1, K_2, \cdots, K_n$ based on which the arrival rate is adjusted. All the arrivals are accepted when the queue size is less than K_1, arrivals from $\tilde{A}_1$ are rejected when the queue size is greater than K_1, arrivals from both $\tilde{A}_1$ and $\tilde{A}_2$ are rejected when the queue size is greater than K_2, etc, and when the queue size is greater than K_n only the arrivals from $\tilde{A}_{n+1}, \tilde{A}_{n+2}, \cdots, \tilde{A}_s$ are accepted, where $K_i < K_{i+1}$. In this case we have $\lambda_{K_i} = \sum_{j=i+1}^{s} \lambda_i^0, i = 0, 1, \cdots, n$, $(K_0 = 0)$ and $\lambda_k = \lambda_{K_i}$ if $K_i \leq k < K_{i+1}$. accepted

3. Diffusion equation and the solution

The essence of the diffusion approximation is to approximate the process with discrete state space by the process with continuous state space which is mathematically tractable. As a process approximation of stochastic process representing the number of customers in the $M/G/m/N$ queue with state dependent arrival rates, we take a diffusion process $\{X(t), t \geq 0\}$ with state space $[0, N]$ and with the elementary return boundaries at $x = 0$ and $x = N$. Preliminary materials on diffusion processes can be found in standard texts such as Karlin and Taylor [13] and Cox and Miller [3]. The behavior of the process $X(t)$ in the interval $(0, N)$ is specified by the diffusion parameters $a(x)$ and $b(x)$, called the infinitesimal variance and infinitesimal mean, respectively. By following the same reason as Kimura [14] and Choi and Shin [4], we choose the diffusion parameters $a(x)$ and $b(x)$ as follows: for $k-1 < x \leq k,\ k = 1, 2, \cdots, N$,

$$\begin{aligned} a(x) &= \lambda_{k-1} + \min(k, m)\mu^3\sigma^2 \\ b(x) &= \lambda_{k-1} - \min(k, m)\mu. \end{aligned} \tag{3.1}$$

Next we specify the boundary behaviors of the process $\{X(t), t \geq 0\}$. The lower and upper boundaries of the state space of $X(t)$ correspond to the zero state and the system capacity N in the queueing system, respectively. Thus the holding times at the lower and upper boundaries of process $\{X(t), t \geq 0\}$ correspond to the idle period of queueing system and the time period during which the system is full, respectively. Since interarrival time has exponential distribution with parameter λ_0 when there is no customers in the system, the idle period has exponential distribution with mean $1/\lambda_0$. Thus we assume that the holding time at the lower boundary of diffusion process $X(t)$ has exponential with parameter λ_0. Even for the ordinary $M/G/m/N$ system, the probability distribution of the time period during which the system is full is not known. Hence the holding time of diffusion process at the upper boundary should be approximated (see (4.3) and (4.4), below). Since the set of Cox distributions is dense in the set of general distributions on $(0, \infty)$ (eg. see Asmusen [1]), first we derive the transient density function of diffusion process when the holding time at the upper boundary has Cox distribution and then by continuity theorem we obtain the probability density function of diffusion process when the holding time has general distribution (see Remark 1, below) The Laplace transform $h^*(s)$ of the density function $h(t)$ of n-stage Cox distribution is $h^*(s) = \sum_{i=1}^{n} d_i(1-c_i)\prod_{j=1}^{i} \frac{\mu_i}{s+\mu_j}$, where $0 \le c_i \le 1$, $(1 \le i \le n)$, $d_1 = 1$ and $d_i = c_1c_2\cdots c_{i-1}$, $(1 < i \le n)$. The mean Λ^{-1} of n-stage Cox distribution is $\Lambda^{-1} = \sum_{i=1}^{n} d_i/\mu_i$. Let $f(x,t|x_0)$ be the probability density function of $X(t)$ given $X(0) = x_0$, i.e.

$$f(x,t|x_0)dx = P(x < X(t) \le x + dx | X(0) = x_0).$$

Throughout this paper we assume that the initial value $X(0) = x_0$ is a nonnegative integer. Then $f(x,t|x_0)$ satisfies the following Fokker-Planck equation (Feller [9], Gelenbe [10])

$$\tag{3.2} \begin{aligned} \frac{\partial f}{\partial t} =& \frac{1}{2}\frac{\partial^2}{\partial x^2}\{a(x)f(x,t|x_0)\} - \frac{\partial}{\partial x}\{b(x)f(x,t|x_0)\} \\ &+ \lambda_0 P(t)\delta(x-1) + \sum_{i=1}^{n} \mu_i(1-c_i)Q_i(t)\delta(x-N+1), \end{aligned}$$

where $\delta(\cdot)$ is Dirac's delta function and $P(t)$ is the probability that the process $X(t)$ is at the lower boundary at time t and $Q_i(t)$ is the probability that the process $X(t)$ is in the ith stage on the upper boundary at time t. The behaviors on the lower and upper boundaries are described by the following first order differential equations

$$\tag{3.3} \frac{dP(t)}{dt} = -\lambda_0 P(t) + \lim_{x\downarrow 0} C_{x,t}f,$$

$$\tag{3.4} \frac{dQ_i(t)}{dt} = \begin{cases} -\mu_i Q_i(t) - \lim_{x\uparrow N} C_{x,t}f & \text{if } i = 1 \\ -\mu_i Q_i(t) + \mu_{i-1}c_{i-1}Q_{i-1}(t) & \text{if } 1 < i \le n \end{cases}$$

where $C_{x,t}f = \frac{1}{2}\frac{\partial}{\partial x}\{a(x)f(x,t|x_0)\} - b(x)f(x,t|x_0)$. Since the boundaries at $x = 0$ and $x = N$ behave as the absorbing boundaries during their holding times, it is natural to assume (Cox and Miller [3])

$$\tag{3.5} \lim_{x\downarrow 0} f(x,t|x_0) = 0,$$

$$\tag{3.6} \lim_{x\uparrow N} f(x,t|x_0) = 0.$$

The initial conditions for $P(t), Q_i(t)$ and $f(x,t|x_0)$ are given by

$$f(x,0|x_0) = \delta(x - x_0), \tag{3.7}$$

$$P(0) = \begin{cases} 1 & \text{if } x_0 = 0 \\ 0 & \text{if } x_0 > 0, \end{cases} \tag{3.8}$$

$$Q_i(0) = \begin{cases} 1 & \text{if } x_0 = N \text{ and } i = 1 \\ 0 & \text{if } x_0 < N \text{ or } i > 1. \end{cases} \tag{3.9}$$

For the convenience of solving partial differential equation, let $a_k = a(k)$, $b_k = b(k)$, $k = 1, 2, \cdots, N$ and $f_k(x,t|x_0)$ be the restriction of $f(x,t|x_0)$ to $k-1 < x \le k, t \ge 0, k = 1, 2, \cdots, N$. For the notational simplicity we set $g_k(t|x_0) = f(k,t|x_0), k = 0, 1, 2, \cdots, N$. Then from (3.5) and (3.6), it is easy to see that $g_0(t|x_0) = g_N(t|x_0) = 0$.

THEOREM 1. *The Laplace transform solution $f^*(x,s|x_0)$ of the partial differential equation (3.2) with conditions (3.3)-(3.9) is given by for $k-1 < x \le k, k = 1, 2, \cdots, N$,*

$$\begin{aligned} f_k^*(x,s|x_0) = {} & \exp\left(\frac{b_k}{a_k}(x-k)\right)\frac{\sinh A_k(x-k+1)}{\sinh A_k} g_k^*(s|x_0) \\ & - \exp\left(\frac{b_k}{a_k}(x-k+1)\right)\frac{\sinh A_k(x-k)}{\sinh A_k} g_{k-1}^*(s|x_0), \end{aligned} \tag{3.10}$$

where $A_k = \sqrt{2a_k s + b_k^2}/a_k, k = 1, 2, \cdots, N$

The unknown terms $P^*(s)$ and $g_k^*(s|x_0)$, $k = 1, 2, \cdots, N-1$, satisfy the following $N \times N$ tridiagonal system

$$T(s)\vec{x}(s) = \vec{v}(s), \tag{3.11}$$

where $\vec{x}(s) = (P^*(s), g_1^*(s|x_0), \cdots, g_{N-1}^*(s|x_0))^t$ and $T(s) = Trid(\vec{p}(s), \vec{q}(s), \vec{r}(s))$ is the $N \times N$ tridiagonal matrix with diagonal vector $\vec{q}(s) = (q_0(s), q_1(s), \cdots, q_{N-1}(s))$, subdiagonal vector $\vec{p}(s) = (p_1(s), p_2(s), \cdots, p_{N-1}(s))$ and superdiagonal vector $\vec{r}(s) = (r_0(s), r_1(s), \cdots, r_{N-2}(s))$.

The components of $\vec{p}(s), \vec{q}(s)$ and $\vec{r}(s)$ are as follows:

$$
\begin{aligned}
&p_1(s) = -\lambda_0,\ p_k(s) = -B_k e^{2\frac{b_k}{a_k}},\ k = 1, 2, \cdots, N-1, \\
&q_0(s) = \lambda_0 + s,\ q_k(s) = C_{k+1},\ k = 1, 2, \cdots, N-2, \\
&q_{N-1}(s) = C_N - h^*(s) B_N e^{2\frac{b_N}{a_N}}, \\
&r_k(s) = -B_{k+1},\ k = 0, 1, \cdots, N-2, \\
&v_k(s) = 1(x_0 = k),\ k = 0, 1, \cdots, N-2, \\
&v_{N-1}(s) = 1(x_0 = N-1) + h^*(s) 1(x_0 = N),
\end{aligned} \tag{3.12}
$$

where

$$
\begin{aligned}
B_k &= \frac{a_k A_k}{2} e^{-\frac{b_k}{a_k}} \frac{1}{\sinh A_k},\ k = 1, 2, \cdots, N \\
C_k &= -\frac{b_{k-1}}{2} + \frac{a_{k-1} A_{k-1}}{2} \coth A_{k-1} + \frac{b_k}{2} + \frac{a_k A_k}{2} \coth A_k, \\
&\quad k = 2, 3, \cdots, N.
\end{aligned}
$$

For the similar derivations of (3.10) and (3.11) with (3.12) see the appendix in Choi and Shin [4] and appendix in Choi and Shin [6, 7], respectively.

REMARK 1. From (3.11) and (3.12), we see that $g_k^*(s|x_0)$ depend on only the Laplace transform $h^*(s)$ of holding time but not on the $Q_i^*(s)$ which is the Laplace transform solution of (3.4). By the continuity theorem of the Laplace transform and the fact that the set of all Cox distributions is dense in the set of probability distributions on $(0, \infty)$ (Assmusen [1]), (3.11) (thus (3.10)) hold true when the holding time at the upper boundary has general distribution.

REMARK 2. An explicit expression of the inverse Laplace transform $g_k(t|s_0)$ of $g_k^*(s|x_0)$ does not seem to be accomplishable. Instead, there are many algorithms available for the numerical inversion of Laplace transforms. For example, there are three standard routines currently available from the ACM library of software algorithms: Algorithm 368 (Stehfest [22]); Algorithm 486(Veillon [23]); Algorithm 619 (Piessens and Huysmana [20]).

Let $P_N(t)$ be the probability that the process $X(t)$ is on the upper boundary $x = N$ at time t. The probability $P_N(t)$ is obtained from the conservation of probability

$$P(t) + \int_0^N f(x,t|x_0)\,dx + P_N(t) = 1.$$

Taking Laplace transform and then applying theorem 1, we have
(3.13)

$$\begin{aligned}
P_N^*(s) =& \frac{1}{s} - \left(P^*(s) + \int_0^N f^*(x,s|x_0)\,dx\right)\\
=& \frac{1}{s} - \Big(\sum_{k=1}^{N} \int_{k-1}^{k} f_k^*(x,s|x_0)dx + P^*(s)\Big)\\
=& \frac{1}{s} - \frac{1}{s}\sum_{k=1}^{N} \frac{a_k A_k}{2}\Big[g_k^*(s|x_0)\Big(\frac{\cosh A_k}{\sinh A_k} - \frac{b_k}{a_k A_k} - \frac{e^{-\frac{b_k}{a_k}}}{\sinh A_k}\Big)\\
&+ g_{k-1}^*(s|x_0)\Big(\frac{\cosh A_k}{\sinh A_k} + \frac{b_k}{a_k A_k} - \frac{e^{\frac{b_k}{a_k}}}{\sinh A_k}\Big)\Big] - P^*(s).
\end{aligned}$$

Now we derive the stationary solution. Let $f(x) = \lim_{t\to\infty} f(x,t|x_0)$, $P = \lim_{t\to\infty} P(t)$ and $P_N = \lim_{t\to\infty} P_N(t)$. To obtain the limiting probability from the Laplace transform solution, we use the final value theorem for Laplace transform; $\lim_{t\to\infty} f(t) = \lim_{s\to 0} sf^*(s)$.

THEOREM 2. *The stationary probability density function $f(x)$ of $X(t)$ is given by;*
(3.14)

$$f(x) = \begin{cases} \frac{e^{\gamma_1 x}-1}{\gamma_1}\frac{2\lambda_0}{a_1}P & \text{if } 0 < x \le 1 \\ q_k e^{\gamma_k(x-k)}P & \text{if } k-1 < x < k, \quad k = 2,3,\cdots,N-1, \\ q_N \frac{1-e^{\gamma_N(x-N)}}{e^{\gamma_N}-1}P & \text{if } N-1 < x < N, \end{cases}$$

where $\gamma_k = \frac{2b_k}{a_k}, k = 1,2,\cdots,N$ *and* $q_1 = \frac{e^{\gamma_1}-1}{\gamma_1}\frac{2\lambda_0}{a_1}$,
$q_k = q_1 \exp(\sum_{j=2}^{k}\gamma_j), k = 2,3,\cdots,N$. *The probability P that the pro-*

cess $X(t)$ is in the lower boundary is given by

(3.15)
$$P = \Big[1 + \frac{e^{\gamma_1} - 1 - \gamma_1}{\gamma_1^2}\frac{2\lambda_0}{a_1} + \sum_{k=2}^{N-1} q_{k-1}\frac{e^{\gamma_k} - 1}{\gamma_k} + \frac{q_N}{e^{\gamma_N} - 1}\big(\frac{b_N}{\Lambda} + \frac{1}{\gamma_N}(e^{-\gamma_N} - 1 + \gamma_N)\big)\Big]^{-1}.$$

The probability P_N that the diffusion process $X(t)$ is on the upper boundary is given by

(3.16)
$$P_N = \begin{cases} \frac{1}{\Lambda}\frac{b_N}{e^{\gamma_N}-1} q_N P & \text{if } b_N \neq 0 \\ \frac{1}{\Lambda}\frac{a_N}{2} q_{N-1} P & \text{if } b_N = 0, \end{cases}$$

where $1/\Lambda$ is the mean holding time at the boundary $x = N$.

For the similar derivations of (3.14)-(3.16), see the appendix of Choi and Shin [6].

REMARK. We consider the case that holding time has general distribution. When we consider the formula (3.16) as a function of holding time distribution at the upper boundary, it depends only on the first moment of holding time. From the fact that if a distribuiton $F(t)$ has finite q-th moment, i.e. $\int_0^\infty x^q dF(x) < \infty$, then there exists a sequence of Cox distributions $\{F_k\}$ such that

$$\int_0^\infty x^p dF_k(x) \to \int_0^\infty x^p dF(x)$$

for $0 \le p \le q$ (e.g. see Asmusen [1]), the formula (3.16) holds for arbitrary distribution of holding time with finite first moment.

4. Diffusion approximation and numerical results

Let $Q(t)$ denote the number of cusomers in the $M/G/m/N$ queue with state dependent arrival rates at time t. Now we derive the approximation $P(k, t|x_0)$ of probability function $\hat{p}(k, t|x_0) = P(Q(t) = k|Q(0) = x_0)$ and its limiting distribution $P(k) = \lim_{t\to\infty} P(k, t|x_0)$. Since we have approximate a discrete state process with a continuous state process, we need to discretize the probability density function $f(x,t|x_0)$. The discretization of probability density function $f(x,t|x_0)$ can be done in serveral different ways (see Gelenbe et al.[11]). Following the method of Choi and Shin [6], we adopt the following:

$$(4.1)\qquad \begin{aligned} &P(0,t) := P(t)/Total \\ &P(n,t) := f(n,t)/Total,\ n = 1,2,\cdots,N-1, \\ &P(N,t) := P_N(t)/Total \end{aligned}$$

where $Total := P(t) + P_N(t) + \sum_{i=1}^{N-1} f(n,t)$.

Using the disretization method (4.1), we have the following diffusion approximation for stationary queue size distribution

$$(4.2)\qquad P(k) = \begin{cases} P/Total & \text{if } k = 0, \\ Pq_k/Total & \text{if } k = 1,2,\cdots,N-1, \\ P_N/Total & \text{if } k = N, \end{cases}$$

where $Total := P + \sum_{i=1}^{N-1} f(k) + P_N$.

From the argument on formulating the diffusion process in section 3, we should determine the distribution of holding times at the boundaries. Since the interarrival time has exponential with parameter λ_0 when there is no customers in the system, the idle period is exponential with parameter λ_0. The distribution of the time period that the system is full, we have the following possible two choices

$$(4.3)\qquad h_{N,1}(t) = m(1-S(t))^{m-1}s(t),$$

(4.4)

$$h_{N,2}(t) = \begin{cases} m(1-S_e(t))^{m-1}s_e(t), & \text{if } m < N \\ (1-S_e(t))^{m-2} & \\ \times((m-1)s_e(t)(1-S(t)) + (1-S_e(t))s(t)), & \text{if } m = N \end{cases}$$

where $S(t)$ is the distribution function of the service time and $S_e(t) = \mu\int_0^t(1-S(\tau))\,d\tau$ is the distribution of the remaining service time and

$s_e(t) = \mu(1 - S(t))$ is its probability density function. (4.4) is obtained from the approximation assumption in Nozaki and Ross [17, p. 828]). For the system with exponential service, the holding time at the upper boundary $x = N$ is exponential with parameter $m\mu$ from the memoryless property of exponential distribution. Note that when $S(t) = 1 - e^{-\mu t}$, i.e. the service time is exponential, $h_{N,1}(t) = h_{N,2}(t) = m\mu e^{-m\mu t}$, which coincides with the exact results. Although (4.4) seems to be more natural than (4.3), many simulation results show that the approximate results using (4.3) is more accurate than those of (4.4). Therefore, we adopt (4.3) as a holding time distribution at the upper boundary in executing the calculations of the approximate formula. The numerical inversion of Laplace transform is derived by using the Algorithm 368 in ACM [22]. The simulation results (denoted by "sim" in tables) are obtained by 30,000 times repetitions.

Table 1 gives the comparision of the diffusion approximation with the analytic result [21] for M/M/3 queue with finite source $N = 9$. Table 2 and table 3 present the numerical comparison of diffusion approximation with simulation for $M/E_2/3$ and $M/H_2/3$ queue with finite source $N = 9$. The numerical results for the two-node cyclic queues $(M/5||M/3)/10$ queue, $(M/5||E_2/3)/10$ queue and $(M/5||H_2/3)/10$ queue are given in tables 4 - 6. In tables $P(n,t)$ denotes the probability that the number of customers in the system at time t is n. In Table 3 and Table 6 we use the service time density function $s(t) = p_1\mu_1 e^{-\mu_1 t} + p_2\mu_2 e^{-\mu_2 t}$, where $p_1 = 0.5(1 + \sqrt{0.2}), p_2 = 1 - p_1$ and $\mu_1 = 2p_1, \mu_2 = 2p_2$. In tables, "diff" denotes the diffusion approximation results. From the numerical results it is concluded that the diffusion approximation gives very accurate results for exponential service time. Our approximation for nonexponential service system is slightly less accurate than for exponential service system.

Table 1. $M/M/3$ Queue with Finite Source $N = 9$
$\lambda = .30,\ \mu = .30,\ x_0 = 4$

P(n,t)	time / method	4.0	8.0	12.0	16.0	20.0	∞
P(0,t)	diff	.003	.002	.002	.002	.002	.002
	exact	.001	.001	.001	.001	.001	.001
P(1,t)	diff	.013	.009	.007	.007	.007	.007
	exact	.010	.007	.006	.006	.005	.005
P(2,t)	diff	.040	.028	.025	.024	.023	.023
	exact	.039	.027	.023	.022	.022	.022
P(3,t)	diff	.083	.060	.054	.052	.052	.052
	exact	.083	.060	.053	.051	.051	.051
P(4,t)	diff	.143	.113	.104	.102	.101	.101
	exact	.147	.114	.105	.102	.102	.101
P(5,t)	diff	.203	.177	.169	.167	.167	.168
	exact	.208	.180	.172	.170	.169	.169
P(6,t)	diff	.224	.223	.222	.222	.221	.221
	exact	.226	.226	.226	.225	.225	.225
P(7,t)	diff	.176	.210	.218	.221	.221	.221
	exact	.176	.212	.222	.224	.225	.225
P(8,t)	diff	.091	.132	.144	.147	.148	.148
	exact	.087	.132	.145	.149	.150	.150
P(9,t)	diff	.023	.047	.055	.057	.057	.058
	exact	.021	.041	.047	.049	.050	.050

Table 2. $M/E_2/3$ Queue with Finite Source $N = 9$
$\lambda = .30,\ \mu = .30,\ x_0 = 0$

P(n,t)	time / method	.5	1.0	3.0	5.0	8.0	10.0	16.0	20.0
P(0,t)	diff	.290	.123	.014	.004	.002	.001	.001	.001
	sim	.264	.077	.004	.001	.000	.000	.000	.000
	c.i	.004	.004	.001	.000	.000	.000	.000	.000
P(1,t)	diff	.354	.227	.046	.018	.008	.006	.005	.004
	sim	.376	.227	.026	.010	.005	.004	.003	.003
	c.i	.006	.007	.002	.001	.001	.001	.001	.001
P(2,t)	diff	.265	.310	.120	.055	.028	.022	.017	.017
	sim	.245	.299	.088	.039	.023	.017	.014	.014
	c.i	.004	.007	.003	.003	.002	.001	.001	.001
P(3,t)	diff	.081	.224	.194	.112	.065	.054	.044	.042
	sim	.089	.234	.173	.098	.061	.051	.044	.043
	c.i	.003	.004	.004	.002	.003	.002	.002	.003
P(4,t)	diff	.009	.095	.237	.183	.129	.112	.096	.094
	sim	.021	.114	.231	.178	.131	.119	.103	.102
	c.i	.002	.004	.005	.005	.005	.005	.003	.003
P(5,t)	diff	.000	.020	.210	.232	.205	.191	.175	.173
	sim	.004	.039	.229	.229	.205	.200	.185	.187
	c.i	.001	.002	.005	.004	.005	.005	.004	.004
P(6,t)	diff	.000	.001	.127	.216	.248	.251	.249	.248
	sim	.000	.009	.155	.220	.246	.245	.256	.252
	c.i	.000	.001	.005	.006	.005	.005	.004	.005
P(7,t)	diff	.000	.000	.046	.133	.205	.226	.245	.248
	sim	.000	.001	.071	.150	.203	.218	.230	.235
	c.i	.000	.000	.003	.004	.005	.005	.005	.005
P(8,t)	diff	.000	.000	.007	.043	.093	.113	.136	.139
	sim	.000	.000	.021	.062	.102	.118	.130	.130
	c.i	.000	.000	.002	.003	.004	.004	.004	.004
P(9,t)	diff	.000	.000	.000	.004	.017	.024	.032	.034
	sim	.000	.000	.002	.012	.023	.027	.034	.035
	c.i	.000	.000	.001	.001	.002	.002	.001	.003

Table 3. $M/H_2/3$ Queue with Finite Source $N = 9$
$\lambda = .30,\ \mu = .30,\ x_0 = 0$

P(n,t)	time method	.5	1.0	3.0	5.0	8.0	10.0	16.0	20.0
P(0,t)	diff	.301	.133	.019	.007	.004	.003	.003	.002
	sim	.292	.105	.009	.003	.002	.001	.001	.001
	c.i	.005	.003	.001	.001	.000	.000	.000	.000
P(1,t)	diff	.337	.223	.055	.026	.015	.012	.010	.010
	sim	.392	.276	.058	.025	.013	.010	.008	.007
	c.i	.006	.006	.003	.002	.001	.001	.001	.001
P(2,t)	diff	.250	.282	.123	.067	.041	.035	.030	.030
	sim	.224	.307	.146	.076	.042	.035	.026	.026
	c.i	.005	.005	.004	.003	.003	.002	.002	.002
P(3,t)	diff	.094	.213	.179	.114	.078	.068	.060	.059
	sim	.073	.193	.206	.131	.087	.074	.059	.057
	c.i	.004	.005	.004	.005	.003	.003	.003	.003
P(4,t)	diff	.018	.108	.207	.165	.128	.116	.106	.105
	sim	.017	.087	.222	.187	.143	.125	.108	.103
	c.i	.002	.002	.004	.005	.004	.004	.004	.003
P(5,t)	diff	.001	.035	.190	.197	.178	.169	.161	.160
	sim	.003	.027	.184	.214	.192	.182	.168	.165
	c.i	.001	.001	.004	.005	.005	.003	.004	.005
P(6,t)	diff	.000	.006	.133	.188	.203	.203	.202	.202
	sim	.000	.005	.114	.186	.213	.216	.215	.210
	c.i	.000	.001	.003	.005	.005	.007	.005	.005
P(7,t)	diff	.000	.000	.068	.138	.181	.192	.201	.202
	sim	.000	.001	.048	.118	.178	.193	.208	.216
	c.i	.000	.000	.002	.004	.003	.004	.004	.004
P(8,t)	diff	.000	.000	.023	.072	.118	.133	.147	.148
	sim	.000	.000	.012	.051	.101	.124	.150	.155
	c.i	.000	.000	.001	.003	.003	.005	.006	.004
P(9,t)	diff	.000	.000	.004	.025	.055	.067	.079	.081
	sim	.000	.000	.001	.009	.028	.039	.057	.060
	c.i	.000	.000	.000	.001	.002	.002	.002	.003

Table 4. Two-node Cyclic Queue $(M/5||M/3)/10$
$\lambda = 0.6,\ \mu = 1.0,\ x_0 = 5$

P(n,t)	time method	.7	1.0	2.0	3.0	5.0	7.0	10.0	20.0
P(0,t)	diff	.003	.009	.023	.029	.033	.034	.035	.035
	sim	.003	.008	.021	.025	.028	.032	.030	.030
	c.i	.001	.001	.002	.002	.002	.002	.002	.001
P(1,t)	diff	.027	.043	.070	.080	.086	.088	.089	.090
	sim	.025	.041	.067	.079	.089	.088	.087	.091
	c.i	.002	.003	.003	.002	.003	.003	.004	.003
P(2,t)	diff	.077	.094	.115	.123	.130	.132	.133	.134
	sim	.071	.091	.113	.127	.130	.132	.135	.137
	c.i	.003	.004	.004	.004	.004	.005	.005	.005
P(3,t)	diff	.125	.127	.127	.128	.131	.133	.133	.134
	sim	.124	.125	.128	.126	.135	.134	.133	.135
	c.i	.004	.003	.005	.004	.004	.005	.004	.003
P(4,t)	diff	.178	.160	.138	.134	.133	.133	.134	.134
	sim	.177	.164	.137	.134	.134	.133	.137	.134
	c.i	.004	.003	.003	.005	.005	.006	.004	.005
P(5,t)	diff	.206	.180	.147	.139	.135	.134	.134	.134
	sim	.214	.182	.151	.138	.131	.136	.137	.138
	c.i	.005	.004	.005	.004	.004	.004	.005	.004
P(6,t)	diff	.197	.179	.151	.142	136	.135	.134	.134
	sim	.200	.181	.155	.145	137	.139	.138	.132
	c.i	.006	.006	.005	.005	004	.005	.005	.004
P(7,t)	diff	.125	.127	.121	.116	110	.108	.107	.107
	sim	.126	.126	.123	.120	114	.108	.108	.109
	c.i	.003	.004	.005	.003	.003	.004	.003	.005
P(8,t)	diff	.049	.061	.071	.070	068	.066	.065	.065
	sim	.047	.062	.071	.072	067	.065	.065	.063
	c.i	.003	.003	.004	.003	.003	.003	.002	.002
P(9,t)	diff	.011	.018	.028	.030	.029	.028	.028	.028
	sim	.010	.017	.027	.027	.029	.027	.026	.026
	c.i	.001	.002	.002	.002	.002	.001	.002	.002
P(10,t)	diff	.001	.003	.007	.008	.008	.008	.008	.008
	sim	.001	.002	.005	.006	.005	.005	.005	.005
	c.i	.001	.001	.001	.001	.001	.001	.001	.001

Table 5. Two-node Cyclic Queue $(M/5||E_l/3)/10$
$\lambda = 0.6,\ \mu = 1.0,\ x_0 = 5$

P(n,t)	time / method	.7	1.0	2.0	3.0	5.0	7.0	10.0	20.0
P(0,t)	diff	.001	.004	.015	.021	.025	.026	.027	.027
	sim	.001	.002	.012	.017	.023	.025	.026	.024
	c.i	.000	.001	.001	.002	.002	.002	.002	.002
P(1,t)	diff	.015	.029	.058	.071	.081	.084	.086	.087
	sim	.004	.012	.040	.057	.070	.074	.079	.078
	c.i	.001	.001	.003	.003	.003	.003	.003	.004
P(2,t)	diff	.060	.081	.112	.125	.136	140	.142	.143
	sim	.022	.044	.080	.097	.117	122	.125	.127
	c.i	.002	.003	.003	.003	.003	003	.004	.003
P(3,t)	diff	.121	.127	.130	.133	.138	141	.142	.143
	sim	.075	.091	.115	.127	.138	140	.143	.143
	c.i	.003	.003	.004	.004	.003	005	.004	.004
P(4,t)	diff	.194	.176	.150	.144	.142	143	.143	.143
	sim	.154	.154	.150	.148	.150	150	.148	.153
	c.i	.004	.004	.005	.003	.003	006	.005	.005
P(5,t)	diff	.236	.205	.165	.153	.146	144	.143	.143
	sim	.240	.211	.175	.165	.160	156	.156	.152
	c.i	.006	.004	.005	.004	.004	.004	.004	.006
P(6,t)	diff	.220	.202	.171	.159	.149	146	.144	.143
	sim	.260	.228	.186	.174	.150	151	.144	.146
	c.i	.004	.006	.004	.004	.004	.005	.004	.005
P(7,t)	diff	.118	.126	.125	.120	.112	108	.106	.105
	sim	.165	.160	.141	.127	.111	.107	.106	.105
	c.i	.004	.004	.004	.004	.003	.004	.005	.004
P(8,t)	diff	.031	.044	.058	.058	.055	.053	.051	.051
	sim	.064	.076	.073	.065	.059	.055	.052	.055
	c.i	.003	.004	.003	.002	.002	.002	.003	.004
P(9,t)	diff	.003	.007	.014	.015	.015	.014	.014	.013
	sim	.014	.021	.025	.021	.019	.017	.017	.016
	c.i	.001	.001	.002	.002	.002	.002	.001	.001
P(10,t)	diff	.000	.000	.001	.002	.002	.002	.002	.002
	sim	.001	.002	.003	.003	.002	.002	.003	.002
	c.i	.001	.001	.001	.001	.001	.001	.001	.001

Table 6. Two-node Cyclic Queue $(M/5||H_2/3)/10$
$\lambda = 0.6,\ x_0 = 5$

P(n,t)	time method	.7	1.0	2.0	3.0	5.0	7.0	10.0	20.0
P(0,t)	diff	.007	.015	.032	.038	.041	.042	.042	.042
	sim	.007	.014	.029	.034	.034	.034	.034	.033
	c.i	.001	.001	.002	.002	.002	.002	.002	.002
P(1,t)	diff	.037	.053	.077	.084	.089	.090	.090	.091
	sim	.041	.059	.091	.098	.099	.095	.098	.099
	c.i	.002	.003	.003	.004	.003	.003	.004	.004
P(2,t)	diff	.087	.100	.115	.120	.124	.126	.126	.126
	sim	.097	.117	.137	.141	.141	.141	.140	.140
	c.i	.003	.003	.005	.004	.004	.004	.004	.005
P(3,t)	diff	.126	.125	.123	.123	.125	.126	.126	.126
	sim	.140	.142	.138	.138	134	.134	.131	.131
	c.i	.003	.005	.004	.004	.004	.003	.004	.004
P(4,t)	diff	.165	.149	.130	.127	.126	.126	.126	.126
	sim	.188	.165	.139	.131	.128	.126	.128	.128
	c.i	.004	.004	.005	.004	.004	.004	.004	.003
P(5,t)	diff	.186	.163	.136	.129	127	.127	.127	.126
	sim	.203	.175	.140	.129	126	.124	.127	.123
	c.i	.004	.005	.004	.004	.004	.004	.004	.004
P(6,t)	diff	.180	.162	.138	.131	.128	.127	.127	.126
	sim	.177	.162	.136	.128	.127	.127	.126	.130
	c.i	.005	.004	.005	.003	.003	.004	.004	.004
P(7,t)	diff	.126	.125	.116	.112	.108	.107	.106	.106
	sim	.102	.103	.105	.104	.104	.108	.106	.104
	c.i	.003	.003	.003	.003	.003	.004	.003	.005
P(8,t)	diff	.062	.071	.077	.077	.074	.073	.073	.073
	sim	.038	.049	.060	.063	.067	.072	.070	.069
	c.i	.003	.003	.003	.002	.003	.003	.003	.004
P(9,t)	diff	.020	.029	.039	.041	.040	.039	.039	.039
	sim	.007	.012	.022	.028	.031	.032	.032	.034
	c.i	.001	.001	.002	.002	.003	.002	.002	.002
P(10,t)	diff	.004	.008	.016	.018	.018	.017	.017	.017
	sim	.001	.002	.004	.007	.008	.007	.009	.007
	c.i	.000	.001	.001	.001	.001	.001	.001	.001

References

1. S. Asmussen, *Applied Probability and Queues*, John Willy & Sons, 1987.
2. S. K. Biswas and T. Sunaga, *Diffusion approximation method for multi server queueing system with balking*, J. of Operations Research Society of Japan **23** (1991), 368-386.
3. D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes*, Chapman and Hall, 1965.
4. B. D. Choi and Y. W. Shin,, *Transient diffusion approximation for $M/G/m$ system,*, J. of Operations Research Society of Japan **34(3)** (1991), 306-328.
5. B. D. Choi and Y. W. Shin, *Approximation analysis of time dependent queue size distribution for $GI/G/m$ system using diffusion process,*, Proceedings of KAIST Mathematics Workshop, Analysis and Geometry, edited by B. D. Choi and J. W. Yim, KAIST, Taejeon, Korea, **6** (1991), 225-242.
6. B. D. Choi and Y. W. Shin, *A diffusion approximation for time dependent queue size distribution for $M/G/m/N$ system*, J. Korean Math. Soc **32** (1995), 211-236.
7. B. D. Choi and Y. W. Shin, *Diffusion approximation of time dependent queue size distribution for $M^X/G^Y/c$ system*, Comm. Korean Math. Soc **10** (1995), 419-438.
8. B. D. Choi, Y. W. Shin and Y. W. Lee, *Diffusion approximation for first overflow time in $GI/G/m$ system*, J. Appl. Math. and Stoch. Anal **8** (1995), 11-28.
9. W. Feller, *Diffusion process in one dimension*, Trans. Amer. Math. Soc **77** (1954), 1-31.
10. E. Gelenbe, *On approximate computer system models*, J. of the Association for Computing Machinery **22** (1975), 261-269.
11. E. Gelenbe, G. Pujolle and J. C. C. Nelson, *Introduction to Queueing Networks*, John Wiley & Sons, New York, 1987.
12. B. Halachmi and W. R. Franta, *A diffusion approximation to the multi-server queue,*, Management Science **24** (1978), 522-529; (erratum, **13**, 1448).
13. S. Karlin and H. M. Taylor, *A Second Course in Stochastic Processes*, Academic Press, 1981.
14. T. Kimura, *Diffusion approximation for $M/G/m$ queue*, Oprerations Research **31** (1983), 304-321.
15. T. Kimura and T. Ohsone, *A diffusion approximation for $M/G/m$ queue with group arrivals*, Management Science **30** (1984), 381-388.
16. T. Kimura, *An $M/M/s$-based approximation for multi-server queues*, Symposium on performance models for infromation communication networks, edited by Hashita (1995), Univ. of Tsukuba, 420-431.
17. S. A. Nozaki and S. M. Ross, *Approximations in finite-capacity multi-server queues with Poisson arrivals*, J. Appl. Prob. **15**, (1978), 826-834.
18. T. Ohsone, *Diffusion approximation for a $GI/G/m$ queue with group arrivals*, J. of Operations Research Society of Japan **27** (1984), 78-96.
19. B. D. Sivazlian and K. H. Wang, *Diffusion approximation to the $G/G/R$ machine repair problem with warm standby spares*, Naval Research Logistics **37** (1990),

753-772.

20. Piessens, R. and Huysmans R., *Algorithm 619. Automatic numerical inversion of the Laplace transform*, ACM Trans. Math. Softw 10 (1984), 348-353.
21. O. P. Sharma and J. Dass, *Transient analysis for $M/M/m/N$ machine interference model*, J. Stochastic Analysis and Applications **6** (1988), 205-216.
22. Stehfest, H., *Algorithm 368. Numerical inversion of Laplace transforms*,, Comm. ACM **13** (1970), 47-49; (erratum **13** 624).
23. Veillon, F., *Algorithm 486. Numerical inversion of Laplace transforms*, Comm. ACM **17** (1974), 587-589.
24. D. D. W. Yao, *Refining the diffusion approximation for the $M/G/m$ queue*, Operations Research **33** (1985), 1266-1277.
25. D. D. W. Yao and J. A. Buzacott, *Queueing models for a flexible machining station. I. the diffusion approximation*, European J. Operations Research **19** (1985), 233-240.

Department of Statistics
Changwon National University
Changwon 641-773, Korea