AN EFFICIENT METHOD TO COMPUTE THE RATE MATRIX FOR MULTI-SERVER RETRIAL QUEUES WITH CLOUD COMPUTING SYSTEMS

Dang Thanh Chuong[1], Hoa Ly Cuong[1], Hoang Dinh Long[2]
and Duong Duc Hung[3]

[1]Faculty of Information Technology, University of Sciences, Hue University, Vietnam
[2]University of Education, Hue University, 32-34 Le Loi St., Hue, Vietnam
[3]Hue University, 03 Le Loi St., Hue, Vietnam

ABSTRACT

This study presents the usage of retrial queues with cloud computing systems in which the operating unit (the server) and the storing unit (buffer) are independently considered. In fact, the tasks cannot occupy the server to the system. Instead, they are stored in the buffer and sent back to the server after a random time. Upon a service completion, the server does not always get to work while waiting for a new task or a task from the buffer. After the idle time, the server instantly starts searching for a task from the buffer. The analysis model proposed in this study refers to a retrial queue system searching for tasks from theorbit with limited size under a multi-server context, and the model is modelized into the 3-dimension Markov chain. The solution is based on building an algorithm under the analytical methodology of the quasi birth-death (QBD) process that utilizes the Q-matrix to calculate the probability of states toward the proposed model.

KEYWORDS

3-Dimension Markov Chain, Cellular Mobile Networks, Cloud Computing, Quasi Birth-Death (QBD) Process, Retrial Queueing.

1. INTRODUCTION

Retrial queueing requently occurs in all aspects of our life. The characteristic of the retrial is that customers are relocated to an orbit, a queueing for retrial customers that continually repeat demand for services after failure [1]. Numerous articles dealing with retrial queueing accept that the servers share either fresh calls or repeated calls from an orbit. Phung-Duc *et al*. proposed the models with two-way communication [2]-[5]. Innovated servers, in some of other cases, are to search blocked calls to serve [6]-[13]. It is assumed that after serving a customer, the server is vacant in a given interval and searches for the blocked customer later. In idle time, if a fresh or repeated call arrives, it will be served immediately. The servers seek a retrial customer with exponential distribution after idle time. At that moment, they are unable to serve other customers. This means that if a new customer arrives, they will be moved to the orbit. After the search time, the customer successfully searched is served. Otherwise, the servers remain idle [12],[13].

The fact is that, cloud computing is available for resources, particularly the capability of storage and computing without user's management. This term is used to represent the available data center for users. Big data clouds allocate resources to different directions from the data centre. If users are closely connected, the servers are assigned. The cloud restricted to an organization is

called private cloud, while the cloud available for organizations is called public cloud. Cloud computing shares resources to maintain the coherence and economies of scale.

The proposed model will be used to analyze retrial queueing in cloud computing platforms with separate processing and storage units[14]-[17]. The processing unit serves only one customer at a specific time. The freshly arriving calls are stored in a buffer if all servers are busy. When finishing serving a customer, a server remains idle for a while, waiting to select a particular customer from the buffer. The idle interval is called the search time. The whole is modelled by utilizing a retrial queueing for searching calls that follow a solution [1].

There were research works involving retrial queueing. Artalejo et al. [6] investigated the retrial queueing by searching customers in the orbit. Specifically, after serving a customer, the server promptly selects another customer from the orbit with the probability p or remains idle with the probability $1-p$. This is similar to the model in which the server picks a customer from the orbit. In the model, there is, nevertheless, no idle and searching time.

In [9], the authors considered the retrial queueing $BMAP/G/1$ with customers that arrive following the BMAP process. Individuals who arrive following the batch when all servers are busy proceed to the orbit. While the customers that access the system after the batch are promptly served, the rest is transferred to the orbit. The customers from the orbit repeatedly attempt to attain success with the inter-arrival time following the exponential distribution based on the number of customers in the orbit. On the other hand, the mechanism of searching clients can be activated at the moment the server just concluded a customer with the given probability, depending on the number of customers in the orbit.

The searching time is stochastic and seems to depend on the number of customers in the orbit. The customer picked after the search process is immediately served if some of the servers are vacant [4]. It is assumed that the service time follows the general distribution regardless of retrial customers from the orbit. The notation of the general distribution is G. Artalejo and Phung-Duc [4],[5] examined the model with two cases in which, after idle time, the server handles outcoming calls with the inter-arrival following the exponential distribution. This is regarded as the search time in the model of this article. After an outcoming call, the server is, notwithstanding, available. For example, there is no customer to pick out. Other articles reported that the retrial rate and the number of customers in the orbit are linear functions [18], [19].

In this paper, we model the mechanism that operates after the server's free time [12]. This means that the mechanism that searches for a customer is immediately initiated after the server's vacant duration. Accordingly, the model in which the platform of cloud computing employs the retrial queueing, is considered where the processing and storage unit are separated. The researches in [22]-[26] mainly base on a 2-dimensional Markov to build models.

The main result of the paper is proposed the multi-server with the orbit's limited buffer in cloud computing systems. The model applies a 3-dimensional Markov chain combined with the quasi birth-death process to work out possibility of blocking. We have also built an algorithm for computing the blocking probability of the system based on the construction of the 3-dimensional infinitesimal generator matrix Q to compute steady-state probabilities corresponding to the quasi birth-death process of the proposed model in the article. The disparity is that the number of the servers that are serving a repeated customer or a fresh customer and the number of the servers that are searching for a customer in the orbit are considered.

The organization of the article is as follows, In Section 2, the detailed problems with the proposed model and an algorithm for computing the blocking probability. The analysis results

will be presented evaluated the model performances in Section 3 and the conclusion is presented in Section 4.

2. ANALYSIS MODEL FOR MULTI-SERVER CLOUD COMPUTING SYSTEM

2.1. Problems

We concentrate our attention on the retrial queueing for cloud computing, in which the processing units (the servers) and the storage units (the buffers) are separately considered as in [12]. Accordingly, a processing unit is capable of serving one and only one customer at a particular moment. Thus, a call arriving while all servers are active is stored in a queueing (a buffer), and by then, it reattempts to be served. Successfully executing a mission, the server is accessible, and the processing unit seeks a job from the queueing within a specified interval. Such periods are commonly called search time. In earlier works [6],[9], the idle time and search time are discrete. As a result, the computational model represents the mechanism that the search time is allowed after the server's idle time [12]. The system employs retrial queueing with searching customers. Compared with the model in [13], the innovation point is the extension with the multi-server.

2.2. The model and Parameters

2.2.1. Some assumptions of the Model

The analytical method is based on the followings:

- The customers to be served arrive to the server with the average rate λ.
- The service time for arriving customers follows the exponential distribution with the average time $1/\nu_1$, or in other words, the average rate is ν_1.
- After completing a task, the server remains vacant with the interval following the exponential distribution with the average time $\frac{1}{\alpha}$.
- During idle time, a client, regardless of repeated clients or fresh clients, entering the service, is immediately served.
- Thereafter, the server continues seeking a customer in the orbit. Also, the search time follows the exponential distribution $1/\nu_2$, or the average rate ν_2.
- Arriving while all servers are busy, a customer is transferred to the orbit in order to be returned in the exponential average time $1/\mu$, or the average rate μ.

2.2.2. The analytical method for the Single server

The analytical method is similar to the model in [12], but the main difference is that the orbit size in this model is limited ($0 \le N(t) \le L$).

Let $S(t)(t \ge 0)$ denote the server state at the moment t. Accordingly, we also define the server states as [1]:

$$S(t) = \begin{cases} 0, & \text{the server is available,} \\ 1, & \text{the server is serving a customer,} \\ 2, & \text{the server is looking for a customer from the orbit.} \end{cases}$$

Let $N(t)(t \ge 0)$ denotes the number of customers in the orbit at the moment $t \ge 0$. Then, $\{X(t) = (S(t), N(t)), t \ge 0\}$ forms the Markov chain in the state space $\mathcal{S} = \{0,1,2\} \times \{0,1,2,\dots,L\}$. The state transition diagram is shown in Figure 1. We assume that the system is stable, which means that the steady-state probabilities subsist. The necessary and sufficient condition for the stable system is $\lambda < \nu_1$ that will be used in the following analyses [12].

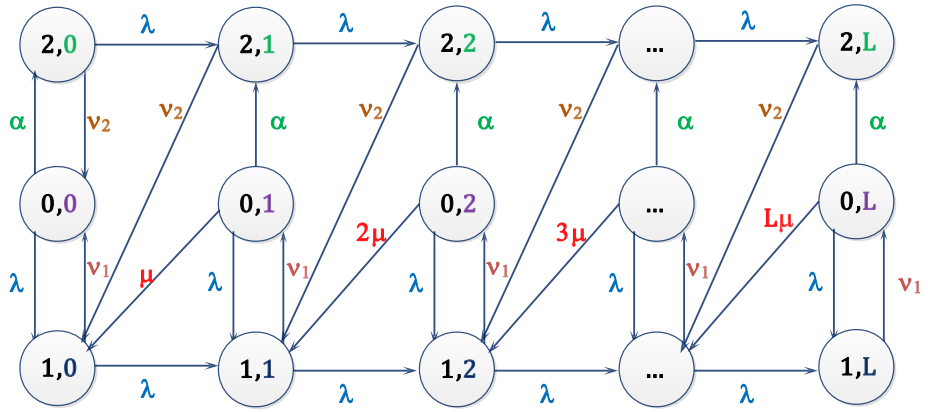

Figure 1.The state transition diagram for the single server

Let $\pi_{i,j} = P[S(t) = i, N(t) = j]$ are the balance probabilities at the state (i, j). The state transition matrices A_j, B_j and C_j, 3×3 matrixes, represent the steps in Figure 1 [1]:

(a). $A_j(i,k)$ denotes the transition rate from the state (i,j) to the state (k,j) $(0 \le j \le L; 0 \le i,k \le 2)$ which is caused by the fact that the server is idle, serving a client or seeking for a customer from the orbit (in case of A_0). The 3×3 matrix A_j with entries $A_j(i,k)$ is written as

$$A_0 = \begin{pmatrix} 0 & \lambda & \alpha \\ \nu_1 & 0 & 0 \\ \nu_2 & 0 & 0 \end{pmatrix}, A_j = \begin{pmatrix} 0 & \lambda & \alpha \\ \nu_1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, (1 \le j \le L).$$

(b). $B_j(i,k)$ denotes the transition rate from the state (i,j) to the state $(k,j+1)$ $(0 \le j \le L-1; 0 \le i,k \le 2)$ which is caused by a rejected demand due to the fact that the server is serving or is searching for a customer. The 3×3 matrix B_j (or B) with entries $B_j(i,k)$ is written as

$$B_j = B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \lambda & \lambda \\ 0 & 0 & 0 \end{pmatrix}, (0 \le j \le L-1)$$

(c). $C_j(i,k)$ denotes the transition rate from the state (i,j) to the state $(k,j-1)$ $(1 \le j \le L; 0 \le i,k \le 2)$ which is caused by the fact that a customer returning from the orbit is served by the idle server, or a client in the orbit is successfully searched. The 3×3 matrix C_j with entries $C_j(i,k)$ is written as.

The infinitesimal generator matrix Q is given by:

$$Q = \begin{pmatrix} Q_0 & B_0 & O & O & O & O \\ C_1 & Q_1 & B_1 & O & O & O \\ O & C_2 & Q_2 & \ddots & O & O \\ O & O & C_3 & \ddots & B_{L-2} & O \\ O & O & O & \ddots & Q_{L-1} & B_{L-1} \\ O & O & O & O & C_L & Q_L \end{pmatrix}$$

where$Q_j = A_j - D^{A_j} - D^B - D^{C_j}$, $(1 \le j \le L-1)$, trong đó:

$$D^{T_j} = \begin{pmatrix} \sum_{k=0}^{2} T_j(0,k) & 0 & 0 \\ 0 & \sum_{k=0}^{2} T_j(1,k) & 0 \\ 0 & 0 & \sum_{k=0}^{2} T_j(2,k) \end{pmatrix}, (T = A, B, C).$$

Note that: $Q_0 = A_0 - D^{A_0} - D^B$ và $Q_L = A_L - D^{A_L} - D^{C_L}$.

Let $\pi_j = (\pi_{0,j}, \pi_{1,j}, \pi_{2,j})$ are the level probability vectors. We obtain the set of balance equations as follows:

$$\pi_0 Q_0 + \pi_1 C_1 = (0,0,0) \quad (1)$$

$$\pi_j B_j + \pi_{j+1} Q_{j+1} + \pi_{j+2} C_{j+2} = (0,0,0),\ (0 \le j \le L-2) \quad (2)$$

$$\pi_{L-1} B_{L-1} + \pi_L Q_L = (0,0,0) \quad (3)$$

The values π_j are computed by solving the set of the equations (1)-(3) by applying the quasi birth-death process according to the infinitesimal generator matrix Q [20] and [21]. The analysis results are shown after modelling the problem for the multi-server.

2.2.3. The analytical model for the multi-server system

This study proposes an improved model of the single server described above with a limited-size orbit multi-server. Thus, the analytical method utilizes the retrial queueing model $M/M/c/L\ (c > 1, L > 1)$ [27] with the aforementioned parameters in which each of the servers is regarded as a single server [1-3, 7].

Let $S_1(t)$ and $S_2(t)$ denote the number of the servers that are serving a repeated customer or a fresh customer and those searching for a customer in the orbit, and $N(t)$ connotes the number of customers in the orbit at the moment t. It is evident that $X(t) = \{S_1(t), S_2(t), N(t); t \ge 0\}$ generates the state space $\mathcal{S} = \{(i,j,k); i = \overline{0,c}, j = \overline{0,c-i}, k = \overline{0,L}\}$.

Thenceforth, the infinitesimal generator matrix Q is a 3-dimensional matrix as in [7][13] (Figure 2):

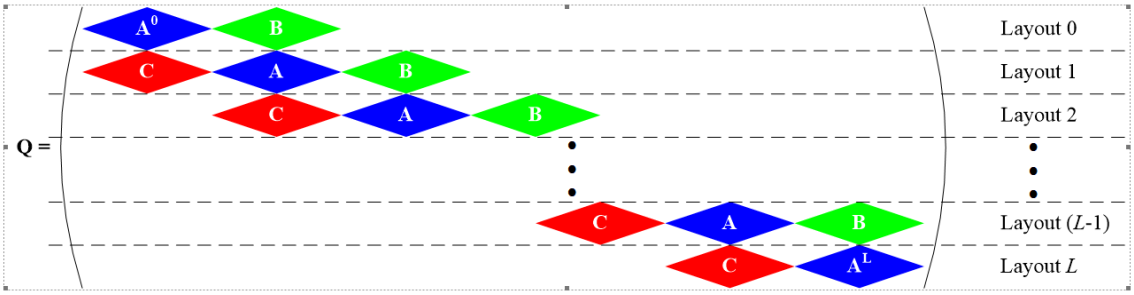

Figure 2. The infinitesimal generator matrix Q is a 3-dimensional matrix

The matrices $A_k (0 \le k \le L)$, B and $C_k (1 \le k \le L)$ are the $\frac{(c+1)(c+2)}{2} \times \frac{(c+1)(c+2)}{2}$ matrices. We consider the indexes (i,j,k) corresponding to state transition steps as follows:

1) The values $k\ (0 \le k \le L)$ are retained (the state transition rates from (i,j,k) to (i',j',k)). The state transition matrices are A_0, A_k and A_L. It is required to compute (i,j) within the range $\big((i+j) \le c;\ i \ge 0, j \ge 0\big)(*)$. The number of pairs (i,j) fulfilling $(*)$ is $C_{c+2}^2 = \frac{(c+2)(c+1)}{2}$. Also, it is the size of rows (or columns) of A_0, A_k and A_L. They contain entries following the indices of rows (or columns) as in Table 1.

Table 1.The indexes of rows (or columns) of submatrices.

Indexes$(i,0)$	$(0,0)$	$(1,0)$	$(2,0)$	$\cdots$	$(c-1,0)$	$(c,0)$
Level0	0	1	2		$c-1$	c
Indexes$(i,1)$	$(0,1)$	$(1,1)$	$(2,1)$	$\cdots$	$(c-1,1)$	
Level1	$c+1$	$c+2$	$c+3$		$2c$	
Indexes$(i,2)$	$(0,2)$	$(1,2)$	$\cdots$			
Level2	$2c+1$	$2c+2$				
$\vdots$	$\vdots$	$\vdots$				
Indexes $(i,c-1)$	$(0,c-1)$	$(1,c-1)$				
Level$(c-1)$	$\frac{(c+2)(c+1)}{2}-3$	$\frac{(c+2)(c+1)}{2}-2$				
Indexes (i,c)	$(0,c)$					
Level$c$	$\frac{(c+2)(c+1)}{2}-1$					

By indexing as mentioned, we compute the general indices (i,j) corresponding to $\frac{(c+2)(c+1)}{2}-\frac{(c+1-j)(c+2-j)}{2}+i=\frac{j(2c-j+3)}{2}+i$.

✓ The matrix A_0 with its non-zero entries is computed as follows:

○ The state transition rates from $(i,j,0)$ to $(i,j+1,0),(i+j\leq c-1)$ correspond to the event that a server restores the searching state.

$$A_0\left(\frac{j(2c-j+3)}{2}+i,\frac{(j+1)(2c-j+2)}{2}+i,0\right)=j\alpha,\ (i+j\leq c-1),$$

○ The state transition rates from $(i,j,0)$ to $(i,j-1,0),(i+j\leq c)$ correspond to the event that a searching server becomes idle.

$$A_0\left(\frac{j(2c-j+3)}{2}+i,\frac{(j-1)(2c-j+4)}{2}+i,0\right)=j\nu_2,\ (i+j\leq c),$$

○ The state transition rates from $(i,j,0)$ to $(i+1,j,0),(i+j\leq c-1)$ correspond to the event that a server enters the searching state because a fresh customer is immediately served by an available server.

$$A_0\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i+1,0\right)=\lambda,\ (i+j\leq c-1),$$

○ The state transition rates from $(i,j,0)$ to $(i-1,j,0),(i+j\leq c)$ correspond to the event that a server has just serviced a customer.

$$A_0\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i-1,0\right)=i\nu_1,\ (i+j\leq c),$$

The elements on the main diagonal of matrix A_0 are:

$$A_0\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,0\right)=-2\lambda-j\alpha-j\nu_2-i\nu_1,\ (i+j\leq c-1),$$

$$A_0\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,0\right)=-j\nu_2-i\nu_1,\ (i+j=c).$$

✓ The matrix $A_k(1 \le k \le L-1)$ and A_L with their non-zero entries are computed as follows:

○ The state transition rates from (i,j,k) to $(i,j+1,k),(i+j \le c-1)$ correspond to the event that one more server restores the searching state.

$$A_k\left(\frac{j(2c-j+3)}{2}+i,\frac{(j+1)(2c-j+2)}{2}+i,k\right)=A_L\left(\frac{j(2c-j+3)}{2}+i,\frac{(j+1)(2c-j+2)}{2}+i,L\right)=j\alpha,\ (i+j\le c-1,1\le k\le L-1),$$

○ The state transition rates from (i,j,k) to $(i+1,j,k),(i+j \le c-1)$ correspond to the event that a server has entered the serving state while a fresh customer is immediately served by an available server.

$$A_k\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i+1,k\right)=A_L\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i+1,L\right)=\lambda,\ (i+j\le c-1,1\le k\le L-1),$$

○ The state transition rates from (i,j,k) to $(i-1,j,k),(i+j \le c)$ correspond to the event that a server just finished serving a customer.

$$A_k\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i-1,k\right)=A_L\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i-1,L\right)=i\nu_1,\ (i+j\le c,1\le k\le L-1),$$

The elements on the main diagonal of the matrices $A_k(1 \le k \le L-1)$ and A_L are:

$$A_k\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,,k\right)=-2\lambda-j\alpha-j\nu_2-i\nu_1-k\mu,\ (i+j\le c-1,1\le k\le L-1),$$

$$A_k\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,k\right)=-j\nu_2-i\nu_1,\ (i+j=c,1\le k\le L-1),$$

$$A_L\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,L\right)=-\lambda-j\alpha-j\nu_2-i\nu_1-k\mu,\ (i+j\le c-1),$$

$$A_L\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,L\right)=-j\nu_2-i\nu_1,\ (i+j=c).$$

2) The values $k\ (0 \le k \le L-1)$ increase by one unit (the state transition rates from (i,j,k) to $(i',j',k+1)$). The state transition matrices are B_k (or B). The size of B is the same as that of A_0, A_{k} and A_L. Their non-zero elements are referred hereafter.

○ The state transition rates from (i,j,k) to $(i,j,k+1),(i+j=c,0\le k\le L-1)$ correspond to the event that a fresh customer perceives that all servers are busy, and she is transferred to the orbit.

$$B\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i,k\right)=\lambda,\ (i+j\le c-1,0\le k\le L-1).$$

3) The values $k\ (1 \le k \le L)$ decrease by one unit (the state transition rates from (i,j,k) to $(i',j',k-1)$). The state transition matrices are C_k. The size of C is the same as that of A_0, A_{k}, A_L and B. Their non-zero elements are as follows:

○ The state transition rates from (i,j,k) to $(i+1,j,k-1),(i+j\le c-1,1\le k\le L)$ correspond to the event that a repeated customer returns to be immediately served by an available server.

$$C_k\left(\frac{j(2c-j+3)}{2}+i,\frac{j(2c-j+3)}{2}+i+1,k\right)=k\mu,\ (i+j\le c-1,1\le k\le L)$$

○ The state transition rates from (i,j,k) to $(i+1,j-1,k-1),(i+j\le c,1\le k\le L)$ correspond to the event that a customer in the orbit is successfully served by the searching server.

$$C_k\left(\frac{j(2c-j+3)}{2}+i,\frac{(j-1)(2c-j+4)}{2}+i+1,k\right)=j\nu_2,\ (i+j\le c,1\le k\le L).$$

Let$\pi_{i,j,k}=\lim_{t\to\infty}P[S_1(t)=i,S_2(t)=j]$where$(i,j,k)\in\mathcal{V}^k$ $(\mathcal{V}^k=\{(i,j,k);i=\overline{0,c},j=\overline{0,c-i},k=\overline{0,L}\})and\pi_i^k=\left(\pi_{i,0,k},\pi_{i,1,k},\dots,\pi_{i,c-i,k}\right),\ \left(i=\overline{0,c},k=\overline{0,L}\right)$. Let$\pi^k=\left(\pi_0^k,\pi_1^k,\dots,\pi_c^k\right),\ \left(k=\overline{0,L}\right)$denotes the stationary distributions of $C(t)$ $(C(t)=\{S_1(t),S_2(t);t\ge 0\})$, $(t\ge 0)$, which is the unique solution of the set of equations:

$$\pi^0 A_0 + \pi^1 B = \underbrace{(0,0,\dots,0)}_{\frac{(c+1)\times(c+2)}{2}} \tag{4}$$

$$\pi^{k-1} C_k + \pi^k A_k + \pi^{k+1} B = \underbrace{(0,0,\dots,0)}_{\frac{(c+1)\times(c+2)}{2}}, (1 \le k \le L-1) \tag{5}$$

$$\pi^{L-1} C_L + \pi^L A_L = \underbrace{(0,0,\dots,0)}_{\frac{(c+1)\times(c+2)}{2}} \tag{6}$$

$$\sum_{i=0}^{c} \sum_{j=0}^{c-i} \sum_{k=0}^{L} \pi_{i,j,k} = 1 \tag{7}$$

The blocking probability is determined as follows:

$$PB = \sum_{i=0}^{c} \pi_{i,c-i,L} \tag{8}$$

From the analyses above, we propose an algorithm for computing the blocking probability as follows:

Algorithm 1:

Input: The state space S.

Output: The probability PB (from equation (8)).

Method:

Step 1: Compute Q

- Step 1.1: Formulate the state transition matrices $A_k (0 \le k \le L)$, B and $C_k (1 \le k \le L)$ as follows:

$$A_k\left(\frac{j(2c-j+3)}{2} + i, \frac{(j+1)(2c-j+2)}{2} + i, k\right) = j\alpha, (i + j \le c - 1, 0 \le k \le L), \tag{9}$$

$$A_k\left(\frac{j(2c-j+3)}{2} + i, \frac{(j-1)(2c-j+4)}{2} + i, 0\right) = j\nu_2, (i + j \le c), \tag{10}$$

$$A_k\left(\frac{j(2c-j+3)}{2} + i, \frac{j(2c-j+3)}{2} + i + 1, k\right) = \lambda, (i + j \le c - 1, 0 \le k \le L), \tag{11}$$

$$A_k\left(\frac{j(2c-j+3)}{2} + i, \frac{j(2c-j+3)}{2} + i - 1, k\right) = i\nu_1, (i + j \le c, 0 \le k \le L), \tag{12}$$

$$B\left(\frac{j(2c-j+3)}{2} + i, \frac{j(2c-j+3)}{2} + i, k\right) = \lambda, (i + j \le c - 1, 0 \le k \le L - 1), \tag{13}$$

$$C_k\left(\frac{j(2c-j+3)}{2} + i, \frac{j(2c-j+3)}{2} + i + 1, k\right) = k\mu, (i + j \le c - 1, 1 \le k \le L), \tag{14}$$

$$C_k\left(\frac{j(2c-j+3)}{2} + i, \frac{(j-1)(2c-j+4)}{2} + i + 1, k\right) = j\nu_2, (i + j \le c, 1 \le k \le L). \tag{15}$$

- Step 1.2: Generate the entries on the main diagonal of the infinitesimal generator matrix Q:

$$A_k\left(\frac{j(2c-j+3)}{2} + i, \frac{j(2c-j+3)}{2} + i, k\right) = \begin{cases} -2\lambda - j\alpha - j\nu_2 - i\nu_1, (i + j \le c - 1, k = 0) \\ -2\lambda - j\alpha - j\nu_2 - i\nu_1 - k\mu, (i + j \le c - 1, 1 \le k \le L - 1), \\ -\lambda - j\alpha - j\nu_2 - i\nu_1 - k\mu, (i + j \le c - 1, k = L) \end{cases} \tag{16}$$

$$A_k\left(\frac{j(2c-j+3)}{2} + i, \frac{j(2c-j+3)}{2} + i, k\right) = -j\nu_2 - i\nu_1, (i + j = c, 0 \le k \le L). \tag{17}$$

Step 2: Transform the 3D matrix Q to the 2D matrix by adding the $k (0 \le k \le L)$ layers of Q to the matrix $\frac{(c+1)\times(c+2)\times L}{4} \times \frac{(c+1)\times(c+2)\times L}{4} Q'$ with its entries:

$Q'(i,j) = \sum_{k=0}^{L} Q(i,j,k), \left(0 \le i \le \frac{(c+1)(c+2)}{2} - 1, 0 \le j \le \frac{(c+1)(c+2)}{2} - 1\right).$

Step 3: Compute the vector v

From the normalization equation (7), it is easily deduced that $vE = e$, where $v = (\pi^0, \pi^1, \pi^2, \dots, \pi^L)$, the matrix $\frac{(c+1)\times(c+2)\times L}{4} \times \frac{(c+1)\times(c+2)\times L}{4} E$ with all entries equal to 1 and the row vector e (same size as v) with all elements being 1.
We have $vQ' + vE = v(Q' + E) = 0 + e$.
Therefore, $v = e(Q' + E)^{-1}$.
Step 4: Determine the blocking probability PB according toequation (8).
The complexity of the **Algorithm 1**is demonstrated as:

- The complexity of formulating the state transition matrices in Step 1.1 is $O(c^2L)$. The complexity of generating the entries on the main diagonal of the infinitesimal generator matrix Q in step 1.2 is $O(c^2L^2)$. The algorithmic complexity in step 1 is $O(c^4L^3)$.
- The complexity in Step 2 is $O(c^2L)$.
- In Step 3, it is $O(c^6L^3)$ to calculating the inverse matrix $(Q' + E)^{-1}$, and $O(c^6L^3)$ is for the matrix product $e(Q' + E)^{-1}$. Hence, the complexity in step 3 is $O(c^{12}L^6)$.
- The complexity in step 4 is $O(c)$

=> As a result, the time complexity of the algorithms 1 calculated with the rules of the addition is $O(c^{12}L^6)$.

2.2.4. Model illustration:

Consider the case where $c = 2$ and$L = 2$.
The state transition diagram for a multiserver is illustrated in Figure 2.

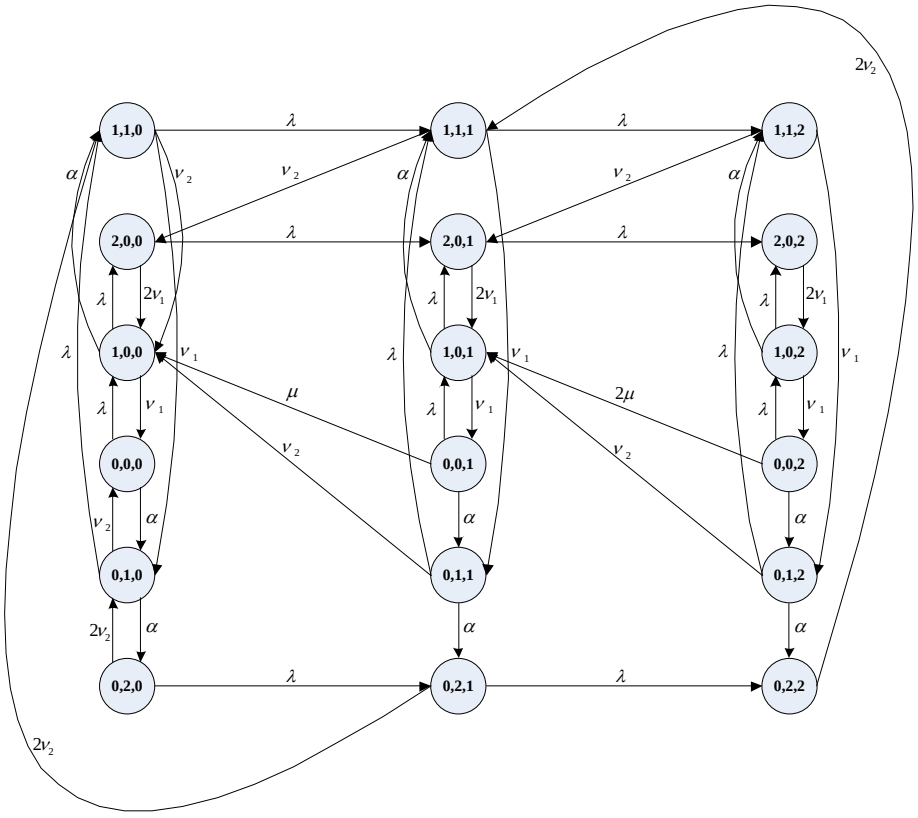

Figure 3. The state transition diagram for a multi-server where $c = 2$ and $L = 2$.

From the diagram in Figure 3, we deduce the submatrices of the matrix $\boldsymbol{Q}$ as follows:
When $c = 2$ and $L = 2$, the size of the matrices $A_k (0 \le k \le 2)$, B and $C_k (1 \le k \le 2)$ is 6×6. It implies that the size of the matrix Q is $18 \times 18 \times 3$.

$$A_0 = \begin{pmatrix} -2\lambda & \lambda & 0 & 0 & 0 & 0 \\ \nu_1 & -2\lambda - \nu_1 & \lambda & 0 & 0 & 0 \\ 0 & 2\nu_1 & -2\nu_1 & 0 & 0 & 0 \\ \nu_2 & 0 & 0 & -2\lambda - \alpha - \nu_2 & \lambda & \alpha \\ 0 & \nu_2 & 0 & \nu_1 & -\nu_1 - \nu_2 & 0 \\ 0 & 0 & 0 & 2\nu_2 & 0 & -2\nu_2 \end{pmatrix},$$

$$A_1 = \begin{pmatrix} -2\lambda - \mu & \lambda & 0 & 0 & 0 & 0 \\ \nu_1 & -2\lambda - \nu_1 - \mu & \lambda & 0 & 0 & 0 \\ 0 & 2\nu_1 & -2\nu_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\lambda - \alpha - \nu_2 - \mu & \lambda & \alpha \\ 0 & 0 & 0 & \nu_1 & -\nu_2 - \nu_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2\nu_2 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} -\lambda - 2\mu & \lambda & 0 & 0 & 0 & 0 \\ \nu_1 & -\lambda - \nu_1 - 2\mu & \lambda & 0 & 0 & 0 \\ 0 & 2\nu_1 & -2\nu_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\lambda - \alpha - \nu_2 - 2\mu & \lambda & \alpha \\ 0 & 0 & 0 & \nu_1 & -\nu_2 - \nu_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2\nu_2 \end{pmatrix},$$

$$B = \begin{pmatrix} \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{C_1} = \begin{pmatrix} \mathbf{0} & \boldsymbol{\mu} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mu} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\nu_2} & \mathbf{0} & \mathbf{0} & \boldsymbol{\mu} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\nu_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{2\nu_2} & \mathbf{0} \end{pmatrix}, \boldsymbol{C_2} = \begin{pmatrix} \mathbf{0} & \boldsymbol{2\mu} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{2\mu} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\nu_2} & \mathbf{0} & \mathbf{0} & \boldsymbol{2\mu} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\nu_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{2\nu_2} & \mathbf{0} \end{pmatrix}.$$

3. RESULTS

The efficiency of performance as the change of the parameters of the system is firstly considered with the fresh and handover calls. When the blocks occur, the fresh calls reattempt to connect in the intervals of the stochastic distribution. We assume the base station of the cell that can process c connections simultaneously. Table 2 enumerates the parameters to analyze the results[13]. The Mathematica program of Wolfram Research [28] is a powerful tool to compute and simulate network models and is utilized in our model.

Table 2.The parameters of the model.

Parameter	Value	Unit	Description
c	[5,22]	server	The number of servers
L	[5,20]	customer	The capacity of the orbit
$\frac{1}{\lambda}$	[0.1,0.9]	second	The interarrival time of the customers
$\frac{1}{\alpha}$	5	second	The idle time of server
$\frac{1}{\mu}$	$\frac{10}{3}$	second	The retrial time
$\frac{1}{\nu_1}$	$\frac{10}{3}$	second	The service time
$\frac{1}{\nu_2}$	2	second	The searching time

We simulate the model with the following parameters: $c = 5$, $\lambda = \frac{2}{5}$ and L from 5 to 20. Thus, we came to the blocking probabilities (PB) presented in Figure 4. The higher L is, the lower the blocking probabilities can be reached, which is caused by the increase of the orbit capacity.

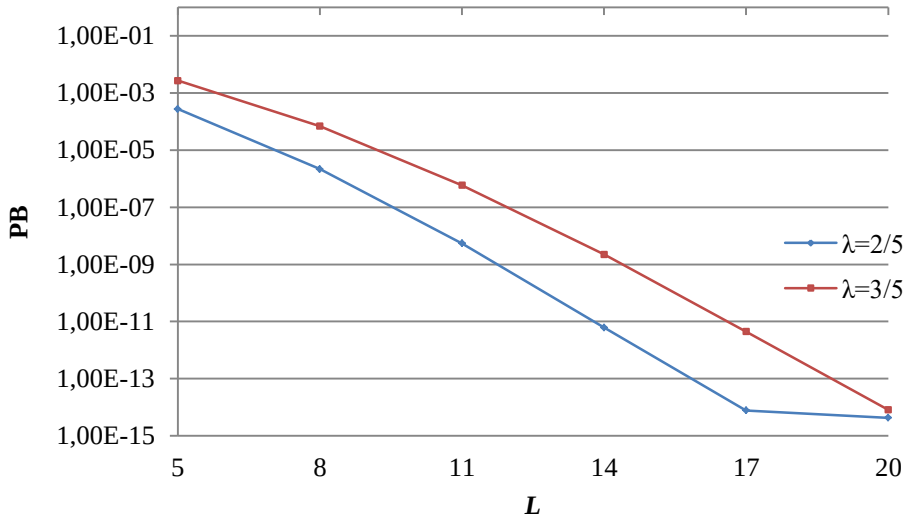

Figure 4. The blocking probabilities PB following L.

Figure 5 presents the effects when $L = 5$, $\lambda = \frac{2}{5}$, c ranges from 5 to 22. As a consequence, the higher c is, the slighter the blocking probabilities are obtained, which is appropriate for the initial assumptions because the possibility for the number of the available servers augments.

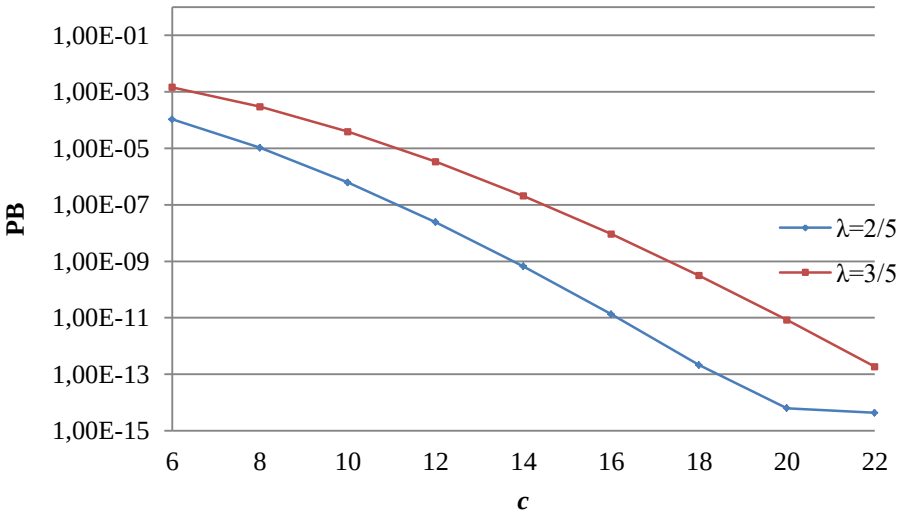

Figure 5. The blocking probabilities PB following c.

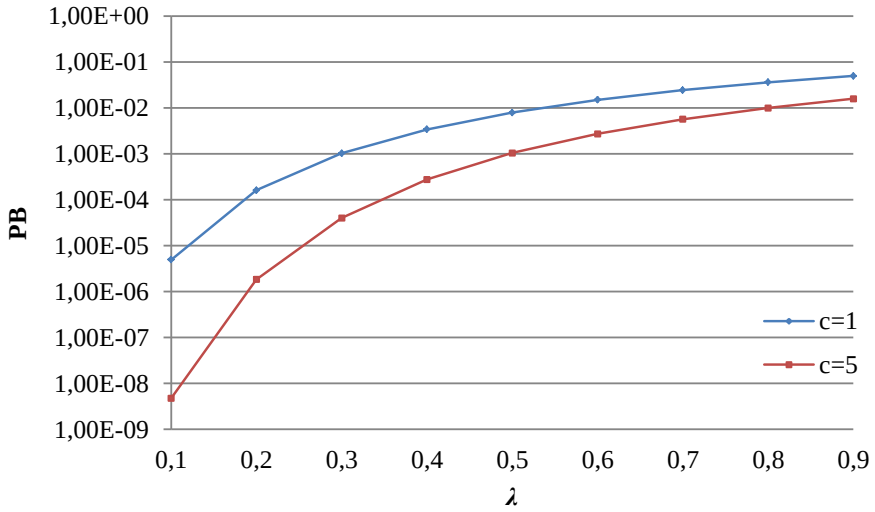

Figure 6. The blocking probabilities PB following λ.

Furthermore, to confirm the correctness of the model, we consider the case where λ fluctuates as presented in Table 1: $c = 5$ and $c = 1$ and the others are default parameters. In comparison with $c = 1$, the instance $c = 5$ provides superior results, and the consequences of $c = 1$ proximate those of $c = 5$ when λ augments (Figure 6).

4. CONCLUSIONS

The paper proposes a retrial queueing model for the single and multi-server systems in cloud computing. The results reveal that the effect of the model for the multiserver system, which reduces the blocking probabilities with the steady average flow of customers in the orbit, depends on the arrival rate λ. The model in this article is characterized by the queueing $M/M/c/L$ with the number of the serving and searching servers. The advantage of the multi-server model is that the blocking probabilities are fairly low compared to the single-server model. The article also proposes an algorithm for computing the blocking probability of the system based on the

construction of the 3-dimension infinitesimal generator matrix Q to compute steady-state probabilities corresponding to the quasi birth-death process of the proposed model.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] T. Phung-Duc, Retrial Queueing Models: A Survey on Theory and Applications, vol. abs/1906.09560, 2019.

[2] H. Sakurai and T. Phung-Duc, Scaling limits for single server retrial queues with two-way communication, Annals of Operations Research, vol. 247, no. 1, pp. 229-256, 2016.

[3] H. Sakurai and T. Phung-Duc, Two-way communication retrial queues with multiple types of outgoing calls, TOP, vol. 23, no. 2, pp. 466-492, 2015.

[4] J. R. Artalejo and T. Phung-Duc, Single server retrial queues with two-way communication, Applied Mathematical Modelling, vol. 37, no. 4, pp. 1811-1822, 2013.

[5] J, Artalejo and T. Phung-Duc, Markovian Retrial Queues with Two Way Communication, Journal of Industrial and Management Optimization, vol. 8, pp. 781, 2012.

[6] J. R. Artalejo, V. C. Joshua, and A. Krishnamoorthy, An M/G/1 retrial queue with orbital search by the server, Advances in stochastic modelling, pp. 41-54, 2002.

[7] S. R. Chakravarthy, A. Krishnamoorthy, and V. C. Joshua, Analysis of a multi-server retrial queue with search of customers from the orbit, Performance Evaluation, vol. 63, no. 8, pp. 776-798, 2006.

[8] T. G. Deepak, A. N. Dudin, V. C. Joshua, and A. Krishnamoorthy, On an M (X)/G/1 retrial system with two types of search of customers from the orbit, Stochastic Analysis and Applications, vol. 31, no. 1, pp. 92-107, 2013.

[9] A. N. Dudin, A. Krishnamoorthy, V. C. Joshua, and G. V. Tsarenkov, Analysis of the BMAP/G/1 retrial system with search of customers from the orbit, European Journal of Operational Research, vol. 157, no. 1, pp. 169-179, 2004.

[10] Krishnamoorthy, T. G. Deepak, and V. C. Joshua, An M|G|1 Retrial Queue with Nonpersistent Customers and Orbital Search, Stochastic Analysis and Applications, vol. 23, no. 5, pp. 975-997, 2005.

[11] P. Rajadurai, V. M. Chandrasekaran, and M. C. Saravanarajan, Analysis of an M[X]/G/1 unreliable retrial G-queue with orbital search and feedback under Bernoulli vacation schedule, OPSEARCH, vol. 53, no. 1, pp. 197-223, 2016.

[12] T. Phung-Duc, Retrial Queue for Cloud Systems with Separated Processing and Storage Units, Cham, 2016, pp. 143-151: Springer International Publishing.

[13] T. Phung-Duc, W. Rogiest, Y. Takahashi, and H. Bruneel, Retrial queues with balanced call blending: analysis of single-server and multiserver model, Annals of Operations Research, vol. 239, no. 2, pp. 429-449, 2016.

[14] R. Hummen, M. Henze, D. Catrein, and K. Wehrle, A Cloud design for user-controlled storage and processing of sensor data, in 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, 2012, pp. 232-240.

[15] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, Cloud Computing: Opportunities and Challenges, IEEE Potentials, vol. 33, no. 1, pp. 34-36, 2014.

[16] T. W. Wlodarczyk, Overview of Time Series Storage and Processing in a Cloud Environment, in 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, 2012, pp. 625-628.

[17] W. Itani, A. Kayssi, and A. Chehab, Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures, in 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009, pp. 711-716.

[18] J. Artalejo and I. Atencia, On the Single Server Retrial Queue with Batch Arrivals, Sankhyā: The Indian Journal of Statistics (2003-2007), vol. 66, pp. 140-158, 2004.

[19]J. R. Artalejo and A. Gomez-Corral, Steady state solution of a single-server queue with linear repeated requests, Journal of Applied Probability, vol. 34, no. 1, pp. 223-233, 1997.

[20] T. Phung-Duc, H. Masuyama, S. Kasahara, and Y. Takahashi, A simple algorithm for the rate matrices of level-dependent QBD processes, presented at the Proceedings of the 5th International Conference on Queueing Theory and Network Applications, Beijing, China, 2010.

[21] T. Phung-Duc, H. Masuyama, S. Kasahara, and Y. Takahashi, A matrix continued fraction approach to multiserver retrial queues, Annals of Operations Research, vol. 202, no. 1, pp. 161-183, 2013.

[22] V. Dragieva and T. Phung-Duc (2020), A finite-source M/G/1 retrial queue with outgoing calls, Annals of Operations Research, vol. 293, no. 1, pp. 101-121.

[23] A. A. Nazarov, S. V. Paul, O. D. Lizyura (2020), Two-way communication retrial queue with unreliable server and multiple types of outgoing calls, Mathematical Modeling, vol. 28, no. 1, pp. 49-61.

[24] A. Nazarov, T. Phung-Duc, S. Paul and O. Lizyura (2020), Diffusion approximation for multiserver retrial queue with two-way communication, Distributed Computer and Communication Networks - Lecture Notes in Computer Science, pp. 567-578.

[25] V. Vavilov (2020), Research on retrial queue with two-way communication in a diffusion environment, Applied Modeling Techniques and Data Analysis 2, pp. 233-249.

[26] A. Blaginin, I. Lapatin (2021), The two-dimensional output process of retrial queue with two-way communication, Information Technologies and Mathematical Modelling. Queueing Theory and Applications - Communications in Computer and Information Science, pp. 279-290.

[27] Dang Thanh Chuong, Hoa Ly Cuong, Pham Trung Duc, Duong Duc Hung, Performance Analysis in Cellular Networks considering the QoS by retrial queueing model with the Fractional Guard Channels Policies, International Journal of Computer Networks & Communications (IJCNC), July 2021, Volume 13, Number 4, pp. 85 – 100, DOI: 10.5121/ijcnc.2021.13406.

[28] Wolfram Mathematica, 2022. [Online]. Available: https://www.wolfram.com/mathematica/

AUTHORS

Dang Thanh Chuong obtained his doctorate in Mathematical Foundation for Computers and Computing Systems in 2014 from the Institute of Information Technology, Vietnam Academy of Science and Technology (VAST). He has published over 20 research papers. His research interests are in the fields of all-optical networks with emphasis on packet/burst-based switching, Contention Resolution, and Quality of Service; Queueing Theory and Retrial Queue; Wireless Networks. Email: dtchuong@hueuni.edu.vn.

Hoa Ly Cuong procuring MSc in Computer Science in 2017 from the Hue University of Science, Hue University. The areas he has engaged in comprise Queueing Theory and Wireless Networks.
Email: hlcuong90@gmail.com.

Hoang Dinh Long is a lecturer at the Faculty of Physics, University of Education, Hue University, Vietnam. He has been working at his research interests including Control Engineering and Automation, Computer Networks. Email: hdlong.dhsp@hueuni.edu.vn

Duong Duc Hung is a Technical Editor at the HU Journal of Science, Hue University, Vietnam. His main research topics are Computer Networks and Communications; Text Mining.
Email: ddhung@hueuni.edu.vn