sciendo

Int. J. Appl. Math. Comput. Sci., 2022, Vol. 32, No. 1, 125–138
DOI: 10.34768/amcs-2022-0010

amcs

A MULTI–SOURCE FLUID QUEUE BASED STOCHASTIC MODEL OF THE PROBABILISTIC OFFLOADING STRATEGY IN A MEC SYSTEM WITH MULTIPLE MOBILE DEVICES AND A SINGLE MEC SERVER

HUAN ZHENG [a,b], SHUNFU JIN [a,b,*]

[a]School of Information Science and Engineering
Yanshan University
No. 438 West Hebei Avenue, Qinhuangdao 066004, China

[b]Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province
Yanshan University
No. 438 West Hebei Avenue, Qinhuangdao 066004, China
e-mail: jsf@ysu.edu.cn

Mobile edge computing (MEC) is one of the key technologies to achieve high bandwidth, low latency and reliable service in fifth generation (5G) networks. In order to better evaluate the performance of the probabilistic offloading strategy in a MEC system, we give a modeling method to capture the stochastic behavior of tasks based on a multi-source fluid queue. Considering multiple mobile devices (MDs) in a MEC system, we build a multi-source fluid queue to model the tasks offloaded to the MEC server. We give an approach to analyze the fluid queue driven by multiple independent heterogeneous finite-state birth-and-death processes (BDPs) and present the cumulative distribution function (CDF) of the edge buffer content. Then, we evaluate the performance measures in terms of the utilization of the MEC server, the expected edge buffer content and the average response time of a task. Finally, we provide numerical results with some analysis to illustrate the feasibility of the stochastic model built in this paper.

Keywords: mobile edge computing, probabilistic offloading strategy, multi-source fluid queue, birth-and-death process, cumulative distribution function.

1. Introduction

The fifth generation (5G) era is approaching, various application scenarios and differentiated service demands are challenging 5G networks in terms of throughput, latency and reliability (Liu *et al.*, 2020; Razaque *et al.*, 2021). Mobile edge computing (MEC) is one of the key technologies to achieve high bandwidth, low latency and reliable services in 5G networks. By deploying resources of computing, storage and service at the edge of the network, MEC enables the central network to reduce congestion and effectively respond to users' requests. Since MEC is located within the radio access network and close to the mobile users, higher bandwidth and lower latency can be achieved to meet users' quality-of-service (QoS) requirements.

With the rapid development and widespread use of the Internet of everything (IoE), more and more mobile devices (MDs), such as mobile phones, tablets and wearable health devices (Goścień and Walkowiak, 2017), are becoming part of the IoE. In this context, a variety of applications, such as connected cars, augmented reality and interactive games, have emerged. Some computation-intensive applications running on the MDs require a large amount of computing and storage resources (Hassan *et al.*, 2015), and accelerate the power consumption of the MDs. However, due to limited battery power, computing capacity and cache size, MDs cannot provide enough resources to achieve satisfactory service (Bai *et al.*, 2020; Lim *et al.*, 2020).

A possible way to overcome this problem is to offload part of tasks to other servers (Mukherjee *et al.*, 2020). The existing task offloading strategies include the Lyapunov optimization-based offloading decision and resource allocation strategy (Wu *et al.*, 2020), the

*Corresponding author

© 2022 H. Zheng and S. Jin.
This is an open access article distributed under
the Creative Commons Attribution-NonCommercial-NoDerivs license
(http://creativecommons.org/licenses/by-nc-nd/3.0/).

deep reinforcement learning-based dynamic offloading strategy (Xu *et al*., 2020), the probabilistic offloading strategy (Bista *et al*., 2020), etc. Traditional mobile cloud computing (MCC) uploads tasks to powerful cloud servers for processing. The major limitations of MCC are the privacy protection, the energy consumption and the latency experienced in reaching the cloud provider through a wide area network (WAN). By using MEC, tasks can be offloaded to nearby edge servers, which satisfies the requirements of high bandwidth, low latency and reliable service. MEC has been applied to a variety of scenarios, such as cyber-physical systems (CPSs) (Song *et al*., 2021) and the Internet of vehicles (IoV) (Xu *et al*., 2020).

In this paper, considering a MEC system with multiple MDs and a single MEC server, we propose a MEC architecture to investigate the probabilistic offloading strategy. By using a multi-source fluid queue, we propose a modeling method to capture the stochastic behavior of tasks, and present an analytic process for the cumulative distribution function (CDF) of the edge buffer content.

The contributions and main results of this paper are summarized as follows:

1. According to the probabilistic offloading strategy, we give a MEC architecture composed of multiple MDs sharing one MEC server, and give a modeling method to capture the stochastic behavior of tasks based on multi-source fluid queue.

2. We set forth an approach to analyze the fluid queue driven by multiple independent heterogeneous birth-and-death processes (BDPs), and present the CDF of the edge buffer content. Then, we derive performance measures in terms of the utilization of the MEC server, the expected edge buffer content and the average response time of a task.

3. By considering a MEC system composed of two MDs as an example, we present the CDF of the edge buffer content in closed form, and carry out numerical results with analysis to investigate the impacts of system parameters on the system performance.

The outline of the paper is as follows: Section 2 surveys related works. The architecture of the MEC system with the probabilistic offloading strategy is presented in Section 3. In Section 4, we present an analytical approach to derive the fluid queue driven by multiple independent heterogeneous BDPs. By considering a special case with two MDs, we present the CDF of the edge buffer content in Section 5. Section 6 gives the performance measures and numerical illustrations to evaluate the feasibility of the multi-source fluid queue model. Finally, conclusions are summarized in Section 7.

2. Related works

Since the emergence of MEC, task offloading has always been one of the hot research spots in related fields. In the available research, investigations on the performance evaluation of task offloading in MEC are mainly based on the traditional queueing theory.

By applying M/G/1 and M/G/m queues, Li (2019) established a system model for multiple user equipments (UEs) and a single MEC to derive the average response time and the average power consumption of each UE and the MEC. Cardellini *et al*. (2016) considered a three-tier network structure, and modeled the MD and the cloudlet as two M/G/1/PS queues to capture the resources contention on these two systems. Li and Jin (2021) built a MEC architecture with a heterogeneous edge and applied M/M/1, M/M/c and M/M/∞ queues to capture the execution process of tasks. Zhao *et al*. (2017) studied a scheduling problem for heterogeneous clouds, including an edge cloud and a remote cloud, and modeled the edge cloud with multiple virtual machines (VMs) as M/M/1 queues to optimize offloading decisions and computational resource allocation. Nouri *et al*. (2020) considered a two-tier heterogeneous network in MEC, and applied M/M/1 and M/M/c queues to evaluate delay and energy consumption.

In the above research, the arrival and the departure of tasks are regarded as discrete events, which fits the discrete nature of the traditional queueing model. At present, the research of queueing theory mainly focuses on the discrete-event queueing model (Zeifman *et al.*, 2018; 2020). In high-speed communication networks, the scale and the speed of network communication are increasing. Applying traditional discrete-event queueing models to modern networks has become increasingly complicated. The discrete systems are approaching the corresponding continuous systems. As a result, fluid queue models have drawn attention and been applied in modern communication networks.

In a fluid queue model, fluid flows into and out of the buffer according to a random process in the external environment, similar to the arrival and the departure of customers in a traditional queueing model. The input of the fluid buffer can be a single ON-OFF source or multiple ON-OFF sources.

Theoretical research on single-source fluid queues has been well studied. Virtamo and Norros (1994) first investigated a fluid queue model in which the driven process is an M/M/1 queue. Sericola *et al*. (2005) analyzed the transient distribution of the fluid queue driven by an M/M/1 queue. Mao *et al*. (2012) studied a fluid model driven by an M/M/1 queue with

multiple exponential vacations and N-policy. Lenin and Parthasarathy (2000) studied a fluid queue driven by an M/M/1/N queue, and derived closed-form expressions for the eigenvalues and eigenvectors of the underlying tridiagonal matrix.

Single-source fluid queues have also come into use in modern communication networks. By using the spectral analysis method, Arunachalam *et al*. (2010) applied a fluid queue driven by two independent BDPs to a wireless network based on the IEEE 802.11 standard. They derived the buffer occupancy distribution of an intermediate node in which the inflow rate is determined by one BDP and the outflow rate is determined by another BDP. El-Baz *et al*. (2020) modeled a cloud storage facility as a fluid queue driven by an M/M/1/N queue, and obtained the analytical solution of the distribution of the buffer occupancy.

In practice, multiple clients are served by a single server, multiple users are served by a single base station, and so on. Therefore, it is necessary to extend the single-source fluid queues to multi-source fluid queues when modeling wireless communication networks.

Most available studies on multi-source fluid queue are based on the work of Anick *et al*. (1982). They established a fluid queue model to study a data-handling switch with N sources and a single channel. Each source independently and asynchronously alternates between ON and OFF states. The inflow rate of the fluid buffer is determined by an M/M/N/N/N queue. Mitra (1988) extended the fluid queue model to the case of multiple input sources and multiple output lines. Elwalid and Mitra (1995) studied a fluid queue model with two buffers and two classes of sources to model an ATM system with different QoS requirements. Kim and Krunz (2000) considered the traffic sources as ON-OFF sources, and investigated the packet loss performance using the Chernoff dominant eigenvalue (CDE) method for the case where several streams are multiplexed onto one wireless link.

In the multi-source fluid queues above, the ON period and the OFF period are specialized to follow exponential distributions. For the case of N homogeneous ON-OFF sources, the number of ON sources behaves like the number of customers in an M/M/N/N/N queue. Therefore, the inflow rate of the fluid buffer is determined by a BDP.

In this paper, we consider a MEC system with multiple MDs and a single MEC server. We establish two single-server queues for an MD, one for the local processing unit and another for the network adapter. We treat the task flow output from the network adapter of an MD as an ON-OFF source generated by the single-server queue. The ON period is in fact the busy period of the single-server queue for the network adapter. Obviously, the ON period no longer follows an exponential distribution. Therefore, the classical multi-source fluid queue models are not applicable to the MEC system considered in this paper. In order to capture the stochastic behavior of tasks offloaded to the MEC server, we build a multi-source fluid queue, in which the inflow rate of the edge buffer is determined by multiple BDPs. We give an approach to analyze the fluid queue driven by multiple independent heterogeneous BDPs, and present the CDF of the edge buffer content in closed form. Finally, we evaluate the performance measures of the probabilistic offloading strategy in the MEC system in terms of the utilization of the MEC server, the expected edge buffer content and the average response time of a task.

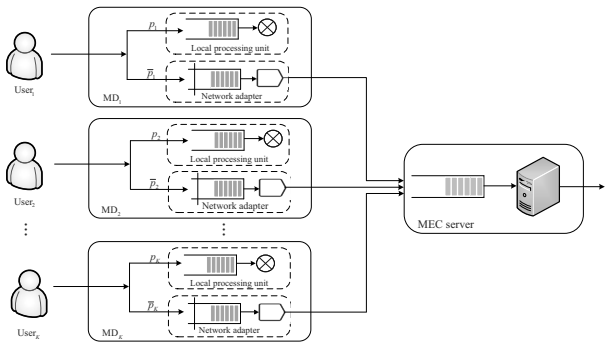


Fig. 1. Architecture of the MEC system.

3. System model

3.1. Architecture of the MEC system with the probabilistic offloading strategy. From a functional point of view, the MEC is a distributed computing system where MEC servers collaborate to provide services to MDs. In this paper, we focus on one of the physical nodes in the distributed system to investigate the probabilistic offloading strategy. For this, we consider a MEC system composed of K $(1 \leq K < \infty)$ MDs sharing one MEC server. The architecture of the MEC system is illustrated in Fig. 1.

The set of MDs in the MEC system is denoted by $\mathcal{K} = \{1, 2, \ldots, K\}$. For the k-th MD, denoted by MD_k $(k \in \mathcal{K})$, we focus on two components: the local processing unit and the network adapter. A single buffer with great capacity and a single-core processor are deployed on the local processing unit. In order to optimize the response performance of the MEC system, we set an access threshold N_k $(N_k > 0)$ for the network adapter. This means that when the number of tasks on the network adapter has reached N_k, i.e., the number of tasks waiting in the buffer has reached $N_k - 1$, the newly offloaded tasks will be discarded. We consider the probabilistic offloading strategy based on binary offloading in which tasks can only be executed locally as a whole or offloaded

to the MEC server. The tasks generated on the MD_k are allocated to the local processing unit with probability p_k $(0 < p_k < 1)$, or offloaded to the MEC server with probability $\bar{p}_k = 1 - p_k$.

In order to better evaluate the performance of the MEC system, we build a system model composed of a local processing model and an edge offloading model.

3.2. Local processing model. Tasks allocated to the local processing unit are executed on the processor at the local processing unit. Based on the execution process of tasks on the local processing unit at MD_k $(k \in \mathcal{K})$, we establish a local processing model.

The generation of tasks at the MD_k is supposed to follow a Poisson process with arrival rate $\lambda_k^{(s)}$. Tasks allocated to the local processing unit first queue in the buffer waiting to get service. Following a first-come first-served (FCFS) policy, the processor serves one task at a time. We assume that the service time of a task follows an exponential distribution with parameter μ_k. Therefore, we model the local processing unit of MD_k as an M/M/1 queue, where the arrival rate is $\lambda_k = p_k\lambda_k^{(s)}$ and the service rate is μ_k.

3.3. Edge offloading model. Tasks to be offloaded first queue in the network adapter at MD, and then transmit to the MEC server over the wireless channel, such as cellular mobile network and Wi-Fi network. Tasks departing from all the MDs will gather at the MEC server. Based on the traffic flows of the offloaded tasks, we establish an edge offloading model.

For MD_k, when a task arrives at the network adapter and the number of tasks in the buffer is less than $N_k - 1$, i.e., the number of tasks on the network adapter does not reach the threshold N_k, the task will queue in the buffer to get service following an FCFS policy. Since the task arrives at the network adapter is a branch of the Poisson process with arrival rate $\lambda_k^{(s)}$, the task that arrives at the network adapter follows a Poisson process with arrival rate $\tilde{\lambda}_k = \bar{p}_k\lambda_k^{(s)}$. We assume that the service time of a task at the local network adapter follows an exponential distribution with parameter $\tilde{\mu}_k$.

Let $Y_k(t)$ denote the number of tasks on the network adapter of MD_k at time t. Based on the analysis before, $\{Y_k(t), t \geq 0\}$ is a BDP with a finite state space $S_k = \{0, 1, 2, \ldots, N_k\}$. The birth rate and the death rate of the BDP $\{Y_k(t), t \geq 0\}$ are $\tilde{\lambda}_k$ and $\tilde{\mu}_k$, respectively. Let $\pi_j^{(k)}$ denote the steady-state probability that the number of tasks on the network adapter is j $(j \in S_k)$. Then

$$\pi_j^{(k)} = \lim_{t\to\infty} P\{Y_k(t) = j\}. \tag{1}$$

After getting processed by the local network adapter, a task will be transmitted to the MEC server over the wireless network. In practice, the size of an individual task is usually very small compared with the capacity of the buffer on the MEC server, called the edge buffer. The size of an individual task is not important to evaluate the edge buffer content. Therefore, for the tasks offloaded to the edge, we are not concerned with the processing of an individual task, but with the processing of the task flow. We regard a task flow output from a network adapter and injected into the edge buffer as a continuous fluid with ON and OFF periods. If a task is leaving the network adapter, the fluid is in ON state, otherwise it is in OFF state. The fluid output from the network adapter at MD_k is defined by the rate

$$R_{Y_k(t)} = \begin{cases} \tilde{\mu}_k, & Y_k(t) > 0, \\ 0, & Y_k(t) = 0. \end{cases} \tag{2}$$

Through the wireless networks, the fluid outputs from all the local network adapters inject into the edge buffer waiting to get service. We assume that the fluids receive services from the MEC server at a constant service rate c, and output from the MEC system. We ignore the discrete nature of the task flow, and treat a task flow output from a network adapter as a fluid. Therefore, the edge offloading model can be considered as a fluid queue with multiple sources. The fluid queue is modulated by K independent heterogeneous BDPs.

The inflow rate of the edge buffer is determined by all the BDPs $\{Y_k(t), t \geq 0\}$ $(k \in \mathcal{K})$, and the outflow rate is constant c. Therefore, we obtain a continuous-time Markov chain (CTMC) $\{Z(t), t \geq 0\}$, where $Z(t) = (Y_1(t), Y_2(t), \ldots, Y_K(t))$. The state space of $\{Z(t), t \geq 0\}$ is given as $S = S_1 \times S_2 \times \cdots \times S_K$, where the symbol $\times$ indicates the Cartesian product.

4. Model analysis of the fluid queue

As discussed in Section 3, we obtain a fluid queue driven by the CTMC $\{Z(t), t \geq 0\}$ with an infinite edge buffer content. The change rate of the edge buffer content is determined by the state of $\{Z(t), t \geq 0\}$ and the outflow rate c of the edge buffer. In this section, we obtain the steady-state condition of the fluid queue, and present the CDF of the edge buffer content.

4.1. Steady-state condition of the fluid queue. Let $\pi_{(i_1,i_2,\ldots,i_K)}$ be the steady-state probability that $Z(t)$ is in state $(i_1, i_2, \ldots, i_K)$. We have

$$\begin{aligned} \pi_{(i_1,i_2,\ldots,i_K)} &= \lim_{t\to\infty} P\{Z(t) = (i_1, i_2, \ldots, i_K)\} \\ &= \pi_{i_1}^{(1)}\pi_{i_2}^{(2)}\ldots\pi_{i_K}^{(K)}, \\ &\qquad (i_1, i_2, \ldots, i_K) \in S, \end{aligned} \tag{3}$$

where $\pi_{i_k}^{(k)}$ $(k \in \mathcal{K})$ is given in Eqn. (1).

By arranging the states of the CTMC $\{Z(t), t \geq 0\}$ in lexicographic order, the steady-state probability distribution of $\{Z(t), t \geq 0\}$ can be written down as

$$\boldsymbol{\pi} = (\pi_{(0,0,\ldots,0)}, \pi_{(0,0,\ldots,1)}, \ldots, \pi_{(N_1,N_2,\ldots,N_K)})^{\mathrm{T}}, \quad (4)$$

where 'T' stands for the a transpose operation.

We denote by $\boldsymbol{Q}$ the infinitesimal generator of the CTMC $\{Z(t), t \geq 0\}$ which is assumed to be irreducible.

The inflow rate $R_Z(t)$ of the fluid queue is determined by the CTMC $\{Z(t), t \geq 0\}$. It is calculated as follows:

$$R_{Z(t)} = \sum_{k \in \mathcal{K}} R_{Y_k(t)}, \quad (5)$$

where $R_{Y_k(t)}$ is given by Eqn. (2).

Equation (5) implies that, at time t, some fluids are in ON state and others are in OFF state. The inflow rate $R_{Z(t)}$ of the edge buffer is the sum of all the output rates for the fluids in the ON state. We define the net inflow rate $r_{Z(t)}$ as the difference between inflow rate $R_{Z(t)}$ and outflow rate c. Here $r_{Z(t)}$ is assumed to be either positive or negative and can be calculated as

$$r_{Z(t)} = R_{Z(t)} - c. \quad (6)$$

Each state of the CTMC $\{Z(t), t \geq 0\}$ has a drift value $r_{(i_1,i_2,\ldots,i_K)}$, $r_{(i_1,i_2,\ldots,i_K)} \neq 0$. We note that there is at least one $r_{(i_1,i_2,\ldots,i_K)} > 0$. Otherwise, the edge buffer will remain empty all the time. Depending on $r_{(i_1,i_2,\ldots,i_K)}$, the states of $\{Z(t), t \geq 0\}$ are classified into over-load states S^+ and under-load states S^- as follows:

$$\begin{aligned} S^+ &= \{(i_1, i_2, \ldots, i_K) \in S | r_{(i_1,i_2,\ldots,i_K)} > 0\}, \\ S^- &= \{(i_1, i_2, \ldots, i_K) \in S | r_{(i_1,i_2,\ldots,i_K)} < 0\}. \end{aligned} \quad (7)$$

Let $s^+ = |S^+|$, $s^- = |S^-|$ and $s = |S|$. Obviously, $S = S^+ \cup S^-$ and $s = s^+ + s^- = (N_1 + 1)(N_2 + 1)\ldots(N_K + 1)$.

Let $C(t)$ be the edge buffer content of the fluid queue at time t. Clearly, $C(t)$ is a non-negative random variable. It is determined by the net inflow rate $r_{Z(t)}$. We can obtain the differential equation of $C(t)$ as follows:

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \begin{cases} r_{Z(t)}, C(t) > 0, \\ 0, \quad C(t) = 0, r_{Z(t)} < 0. \end{cases} \quad (8)$$

Equation (8) implies that when the edge buffer is not empty, the edge buffer content $C(t)$ varies at rate $r_{Z(t)}$; when the edge buffer content reaches zero and the net inflow rate $r_{Z(t)}$ is negative, the edge buffer remains empty until $r_{Z(t)}$ becomes positive.

In order to guarantee the convergence and stability of the distribution of $C(t)$, the steady-state net inflow rate should be negative. The steady-state condition of the fluid queue with multiple sources considered in this paper can be written as follows:

$$d = \sum_{(i_1,i_2,\ldots,i_K) \in S} r_{(i_1,i_2,\ldots,i_K)} \pi_{(i_1,i_2,\ldots,i_K)} < 0, \quad (9)$$

where d is called the mean drift of the fluid queue. In the following analysis, this steady-state condition is assumed to be constantly satisfied.

4.2. Edge buffer content distribution. Let $F_{(i_1,i_2,\ldots,i_K)}(t, x)$ be the instantaneous joint distribution function of the $(K + 1)$-dimensional Markov process $\{(Z(t), C(t)), t \geq 0\}$. $F_{(i_1,i_2,\ldots,i_K)}(t, x)$ can be written as follows:

$$\begin{aligned} &F_{(i_1,i_2,\ldots,i_K)}(t, x) \\ &\quad = P\{Z(t) = (i_1, i_2, \ldots, i_K), C(t) \leq x\}, \\ &\quad (i_1, i_2, \ldots, i_K) \in S, \quad x \geq 0. \end{aligned} \quad (10)$$

When the Markov process $\{(Z(t), C(t)), t \geq 0\}$ is stable, its steady-state random variable is denoted by (Z, C). The steady-state joint distribution function $F_{(i_1,i_2,\ldots,i_K)}(x)$ can be given as follows:

$$F_{(i_1,i_2,\ldots,i_K)}(x) = \lim_{t \to \infty} F_{(i_1,i_2,\ldots,i_K)}(t, x). \quad (11)$$

By arranging the states of the CTMC $\{Z(t), t \geq 0\}$ in lexicographic order, we introduce the vector

$$\begin{aligned} \boldsymbol{F}(x) = (&F_{(0,0,\ldots,0)}(x), F_{(0,0,\ldots,1)}(x), \\ &\ldots, F_{(N_1,N_2,\ldots,N_K)}(x))^{\mathrm{T}}. \end{aligned} \quad (12)$$

Then the system of ordinary differential equations can be written in a matrix form as follows:

$$\boldsymbol{\Lambda} \frac{\mathrm{d}}{\mathrm{d}x} \boldsymbol{F}(x) = \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{F}(x), \quad (13)$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix, called the rate matrix. The diagonal elements are the drift values of the states arranged in lexicographic order. $\boldsymbol{\Lambda}$ can be given as follows:

$$\boldsymbol{\Lambda} = \mathrm{diag}\,(r_{(0,0,\ldots,0)}, r_{(0,0,\ldots,1)}, \ldots, r_{(N_1,N_2,\ldots,N_K)}). \quad (14)$$

When the drift value is positive, the edge buffer content must increase, so the boundary conditions can be given as follows:

$$F_{(i_1,i_2,\ldots,i_K)}(0) = 0, \quad (i_1, i_2, \ldots, i_K) \in S^+, \quad (15)$$

where S^+ is defined in Eqn. (7).

$$\begin{aligned} F_{(i_1,i_2,\ldots,i_K)}(\infty) &= \lim_{x \to \infty} F_{(i_1,i_2,\ldots,i_K)}(x) \\ &= \pi_{(i_1,i_2,\ldots,i_K)}. \end{aligned} \quad (16)$$

By solving this system of ordinary differential

equations with boundary conditions, we can derive the steady-state joint distribution function of the Markov process $\{(Z(t), C(t)), t \geq 0\}$ and hence the CDF of the edge buffer content.

Lemma 1. *The following results hold for matrix* $\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{\mathrm{T}}$*:*

1. *Matrix* $\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{\mathrm{T}}$ *has* s *eigenvalues, denoted by* $\xi_1, \xi_2, \ldots, \xi_s$.
2. *One of the eigenvalues for matrix* $\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{\mathrm{T}}$ *is zero.*
3. *The numbers of eigenvalues with negative and positive real parts for matrix* $\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{\mathrm{T}}$ *are* s^+ *and* $s^- - 1$*, respectively.*

According to Lemma 1, we order the real parts of the eigenvalues for matrix $\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}^{\mathrm{T}}$ as follows:

$$R_e(\xi_1) > R_e(\xi_2) > \ldots > R_e(\xi_{s^- - 1}) > R_e(\xi_{s^-}) = 0 > R_e(\xi_{s^- + 1}) > \cdots > R_e(\xi_s). \tag{17}$$

The set of numbers $\{\xi_j,\ j = 1, 2, \ldots, s\}$ is called the spectrum of the fluid queue. The general solution of Eqn. (13) can be expressed as follows:

$$\boldsymbol{F}(x) = \sum_{j=1}^{s} a_j e^{\xi_j x} \boldsymbol{\psi}_j, \tag{18}$$

where

$$\boldsymbol{\psi}_j = (\psi_j^{(0,0,\ldots,0)}, \psi_j^{(0,0,\ldots,1)}, \ldots, \psi_j^{(N_1,N_2,\ldots,N_K)})^{\mathrm{T}}, \quad j \in \{1, 2, \ldots, s\}$$

is the eigenvector corresponding to the eigenvalue ξ_j.

Since $\boldsymbol{F}(x)$ is a probability vector, a_j satisfies the condition $R_e(\xi_j) > 0 \Rightarrow a_j = 0$. Otherwise, the solution of Eqn. (13) will increase at an exponential rate, and the fluid queue will no longer be stable. Then, according to Eqn. (17), we can simplify Eqn. (18) as follows:

$$\boldsymbol{F}(x) = \sum_{j=s^-}^{s} a_j e^{\xi_j x} \boldsymbol{\psi}_j. \tag{19}$$

The coefficients $a_j,\ j \in \{1, 2, \ldots, s\}$ are determined by the boundary conditions. According to the boundary condition given in Eqn. (16), we have $a_{s^-}\boldsymbol{\psi}_{s^-} = \boldsymbol{\pi}$, where $\boldsymbol{\pi}$ is given in Eqn. (4). The remaining coefficients can be determined by the boundary condition given in Eqn. (15). According to Eqns. (15) and (16), Eqn. (19) can be written as follows:

$$\pi_{(i_1,i_2,\ldots,i_K)} + \sum_{j=s^-+1}^{s} a_j \psi_j^{(i_1,i_2,\ldots,i_K)} = 0, \quad (i_1, i_2, \ldots, i_K) \in S^+. \tag{20}$$

The system of linear equations in Eqn. (20) can be written in a matrix form as $\boldsymbol{\Psi x} = \boldsymbol{\tau}$. In the fluid queue with multiple sources, when the number of sources increases, the number of sources in the ON state increases accordingly. This may lead to an increase in the number s^+ of the over-load states S^+. From Eqns. (7) and (20), we note that the increase in s^+ leads to an increase in the size of the matrix equation, which may lead to ill-behaved numerical results (Kulkarni, 1997; Fiedler and Voos, 2000).

We first apply the so-called Gaussian transformation to get a positive definite matrix. The matrix equation $\boldsymbol{\Psi x} = \boldsymbol{\tau}$ can be replaced with $\widehat{\boldsymbol{\Psi}}\boldsymbol{x} = \widehat{\boldsymbol{\tau}}$, where $\widehat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$ and $\widehat{\boldsymbol{\tau}} = \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\tau}$. Then, in order to reduce the condition number of the matrix $\widehat{\boldsymbol{\Psi}}$, we perform an incomplete Cholesky (IC) factorization for the matrix to obtain the preconditioner $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \approx \widehat{\boldsymbol{\Psi}}$, where cond $(\boldsymbol{M}^{-1}\widehat{\boldsymbol{\Psi}}) \ll$ cond $(\widehat{\boldsymbol{\Psi}})$. The matrix equation can be further converted to $\boldsymbol{L}^{-1}\widehat{\boldsymbol{\Psi}}\boldsymbol{L}^{-\mathrm{T}}\widehat{\boldsymbol{x}} = \boldsymbol{L}^{-1}\widehat{\boldsymbol{\tau}}$, where $\widehat{\boldsymbol{x}} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{x}$. The process of using the conjugate gradient (CG) method to solve the new matrix equation is called the incomplete Cholesky conjugate gradient (ICCG) method. The ICCG method is given as Algorithm 1.

Let $F(x)$ and $\bar{F}(x)$ be the CDF and the residual CDF of the edge buffer content $C(t)$, respectively. We have

$$F(x) = \lim_{t \to \infty} P\{C(t) \leq x\}, \tag{21}$$

$$\bar{F}(x) = 1 - F(x). \tag{22}$$

Equation (21) can be expressed as follows:

$$F(x) = \boldsymbol{e}_s \boldsymbol{F}(x), \tag{23}$$

where $\boldsymbol{e}_s$ is the s-dimensional row vector of ones.

Algorithm 1. ICCG method.

Require: $\boldsymbol{\Psi}, \boldsymbol{\tau}, \boldsymbol{x}_0$

Calculate $\widehat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}$, $\widehat{\boldsymbol{\tau}} = \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\tau}$;
Do the incomplete Cholesky (IC) factorization of $\widehat{\boldsymbol{\Psi}}$, $\widehat{\boldsymbol{\Psi}} \approx \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}$;
Initialize $k = 0$, $\boldsymbol{r}_0 = \widehat{\boldsymbol{\tau}} - \widehat{\boldsymbol{\Psi}}\boldsymbol{x}_0$, $\widehat{\boldsymbol{r}}_0 = \boldsymbol{L}^{-1}\boldsymbol{r}_0$, $\boldsymbol{P}_0 = \boldsymbol{L}^{-\mathrm{T}}\widehat{\boldsymbol{r}}_0$;
while $\| \widehat{\boldsymbol{r}}_k \|_2 > \varepsilon$ **do**
  $\alpha_k = (\widehat{\boldsymbol{r}}_k, \widehat{\boldsymbol{r}}_k)/(\widehat{\boldsymbol{\Psi}}\boldsymbol{P}_k, \boldsymbol{P}_k)$;
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{P}_k$;
  $\widehat{\boldsymbol{r}}_{k+1} = \widehat{\boldsymbol{r}}_k - \alpha_k \boldsymbol{L}^{-1}\widehat{\boldsymbol{\Psi}}\boldsymbol{P}_k$;
  $\beta_k = (\widehat{\boldsymbol{r}}_{k+1}, \widehat{\boldsymbol{r}}_{k+1})/(\widehat{\boldsymbol{r}}_k, \widehat{\boldsymbol{r}}_k)$;
  $\boldsymbol{P}_{k+1} = \boldsymbol{L}^{-\mathrm{T}}\widehat{\boldsymbol{r}}_{k+1} + \beta_k \boldsymbol{P}_k$;
  $k = k + 1$.
end while
return $\boldsymbol{x}_k$

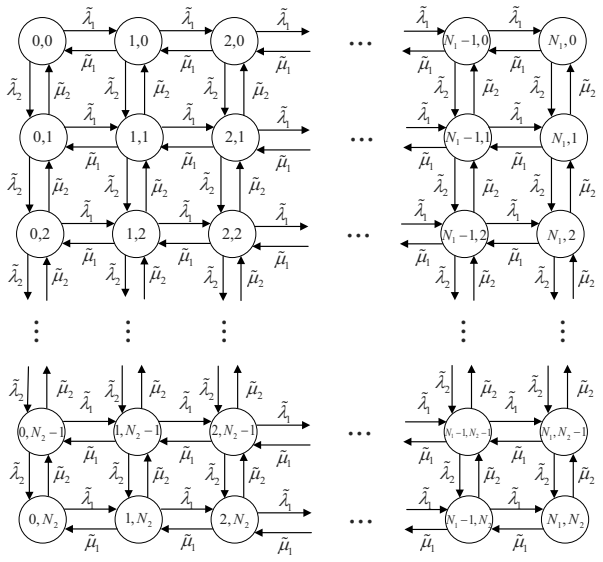


Fig. 2. State transition of the CTMC $\{Z(t), t \geq 0\}$.

According to Eqns. (19) and (23),

$$F(x) = 1 + \sum_{j=s^-+1}^{s} a_j e^{\xi_j x} \boldsymbol{e_s} \boldsymbol{\psi}_j. \tag{24}$$

5. Special case with two MDs

For a MEC system composed of two MDs, MD_1 and MD_2, sharing one MEC server, we present the CDF of the edge buffer content following the analytical procedure in Section 4.

We establish BDPs $\{Y_1(t), t \geq 0\}$ and $\{Y_2(t), t \geq 0\}$ to model the processing of tasks on the network adapter at MD_1 and MD_2, respectively. Furthermore, we obtain a CTMC $\{Z(t), t \geq 0\}$ with state space

$$\begin{aligned} S = \{&(0,0), \ldots, (0,N_2), (1,0), \ldots, (1,N_2), \\ &\ldots, (N_1,0), \ldots, (N_1,N_2)\}. \end{aligned} \tag{25}$$

The state transition of the CTMC $\{Z(t), t \geq 0\}$ is illustrated in Fig. 2.

From Eqn. (25), we note that the CTMC $\{Z(t), t \geq 0\}$ has $s = (N_1+1)(N_2+1)$ states. According to Eqn. (3), the steady-state probability $\pi_{(i_1,i_2)}$ that $Z(t)$ is in state $(i_1, i_2) \in S$ can be written as follows:

$$\pi_{(i_1,i_2)} = \lim_{t\to\infty} P\{Z(t) = (i_1,i_2)\} = \pi_{i_1}^{(1)} \pi_{i_2}^{(2)}. \tag{26}$$

The infinitesimal generator of the CTMC $\{Z(t), t \geq 0\}$ is given as follows:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{A}_0 & \boldsymbol{C} & & & \\ \boldsymbol{B} & \boldsymbol{A} & \boldsymbol{C} & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{B} & \boldsymbol{A} & \boldsymbol{C} \\ & & & \boldsymbol{B} & \boldsymbol{A}_{N_1} \end{bmatrix}_{s\times s}, \tag{27}$$

where each submatrix is a square matrix of dimension $(N_2+1)\times(N_2+1)$. $\boldsymbol{A}_0$, $\boldsymbol{A}$, $\boldsymbol{A}_{N_1}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are given as follows:

$$\boldsymbol{A}_0 = \begin{bmatrix} -(\tilde{\lambda}_1+\tilde{\lambda}_2) & \tilde{\lambda}_2 & & & & \\ \tilde{\mu}_2 & -(\tilde{\lambda}_1+\tilde{\lambda}_2+\tilde{\mu}_2) & \tilde{\lambda}_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \tilde{\mu}_2 & -(\tilde{\lambda}_1+\tilde{\lambda}_2+\tilde{\mu}_2) & \tilde{\lambda}_2 & \\ & & & \tilde{\mu}_2 & -(\tilde{\lambda}_1+\tilde{\mu}_2) & \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} -(\tilde{\lambda}_1+\tilde{\lambda}_2+\tilde{\mu}_1) & \tilde{\lambda}_2 & & & & \\ \tilde{\mu}_2 & -(\tilde{\lambda}_1+\tilde{\lambda}_2+\tilde{\mu}_1+\tilde{\mu}_2) & \tilde{\lambda}_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \tilde{\mu}_2 & -(\tilde{\lambda}_1+\tilde{\lambda}_2+\tilde{\mu}_1+\tilde{\mu}_2) & \tilde{\lambda}_2 & \\ & & & \tilde{\mu}_2 & -(\tilde{\lambda}_1+\tilde{\mu}_1+\tilde{\mu}_2) & \end{bmatrix},$$

$$\boldsymbol{A}_{N_1} = \begin{bmatrix} -(\tilde{\lambda}_2+\tilde{\mu}_1) & \tilde{\lambda}_2 & & & & \\ \tilde{\mu}_2 & -(\tilde{\lambda}_2+\tilde{\mu}_1+\tilde{\mu}_2) & \tilde{\lambda}_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \tilde{\mu}_2 & -(\tilde{\lambda}_2+\tilde{\mu}_1+\tilde{\mu}_2) & \tilde{\lambda}_2 & \\ & & & \tilde{\mu}_2 & -(\tilde{\mu}_1+\tilde{\mu}_2) & \end{bmatrix},$$

$$\boldsymbol{B} = \operatorname{diag}(\tilde{\mu}_1), \quad \boldsymbol{C} = \operatorname{diag}(\tilde{\lambda}_1).$$

We assume that $\tilde{\mu}_1 > \tilde{\mu}_2 > c$; then, according to Eqn. (6), the drift value $r_{(i_1,i_2)}$, $(i_1,i_2) \in S$ can be given as follows:

$$r_{(i_1,i_2)} = \begin{cases} -c, & i_1 = i_2 = 0, \\ \tilde{\mu}_1 - c, & i_1 > 0,\ i_2 = 0, \\ \tilde{\mu}_2 - c, & i_1 = 0,\ i_2 > 0, \\ \tilde{\mu}_1 + \tilde{\mu}_2 - c, & i_1 > 0,\ i_2 > 0. \end{cases} \tag{28}$$

Based on the value of $r_{(i_1,i_2)}$, we classify the states of the CTMC $\{Z(t), t \geq 0\}$ into under-load states $S^- = \{(0,0)\}$ and over-load states $S^+ = S\backslash\{(0,0)\}$. Obviously, the numbers of elements in S^- and S^+ are $s^- = 1$ and $s^+ = s-1$, respectively.

From Eqns. (9), (26) and (28), the steady-state condition of the fluid queue can be written as follows:

$$\begin{aligned} d = &-c\pi_{(0,0)} + (\tilde{\mu}_1 - c) \sum_{i_1=1}^{N_1} \pi_{(i_1,0)} \\ &+ (\tilde{\mu}_2 - c) \sum_{i_2=1}^{N_2} \pi_{(0,i_2)} \end{aligned}$$

$$+ (\tilde{\mu}_1 + \tilde{\mu}_2 - c) \sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} \pi_{(i_1,i_2)} < 0. \tag{29}$$

According to Eqn. (13), the system of ordinary differential equations for the fluid queue is given in a matrix form as follows:

$$\boldsymbol{\Lambda} \frac{\mathrm{d}}{\mathrm{d}x} \boldsymbol{F}(x) = \boldsymbol{Q}^{\mathrm{T}} \boldsymbol{F}(x), \tag{30}$$

where $\boldsymbol{Q}$ is given in Eqn. (27), and according to Eqn. (12), $\boldsymbol{F}(x)$ is given as follows:

$$\begin{aligned} \boldsymbol{F}(x) = (&F_{(0,0)}(x), \ldots, F_{(0,N_2)}(x), \ldots, \\ &F_{(N_1,0)}(x), \ldots, F_{(N_1,N_2)}(x))^{\mathrm{T}}. \end{aligned} \tag{31}$$

From Eqns. (14) and (28),

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda_0} & & & \\ & \boldsymbol{\Lambda_1} & & \\ & & \ddots & \\ & & & \boldsymbol{\Lambda_1} \end{bmatrix}_{s \times s},$$

$$\boldsymbol{\Lambda_0} = \text{diag}\,(-c, \tilde{\mu}_2 - c, \ldots)_{(N_2+1)\times(N_2+1)},$$

$$\boldsymbol{\Lambda_1} = \text{diag}\,(\tilde{\mu}_1 - c, \tilde{\mu}_1 + \tilde{\mu}_2 - c, \ldots)_{(N_2+1)\times(N_2+1)}.$$

Based on Eqns. (15) and (16), we define the boundary conditions for the system of ordinary differential equations given in Eqn. (30) as follows:

$$F_{(i_1,i_2)}(0) = 0, \quad (i_1, i_2) \in S^+, \tag{32}$$

$$F_{(i_1,i_2)}(\infty) = \pi_{(i_1,i_2)}, \quad (i_1, i_2) \in S. \tag{33}$$

From Eqns. (19) and (33), it follows that

$$\boldsymbol{F}(x) = \sum_{j=1}^{s} a_j e^{\xi_j x} \boldsymbol{\psi}_j = \boldsymbol{\pi} + \sum_{j=2}^{s} a_j e^{\xi_j x} \boldsymbol{\psi}_j, \tag{34}$$

where

$$\boldsymbol{\pi} = (\pi_{(0,0)}, \ldots, \pi_{(0,N_2)}, \ldots, \pi_{(N_1,0)}, \ldots, \pi_{(N_1,N_2)})^{\mathrm{T}}.$$

According to Eqns. (20) and (32), the coefficients $a_j,\ j \in [2, s]$ can be given by solving the following system of linear equations:

$$\pi_{(i_1,i_2)} + \sum_{j=2}^{s} a_j \psi_j^{(i_1,i_2)} = 0, \quad (i_1, i_2) \in S^+. \tag{35}$$

The system of linear equations (35) can be written in matrix form as follows:

$$\begin{bmatrix} \psi_2^{(0,1)} & \psi_3^{(0,1)} & \cdots & \psi_s^{(0,1)} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_2^{(0,N_2)} & \psi_3^{(0,N_2)} & \cdots & \psi_s^{(0,N_2)} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_2^{(N_1,0)} & \psi_3^{(N_1,0)} & \cdots & \psi_s^{(N_1,0)} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_2^{(N_1,N_2)} & \psi_3^{(N_1,N_2)} & \cdots & \psi_s^{(N_1,N_2)} \end{bmatrix} \begin{bmatrix} a_2 \\ \vdots \\ a_{(N_2+1)} \\ \vdots \\ a_{N_1(N_2+1)+1} \\ \vdots \\ a_s \end{bmatrix}$$

$$= \begin{bmatrix} -\pi_{(0,1)} \\ \vdots \\ -\pi_{(0,N_2)} \\ \vdots \\ -\pi_{(N_1,0)} \\ \vdots \\ -\pi_{(N_1,N_2)} \end{bmatrix}. \tag{36}$$

Applying Algorithm 1 to solve Eqn. (36), we can obtain the coefficients $a_j,\ j \in [2, s]$. Then, according to Eqns. (31) and (34), the CDF $F(x)$ of the edge buffer content can be given as follows:

$$F(x) = \boldsymbol{e}_s \boldsymbol{F}(x) = 1 + \sum_{j=2}^{s} a_j e^{\xi_j x} \boldsymbol{e}_s \boldsymbol{\psi}_j. \tag{37}$$

6. Performance measures and numerical results

In this section, we evaluate the probabilistic offloading strategy in the MEC system in terms of the utilization of the MEC server, the expected edge buffer content and the average response time of a task. Then, we present numerical results to demonstrate the impact of system parameters on these performance measures.

6.1. Performance measures. We define the utilization U_s of the MEC server as the probability that the MEC server is busy. For the fluid queue model considered in this paper, the utilization U_s of the MEC server is the probability that the edge buffer is non-empty.

Based on the CDF of the edge buffer content given in Section 4, we compute the utilization U_s of the MEC server as follows:

$$U_s = 1 - F(0), \tag{38}$$

where $F(0)$ can be obtained by Eqn. (24).

We define the expected edge buffer content $E[C]$ as the average number of tasks in the edge buffer.

Based on the CDF of the edge buffer content given in

Section 4, we deduce the expected edge buffer content

$$E[C] = \int_0^\infty \bar{F}(x)\,\mathrm{d}x = \sum_{j=s^-+1}^{s} \frac{a_j}{\xi_j} \boldsymbol{e_s}\boldsymbol{\psi}_j, \tag{39}$$

where $\bar{F}(x)$ is defined in Eqn. (22).

We define the average response time $E[T_k]$ of a task generated on the MD_k as the average duration from the epoch a task is generated on the MD_k to the epoch this task is completely executed on the local processing unit or the MEC server.

For a task executed on the local processing unit, called the local task, the response time is the duration from the instant of a task arriving at the local processing unit to the instant of the task departing from the local processing unit.

Applying the analysis result of the M/M/1 queue, we give the average response time $E[T_k^{(l)}]$ of a local task generated on the MD_k as follows:

$$E[T_k^{(l)}] = \frac{1}{\mu_k - \lambda_k}. \tag{40}$$

For a task offloaded to the MEC server, called the offloaded task, the response time is the sum of the processing time on the network adapter, the propagation delay from the MD to the MEC server and the processing time on the MEC server. In a MEC system, the distance from the MD to the MEC server is relatively close; in this paper, we ignore the propagation delay from the MD to the MEC server.

Applying the analysis result of the M/M/1/N_k queue, for an offloaded task generated on the MD_k, we get the average processing time on the network adapter

$$t_k^{(l)} = \frac{1}{\tilde{\mu}_k(1-\tilde{\rho}_k)} - \frac{N_k\tilde{\rho}_k^{N_k}}{\tilde{\mu}_k(1-\tilde{\rho}_k^{N_k})}, \tag{41}$$

where $\tilde{\rho}_k = \tilde{\lambda}_k/\tilde{\mu}_k$.

According to Little's law, the average processing time of an offloaded task on the MEC server is

$$t^{(e)} = \frac{E[C]}{\lambda_{in}}, \tag{42}$$

where

$$\lambda_{\text{in}} = \sum_{(i_1,i_2,\ldots,i_K)\in S} R_{(i_1,i_2,\ldots,i_K)}\pi_{(i_1,i_2,\ldots,i_K)}$$

is the mean inflow rate of the edge buffer content.

Combing Eqns. (41) and (42), the average response time of an offloaded task generated on the MD_k is

$$\begin{aligned} E[T_k^{(e)}] &= t_k^{(l)} + t^{(e)} \\ &= \frac{1}{\tilde{\mu}_k(1-\tilde{\rho}_k)} - \frac{N_k\tilde{\rho}_k^{N_k}}{\tilde{\mu}_k(1-\tilde{\rho}_k^{N_k})} \\ &\quad + \frac{E[C]}{\sum_{(i_1,i_2,\ldots,i_K)\in S} R_{(i_1,i_2,\ldots,i_K)}\pi_{(i_1,i_2,\ldots,i_K)}}. \end{aligned} \tag{43}$$

A task generated on the MD_k is allocated to the local processing unit with probability p_k, or offloaded to the MEC server with probability $\bar{p}_k = 1 - p_k$. Combining Eqns. (40) and (43), we obtain the average response time of a task generated on the MD_k

$$\begin{aligned} E[T_k] &= p_k E[T_k^{(l)}] + \bar{p}_k E[T_k^{(e)}] \\ &= \frac{p_k}{\mu_k - \lambda_k} + \left(\frac{1}{\tilde{\mu}_k(1-\tilde{\rho}_k)} - \frac{N_k\tilde{\rho}_k^{N_k}}{\tilde{\mu}_k(1-\tilde{\rho}_k^{N_k})}\right)\bar{p}_k \\ &\quad + \left(\frac{E[C]}{\sum_{(i_1,i_2,..,i_K)\in S} R_{(i_1,i_2,..,i_K)}\pi_{(i_1,i_2,..,i_K)}}\right)\bar{p}_k. \end{aligned} \tag{44}$$

6.2. Numerical results. We consider a MEC system composed of two MDs and carry out numerical experiments complemented with some analysis of the impact of system parameters on the system performance in terms of the utilization of the MEC server, the expected edge buffer content and the average response time of a task. The results are obtained in Matlab 2016a based on Eqns. (37)–(39) and (44). We note that the steady-state constraints of the system under consideration are $\lambda_1 < \mu_1$, $\lambda_2 < \mu_2$ and the mean drift $d < 0$. Under these constraints, we carry out repeated experiments with different parameters. As an example, we set the parameters in Table 1. The parameter values in Table 1 are chosen only for the purpose of numerical illustration of the closed-form results.

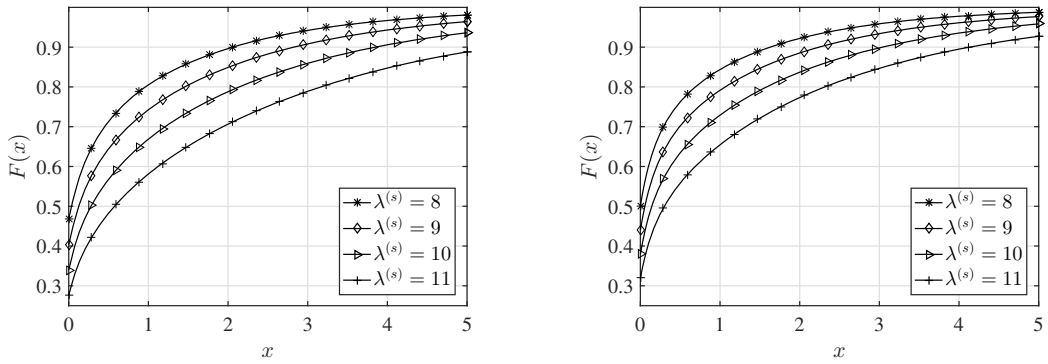
By setting the arrival rate $\lambda_1^{(s)} = \lambda_2^{(s)} = \lambda^{(s)}$ of tasks generated on the MD_1 and the MD_2, we investigate the impact of the arrival rate $\lambda^{(s)}$ of tasks on the CDF $F(x)$ of the edge buffer content with different service rate c of a task on the MEC server in Fig. 3.

As can be seen from all the four subplots in Fig. 3, the curve of the CDF $F(x)$ increases monotonically and takes values in the range $(0, 1)$, which is consistent with the probabilistic interpretation of the distribution function. When the service rate c of a task on the MEC server is fixed, the CDF $F(x)$ of the edge buffer content decreases with an increase in the arrival rate $\lambda^{(s)}$ of tasks.

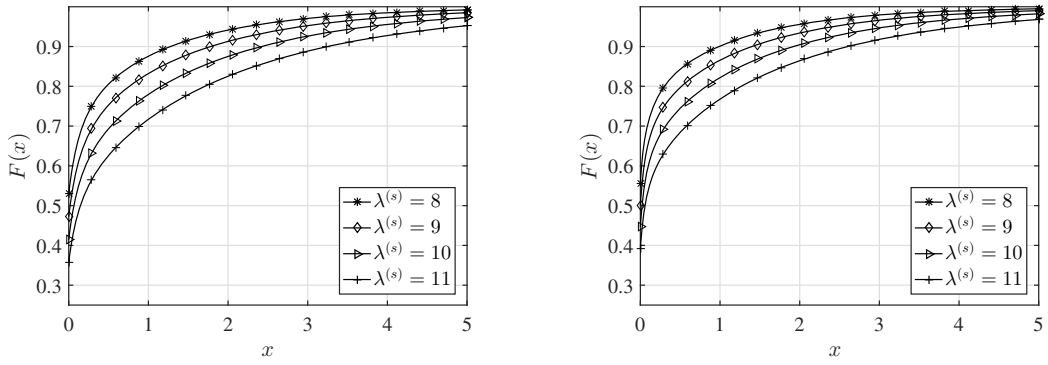
The intersection of the curve with the vertical axis is the probability that the edge buffer is empty, i.e., $F(0) = P\{C = 0\}$. The numerical results of $F(0)$ are shown in Table 2.

Table 1. Parameter settings.

Parameters	Meaning	Values
$\lambda_1^{(s)}$	Arrival rate of tasks generated on the MD_1	$[8, 12]$
$\lambda_2^{(s)}$	Arrival rate of tasks generated on the MD_2	$[8, 12]$
p_1	Allocation rate of a task to the local processing unit at MD_1	0.5
p_2	Allocation rate of a task to the local processing unit at MD_2	0.5
μ_1	Service rate of a task on the local processing unit at MD_1	20
μ_2	Service rate of a task on the local processing unit at MD_2	15
$\tilde{\mu}_1$	Service rate of a task on the network adapter at MD_1	22
$\tilde{\mu}_2$	Service rate of a task on the network adapter at MD_2	19
N_1	Access threshold of the network adapter at MD_1	6
N_2	Access threshold of the network adapter at MD_2	6
c	Service rate of a task on the MEC server	$[14, 18]$



(a) $c = 15$ (b) $c = 16$



(c) $c = 17$ (d) $c = 18$

Fig. 3. CDF $F(x)$ of the edge buffer content.

From Fig. 3 and Table 2, we find that when the service rate c of a task on the MEC server is fixed, $F(0)$ decreases with an increase in the arrival rate $\lambda^{(s)}$ of tasks. This is because an increase in the arrival rate $\lambda^{(s)}$ of tasks leads to an increase in the number of tasks offloaded to the MEC server and a decrease in the probability for the edge buffer to be empty. When the arrival rate $\lambda^{(s)}$ of tasks is fixed, $F(0)$ increases with an increase in the service rate c of a task on the MEC server. This is because an increase in service rate c of a task on the MEC server leads to a decrease in the edge buffer content and an increase in the probability for the edge buffer to be empty.

Figure 4 and Table 3 depict the utilization U_s of the MEC server versus the arrival rate $\lambda^{(s)}$ of tasks for different service rates c of a task on the MEC server.

Table 2. Numerical results for $F(0)$.

	$\lambda^{(s)}=8$	$\lambda^{(s)}=9$	$\lambda^{(s)}=10$	$\lambda^{(s)}=11$	$\lambda^{(s)}=12$
$c=14$	0.4312	0.3621	0.2946	0.2269	0.1585
$c=15$	0.4684	0.4034	0.3398	0.2764	0.2117
$c=16$	0.5011	0.4398	0.3792	0.3193	0.2586
$c=17$	0.5301	0.4720	0.4146	0.3578	0.3001
$c=18$	0.5560	0.5009	0.4463	0.3914	0.3374

Table 3. Numerical results for U_s.

	$\lambda^{(s)}=8$	$\lambda^{(s)}=9$	$\lambda^{(s)}=10$	$\lambda^{(s)}=11$	$\lambda^{(s)}=12$
$c=14$	0.5688	0.6379	0.7054	0.7731	0.8415
$c=15$	0.5317	0.5965	0.6605	0.7236	0.7883
$c=16$	0.4989	0.5602	0.6207	0.6819	0.7414
$c=17$	0.4699	0.5280	0.5852	0.6425	0.6999
$c=18$	0.4440	0.4990	0.5537	0.6086	0.6626

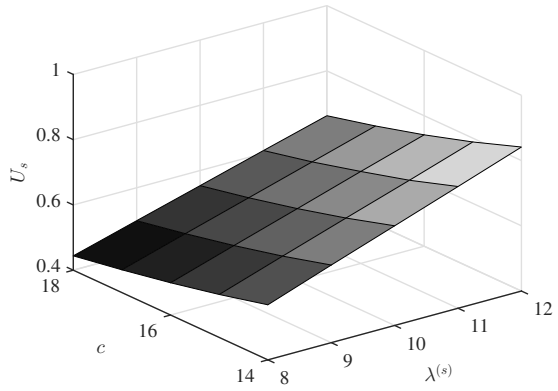


Fig. 4. Utilization of the MEC server.

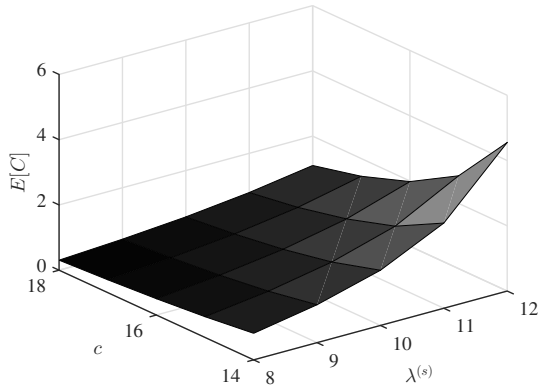


Fig. 5. Expected edge buffer content.

From Fig. 4 and Table 3, we find that the utilization U_s of the MEC server is positively correlated with the arrival rate $\lambda^{(s)}$ of tasks and negatively correlated with service rate c of a task on the MEC server. This is in line with our expectations. More task arrivals means a higher load on the MEC server and higher utilization of the MEC server. Greater service rate means a higher probability of the edge buffer content being empty; hence the utilization of the MEC server is lower.

Figure 5 and Table 4 depict the expected edge buffer content $E[C]$ versus the arrival rate $\lambda^{(s)}$ of tasks for different service rates c of a task on the MEC server.

From Fig. 5 and Table 4, we find that the expected edge buffer content $E[C]$ increases with an increase in the arrival rate $\lambda^{(s)}$ of tasks and a decrease in service rate c of a task on the MEC server, as predicted. This is because an increase in the arrival rate of tasks or a decrease in the service rate of a task on the MEC server leads to an increase in the edge buffer content.

Figure 6 and Table 5 depict the average response times $E[T_1]$ and $E[T_2]$ of a task generated on the MD_1 and the MD_2 versus the arrival rate $\lambda^{(s)}$ of tasks for different service rates c of a task on the MEC server.

From Fig. 6 and Table 5, we find that the average response time $E[T_1]$ of a task generated on the MD_1 is smaller than the average response time $E[T_2]$ of a task generated on the MD_2 when the service rate c of a task on the MEC server is fixed. This is due to the fact that $\mu_1 > \mu_2$ and $\tilde{\mu}_1 > \tilde{\mu}_2$, i.e., the service rates of a task on both the local processing unit and the network adapter at MD_1 are greater than that at MD_2. The average response time of a task increases with an increase in the arrival rate $\lambda^{(s)}$ of tasks or a decrease in the service rate c of a task on the MEC server. The reason is that if the arrival rate of tasks is greater and the service rate of a task on the MEC server is lower, the average response time of a task will be larger.

To further illustrate the differences in all the

Table 4. Numerical results of $E[C]$.

	$\lambda^{(s)}=8$	$\lambda^{(s)}=9$	$\lambda^{(s)}=10$	$\lambda^{(s)}=11$	$\lambda^{(s)}=12$
$c=14$	0.8137	1.1615	1.6918	2.6113	4.5544
$c=15$	0.6360	0.8902	1.2577	1.8283	2.8455
$c=16$	0.5003	0.6931	0.9608	1.3639	1.9814
$c=17$	0.3935	0.5430	0.7429	1.0306	1.4572
$c=18$	0.3073	0.4240	0.5792	0.7993	1.1003

Table 5. Numerical results of $E[T_1]$ and $E[T_2]$.

(a) numerical results of $E[T_1]$

	$\lambda^{(s)}=8$	$\lambda^{(s)}=9$	$\lambda^{(s)}=10$	$\lambda^{(s)}=11$	$\lambda^{(s)}=12$
$c=14$	0.1099	0.1254	0.1473	0.1835	0.2568
$c=15$	0.0988	0.1103	0.1256	0.1479	0.1855
$c=16$	0.0903	0.0993	0.1108	0.1268	0.1495
$c=17$	0.0836	0.0910	0.0999	0.1116	0.1277
$c=18$	0.0782	0.0844	0.0917	0.1011	0.1128

(b) numerical results of $E[T_2]$

	$\lambda^{(s)}=8$	$\lambda^{(s)}=9$	$\lambda^{(s)}=10$	$\lambda^{(s)}=11$	$\lambda^{(s)}=12$
$c=14$	0.1296	0.1466	0.1703	0.2083	0.2837
$c=15$	0.1185	0.1315	0.1486	0.1727	0.2125
$c=16$	0.1100	0.1206	0.1337	0.1516	0.1765
$c=17$	0.1034	0.1122	0.1228	0.1364	0.1546
$c=18$	0.0980	0.1056	0.1146	0.1259	0.1397

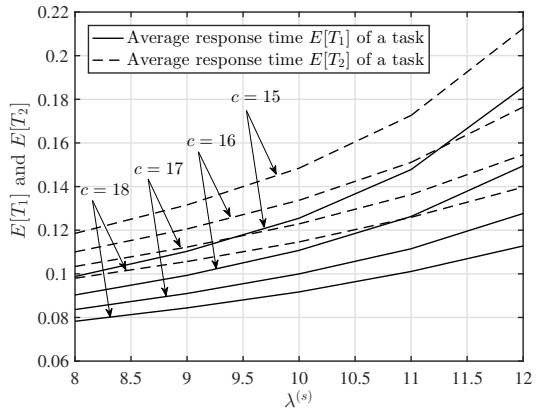


Fig. 6. Average response time of a task.

performance measures with different $\lambda^{(s)}$ and c, we perform the one-sided Wilcoxon rank-sum test to calculate p-values. By taking the numerical results for $F(0)$, U_s, $E[C]$, $E[T_1]$ and $E[T_2]$ with $\lambda^{(s)}=8$ and $c=14$ as the initial results, we make comparisons between the distribution of the initial results and that of the numerical results with increased $\lambda^{(s)}$ and c, respectively. The p-values are shown in Table 6.

As can be seen from Table 6, the p-values for all the performance measures gradually decrease as $\lambda^{(s)}$ and c increase. This indicates that the difference between the changed numerical results of each performance measure and the initial results becomes more and more significant as $\lambda^{(s)}$ and c increase. The arrival rate $\lambda^{(s)}$ of tasks has a greater impact on each performance measure than the service rate c of a task on the MEC server, due to the fact that an increase in $\lambda^{(s)}$ leads to a simultaneous increase in the arrival rate of tasks generated on both the MD_1 and MD_2.

The calculated p-values of $F(0)$ and U_s are equal to each other, as shown in Table 6. According to Eqn. (38), it can be seen that there is a linear relationship between $F(0)$ and U_s. As $\lambda^{(s)}$ and c increase, $F(0)$ and U_s change at the same rate but in opposite directions. The p-values for $E[C]$, $E[T_1]$ and $E[T_2]$ give the same results. It can be seen from Figs. 5 and 6 that all of $E[C]$, $E[T_1]$ and $E[T_2]$ show the same exponential trend.

7. Conclusions

In this paper, we evaluated the probabilistic offloading strategy in a MEC system composed of K $(1 \le K < \infty)$ MDs sharing one MEC server. In a MEC system, the size of an individual task is usually very small compared with the capacity of the edge buffer. Therefore, for the tasks offloaded to the edge, we were not concerned with the

Table 6. p-Values calculated by the Wilcoxon rank-sum test.

	$\lambda^{(s)}=9$	$\lambda^{(s)}=10$	$\lambda^{(s)}=11$	$\lambda^{(s)}=12$
$F(0)$	0.0754 *	0.0079 ***	0.0040 ***	0.0040 ***
U_s	0.0754 *	0.0079 ***	0.0040 ***	0.0040 ***
$E[C]$	0.1111	0.0278 **	0.0079 ***	0.0040 ***
$E[T_1]$	0.1111	0.0278 **	0.0079 ***	0.0040 ***
$E[T_2]$	0.1111	0.0278 **	0.0079 ***	0.0040 ***
	$c=15$	$c=16$	$c=17$	$c=18$
$F(0)$	0.3452	0.1111	0.0754 *	0.0278 **
U_s	0.3452	0.1111	0.0754 *	0.0278 **
$E[C]$	0.3452	0.1111	0.0278 **	0.0079 ***
$E[T_1]$	0.3452	0.1111	0.0278 **	0.0079 ***
$E[T_2]$	0.3452	0.1111	0.0278 **	0.0079 ***

*** p-value<0.01, ** p-value<0.05, * p-value<0.1

processing of an individual task, but with the processing of the task flow, and regarded the task flow as a continuous fluid. By extending the single background BDP to K background BDPs, we proposed a modeling method to capture the stochastic behavior of tasks and an analysis approach to derive the multi-source fluid queue driven by K independent heterogeneous BDPs. We presented the CDF of the edge buffer content, and evaluated the performance measures of the probabilistic offloading strategy in the MEC system in terms of the utilization of the MEC server, the expected edge buffer content and the average response time of a task. As a special case, we considered a MEC system composed of two MDs and carried out numerical experiments accompanied by an analysis. With numerical results, we investigated the impact of the arrival rate of tasks on the performance of the MEC system with different service rates of a task on the MEC server.

In this paper, by considering a single MEC server, we proposed a modeling approach for the probabilistic offloading strategy based on a multi-source fluid queue. We assumed that all tasks are offloaded to a single physical node for processing, disregarding the distributed nature of the MEC model. This is a limitation of our research. As a future research direction, we will consider a MEC system based on cloud-edge collaboration to investigate task offloading strategies. In addition, we will model a distributed MEC system by considering multiple MEC servers and multiple MDs.

Acknowledgment

This work was supported in part by the National Natural Science Foundation (nos. 61872311, 61973261), China.

References

Anick, D., Mitra, D. and Sondhi, M. (1982). Stochastic theory of a data-handling system with multiple sources, *Bell System Technical Journal* **61**(8): 1871–1894.

Arunachalam, V., Gupta, V. and Dharmaraja, S. (2010). A fluid queue modulated by two independent birth-death processes, *Computers and Mathematics with Applications* **60**(8): 2433–2444.

Bai, T., Pan, C., Deng, Y., Elkashlan, M., Nallanathan, A. and Hanzo, L. (2020). Latency minimization for intelligent reflecting surface aided mobile edge computing, *IEEE Journal on Selected Areas in Communications* **38**(11): 2666–2682.

Bista, B., Wang, J. and Takata, T. (2020). Probabilistic computation offloading for mobile edge computing in dynamic network environment, *Internet of Things* **11**, Article no. 100225.

Cardellini, V., Personé, V., Valerio, V., Facchinei, F., Grassi, V., Presti, F. and Piccialli, V. (2016). A game-theoretic approach to computation offloading in mobile cloud computing, *Mathematical Programming* **157**(2): 421–449.

El-Baz, A., Tarabia, A. and Darwiesh, A. (2020). Cloud storage facility as a fluid queue controlled by Markovian queue, *Probability in the Engineering and Informational Sciences*: 1–17, DOI: 10.1017/S0269964820000613.

Elwalid, A. and Mitra, D. (1995). Analysis, approximations and admission control of a multi-service multiplexing system with priorities, *Proceedings of International Conference on Computer Communications, INFOCOM 1995, Boston, USA*, pp. 463–472.

Fiedler, M. and Voos, H. (2000). New results on the numerical stability of the stochastic fluid flow model analysis, *Proceedings of the Networking 2000 Conference, Paris, France*, pp. 446–457.

Goścień, R. and Walkowiak, K. (2017). A column generation technique for routing and spectrum allocation in cloud-ready survivable elastic optical networks, *International Journal of Applied Mathematics and Computer Science* **27**(3): 591–603, DOI: 10.1515/amcs-2017-0042.

Hassan, M., Qi, W. and Chen, S. (2015). ELICIT: Efficiently identify computation-intensive tasks in mobile applications

for offloading, *Proceedings of IEEE International Conference on Networking, Architecture and Storage, NAS 2015, Boston, USA*, pp. 12–22.

Kim, J. and Krunz, M. (2000). Bandwidth allocation in wireless networks with guaranteed packet-loss performance, *Mathematical Programming* **8**(3): 337–349.

Kulkarni, V. (1997). *Fluid Models for Single Buffer Systems*, CRC Press, Boca Raton.

Lenin, R. and Parthasarathy, P. (2000). Fluid queues driven by an M/M/1/N queue, *Mathematical Problems in Engineering* **6**(5): 439–460.

Li, K. (2019). How to stabilize a competitive mobile edge computing environment: A game theoretic approach, *IEEE Access* **7**: 69960–69985.

Li, W. and Jin, S. (2021). Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity, *Journal of Supercomputing* **77**(11): 1286–12507, DOI: 10.1007/S11227-021-03781-W.

Lim, W., Luong, N., Hoang, D., Jiao, Y., Liang, Y., Yang, Q., Niyato, D. and Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey, *IEEE Communications Surveys and Tutorials* **22**(3): 2031–2063.

Liu, Y., Peng, M., Shou, G., Chen, Y. and Chen, S. (2020). Toward edge intelligence: Multi-access edge computing for 5G and internet of things, *IEEE Internet of Things Journal* **7**(8): 6722–6747.

Mao, B., wang, F. and Tian, N. (2012). Fluid model driven by an M/M/1 queue with multiple vacations and N-policy, *Journal of Applied Mathematics and Computing* **38**(1): 119–131.

Mitra, D. (1988). Stochastic theory of a fluid model of producers and consumers coupled by a buffer, *Advances in Applied Probability* **20**(1): 646–676.

Mukherjee, M., Kumar, V., Kumar, S., Matamy, R., Mavromoustakis, C., Zhang, Q., Shojafar, M. and Mastorakis, G. (2020). Computation offloading strategy in heterogeneous fog computing with energy and delay constraints, *Proceedings of IEEE International Conference on Communications, ICC 2020, Dublin, Ireland*, pp. 1–5.

Nouri, N., Abouei, J., Jaseemuddin, M. and Anpalagan, A. (2020). Joint access and resource allocation in ultradense mmWave NOMA networks with mobile edge computing, *IEEE Internet of Things Journal* **7**(2): 1531–1547.

Razaque, A., Aloqaily, M., Almiani, M., Jararweh, Y. and Srivastava, G. (2021). Efficient and reliable forensics using intelligent edge computing, *Future Generation Computer Systems* **118**: 230–239, DOI: 10.1016/j.future.2021.01.012.

Sericola, B., Parthasarathy, P. and Vijayashree, K. (2005). Exact transient solution of an M/M/1 driven fluid queue, *International Journal of Computer Mathematics* **82**(6): 659–671.

Song, F., Ai, Z., Zhang, H., You, I. and Li, S. (2021). Smart collaborative balancing for dependable network components in cyber-physical systems, *IEEE Transactions on Industrial Informatics* **17**(10): 6916–6924.

Virtamo, J. and Norros, I. (1994). Fluid queue driven by an M/M/1 queue, *Queueing Systems* **16**(3): 373–386.

Wu, H., Sun, Y. and Wolter, K. (2020). Energy-efficient decision making for mobile cloud offloading, *IEEE Transactions on Cloud Computing* **8**(2): 570–584.

Xu, X., Shen, B., Ding, S., Srivstava, G., Bilal, M., Khosravi, M., Menon, V., Jan, M. and Wang, M. (2020). Service offloading with deep Q-network for digital twinning empowered internet of vehicles in edge computing, *IEEE Transactions on Industrial Informatics* **18**(2): 1414–1423, DOI: 10.1109/TII.2020.3040180.

Zeifman, A., Razumchik, R., Satin, Y., Kiseleva, K., Korotysheva, A. and Korolev, V. (2018). Bounds on the rate of convergence for one class of inhomogeneous Markovian queueing models with possible batch arrivals and services, *International Journal of Applied Mathematics and Computer Science* **28**(1): 141–154, DOI: 10.2478/amcs-2018-0011.

Zeifman, A., Satin, Y., Kryukova, A., Razumchik, R., Kiseleva, K. and Shilova, G. (2020). On three methods for bounding the rate of convergence for some continuous-time Markov chains, *International Journal of Applied Mathematics and Computer Science* **30**(2): 251–266, DOI: 10.34768/amcs-2020-0020.

Zhao, T., Zhou, S., Guo, X. and Niu, Z. (2017). Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing, *Proceedings of IEEE International Conference on Communications, ICC 2017, Paris, France*, pp. 1–7.

Huan Zheng received her BS degree from Xihua University, Chengdu, Sichuan, China, in 2019. She is currently pursuing her MS degree with the School of Information Science and Engineering, Yanshan University. Her research interests include mobile edge computing, performance evaluation and queueing theory.

Shunfu Jin received her BEng degree in computer and application from North East Heavy Machinery College, Qiqihaer, China, and an MEng degree in computer science as well as a PhD in circuits and systems from Yanshan University, Qinhuangdao, China. She is currently a professor at the School of Information Science and Engineering, Yanshan University. Her research interests include stochastic modeling for telecommunication, performance evaluation of communication networks and queueing systems.

Received: 2 August 2021
Revised: 25 November 2021
Re-revised: 24 December 2021
Accepted: 28 December 2021