On the study of simultaneous service by random number of servers with retrial and preemptive priority

Resham Vinayak and S. Dharmaraja*
Department of Mathematics,
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi – 110016, India
E-mail: resham.iitd@gmail.com
E-mail: dharmar@maths.iitd.ac.in
*Corresponding author

Viswanathan Arunachalam
Department of Mathematics,
Universidad de los Andes,
Cra 1 No 18A-10, Bogota, Colombia
and
Department of Statistics,
Universidad Nacional de Colombia,
Carrera 45 No 26-85, Bogota, Colombia
E-mail: aviswana@uniandes.edu.co
E-mail: varunachalam@unal.edu.co

Abstract: In this paper, we propose to study the performance of a multi-server queuing system in which a customer requires simultaneous service from a random number of servers with the queuing disciplines retrial and preemptive priority. The infinitesimal generator matrix is presented for the proposed model and steady state measures are discussed. In particular, analytical expressions are obtained for the case where a customer (single type) requires simultaneous service from a random number of servers. Also, the generalised stochastic Petri net (GSPN) is developed for the proposed model and the particular cases. The performance of the models is analysed and compared, in terms of average system size and throughput. It is found that queuing systems in which a customer requires simultaneous service from a random number of servers performs better when the queuing discipline of retrial is appended into the system.

Keywords: multi-server system; retrial; preemptive priority; random number of servers.

Reference to this paper should be made as follows: Vinayak, R., Dharmaraja, S. and Arunachalam, V. (2014) ‘On the study of simultaneous service by random number of servers with retrial and preemptive priority’, Int. J. Operational Research, Vol. 20, No. 1, pp.68–90.

Biographical notes: Resham Vinayak is currently doing her PhD in Mathematics from Indian Institute of Technology Delhi, India. She is a post graduate in Mathematics from University of Delhi, India. Her current research interests include queuing theory, Markov modelling, and performance issues of wireless networks and dependability analysis of communication systems.

Copyright © 2014 Inderscience Enterprises Ltd.
1 Introduction

Multi-server queuing systems find wide range of applications in telecommunication and computer systems (Kleinrock, 1976). Customers arrive from a population or source into the system to receive service from one or more server. In this research work, we explore a multi-server queuing system, whereby a customer requires simultaneous service from a random number of servers along with queuing disciplines of retrial and preemptive priority. The assumption that servers serve simultaneously implies that the servers begin and end service simultaneously. Hence, in such systems, the service for a customer cannot begin until the required number of servers are available, since the servers associated with a particular customer start and end service at the same time. Such a scenario is commonly found in wireless networks where resources are to be shared (such as bandwidth). The queuing discipline of priority is assumed to cater to the needs of some special class of customers. But in view of this, the other customers cannot be lost or ignored. Hence, the queuing discipline of retrial is also imposed upon the system. Note that here the word customer is used in its generic sense, and thus maybe a packet in a communication network, a job or a programme in a computer system, a request or an inquiry in a database system, etc.

The rest of this paper is organised as follows: background of the research work is discussed in Section 2 and a detailed description of the proposed model is presented in Section 3. The measures of interest are discussed in Section 4. In Section 5, special cases of the proposed model are discussed. In particular, analytical results are obtained for a multi-server system, in which customers require simultaneous service from a random number of servers. The GSPN for the proposed model is developed in Section 6. The GSPN for the special cases are also derived from the GSPN of the proposed model. The performance of the various models is compared. The results are graphically illustrated in Section 7. Finally, the concluding remarks and future work is presented in Section 8.
2 Background

Queuing systems have been widely explored along with the disciplines of retrial and priority in Arivudainambi et al. (2009), Artalejo (2009, 2010), Artalejo and Corral (2008), Atencia and Moreno (2006), Ayyappan et al. (2010), Artalejo et al. (2001), Azadeh et al. (2010), Jain and Bhargava (2008), Liang and Kulkarni (1993), Roszik et al. (2007) and Wu et al. (2005). However, little has been done to explore multi-server systems whereby a customer requires simultaneous service from a random number of servers. Such systems are complex but their analysis is of much importance since it enables understanding the performance in terms of service being provided and efficient utilisation of resources.

In the past, Green (1980) has explored a multi-server queuing system in which customers require a random number of servers for service. The model is discussed with the assumption that the servers begin service simultaneously, but may free separately. The steady state distribution of the waiting time, the distribution of busy servers and few other measures of importance are obtained. Brill and Green (1984), and Fletcher et al. (1986) presented a multi-server queuing system where customers require simultaneous service from a random number of servers, under the assumption that the servers begin and end the service concurrently, but ignores the queuing discipline of priority and retrial. Federgruen and Green (1984) investigated a queueing model in which an arriving customer requires random number of servers, for a closed queuing system. But the servers are released one at a time.

We observe that much work has been done for a system in which customers require simultaneous service from a random number of servers, however, the system has not been analysed along with the queuing disciplines of retrial and preemptive priority. Hence, we propose to study the system with the queuing discipline of retrial and preemptive priority. The performance of the proposed model is explored and compared with the queuing systems assuming each queuing discipline, i.e., first come first serve, retrial and preemptive priority, separately. Also, since the system is complex, hence, we propose to study the system by the use of generalised stochastic Petri net (GSPN). The Petri net technique was successfully used in Ciardo et al. (1991) and Gharbi et al. (2009) to analyse a retrial queuing system.

3 The queuing model

In this section, a detailed description of the proposed model is provided. The model under investigation is a multi-server system with \( C \) identical servers. Customers arrive from two sources, one being high priority (type 1) and other low priority (type 2). The type 1 customer has preemptive priority over the type 2 customer. We consider the preemption discipline with the push out principle, i.e., the type 2 customer is pushed out or dropped from the system, on preemption. It is further assumed that a customer requires random number of servers, at most up to \( C \) and that it enjoys the monopoly of being served alone by the servers allotted to it. In order to distinguish between the customers on the basis of the number of servers required by them, we classify the customer of type \( i, (i = 1, 2) \) into class \( j, (j = 1, 2, \ldots, C) \), where \( j \) is the number of servers required by a type \( i \) customer. After service completion, all servers are released simultaneously. There is a possibility
that an arriving customer does not find desired number of servers. In case of a type 1 customer does not get the desired number of servers, even by preempting the type 2 customer, then it is dropped. But if an arriving type 2 customer does not find desired number of idle servers, then it joins the virtual orbit (which is of infinite capacity) for retrial. On retrial, each customer is treated the same as a primary customer (a new arriving customer) of type 2. The proposed queueing system is explained diagrammatically in Figure 1.

**Figure 1**  Multi-server queueing system with retrial, preemptive priority and customer requiring random number of servers

![Diagram](https://via.placeholder.com/150)

Mathematically, the above system is described by a stochastic process as follows:

\[ M(t) = \left\{ (X_{i1}(t), ..., X_{iC}(t), X_{21}(t), ..., X_{2C}(t), R(t)) : t \geq 0 \right\} \]

where

\[ X_{ij}(t) \] number of customers of type \( i \) and class \( j \) in the main system at time \( t \); \( i = 1, 2, \]
\[ j = 1, 2, ..., C \]

\[ R(t) \] number of customers of type 2 and class \( j \) in the virtual orbit at time \( t \).

Accordingly, the state space of the process is given by:

\[ \Omega = \left\{ (n_{i1}, ..., n_{iC}, n_{21}, ..., n_{2C}, r) : 0 \leq n_{ij} \leq C, \sum_{i=1}^{2} \sum_{j=1}^{C} (j \times n_{ij}) \leq C, r \in \mathbb{Z}^{+} \cup \{0\} \right\} \]

where \( n_{ij} \) is the number of customers of type \( i \) and class \( j \) and \( r \) is the number of type 2 customers in the virtual orbit. The number of busy servers is the sum of number of servers occupied by type 1 and type 2 customers. Hence:
where $s_i$ is the number of servers busy with customer of type $i$, $i = 1, 2$. In the proposed model, we assume that the arrival of a customer of type $i$ and class $j$ follows Poisson process with rate $\lambda_{ij}$. The service time for both the customers is assumed to be identical and follow exponential distribution with parameter $\mu$. The waiting time of the type 2 customer in the virtual orbit is also assumed to follow exponential distribution with parameter $\alpha$.

With these assumptions, the underlying stochastic process $\{M(t): t \geq 0\}$ satisfies the Markov property at all time instants and is therefore a continuous time Markov chain (CTMC). The entire system will be in equilibrium if the corresponding CTMC is stable. The CTMC will be stable if the main system and the virtual orbit are both stable. Definitely, the main system is stable since it is a finite capacity system (Gross and Harris, 1998; Kleinrock, 1976). The virtual orbit, which is of infinite capacity, mimics the M/M/M/$\infty$ queueing system which is stable if (Liang and Kulkarni, 1993):

$$\rho = \frac{\lambda_R}{\alpha} < 1$$

where $\lambda_R$ is the arrival rate of the customers into the virtual orbit and $\alpha$ is the service rate. Consequently, with this assumption, the entire system will be stable. In the proposed model, we assume a stable system, i.e., $\frac{\lambda_R}{\alpha} < 1$.

Below, the infinitesimal generator matrix for the proposed model is provided. For notational convenience, let $v = (n_{11}, \ldots, n_{1C}, n_{21}, \ldots, n_{2C}, r)$ and $v' = (n'_{11}, \ldots, n'_{1C}, n'_{21}, \ldots, n'_{2C}, r') \in \Omega$. Then the infinitesimal generator matrix, $Q = [q_{v,v'}]$ where $v, v' \in \Omega$, for the corresponding CTMC is given as:

- For $v = (0, 0, \ldots, 0, 0, \ldots, 0)$, the non-diagonal entries are:

$$q_{v,v'} = \begin{cases} 
\lambda_{1,k} & n'_{lk} = 1 \text{ for some } k, \ 1 \leq k \leq C, \\
\lambda_{2,k} & n'_{lk} = 0 \text{ for some } k, \ 1 \leq k \leq C, \\
\lambda_{l,1} & n'_l = 1 \ \forall \ 1 \leq l \neq k \leq C, \\
\lambda_{l,2} & n'_l = 0 \ \forall \ 1 \leq l \neq k \leq C, \\
r' = r = 0 & \\
0 & \text{otherwise}
\end{cases}$$

- For the remaining states, the entries of the matrix, other than the diagonal entries are given below:
On the study of simultaneous service by random number of servers

The diagonal entries are such that the sum of each row of $Q$ is 0. Since the system under consideration is a stable system and the state space is a closed communicating class, therefore the steady state probability exists, independent of the initial state of the system. We denote the steady state probability of state $(n_{11}, \ldots, n_{1C}, n_{21}, \ldots, n_{2C}, r)$ by $P(n_{11}, \ldots, n_{1C}, n_{21}, \ldots, n_{2C}, r)$. In the next section, we mention the measures of interest.

4 Performance measures

For the model discussed in Section 3, the main system is a finite capacity queuing system. Therefore, customers of type 1 enter the virtual orbit and type 2 customer is lost.
if desired number of servers are not available for service. This propels us to evaluate the retrial probability of the type 1 customer and the loss probability of the type 2 customer. At the same time, the average system size and throughput of the system are important measures for understand the overall performance of the system. We now discuss these measures, in detail.

1 Probability that a type 1 customer is lost: A type 1 customer is lost if the customer does not get desired number of servers, even by preemtping the low priority customer. We denote the probability of losing a type 1 and class $j$, ($j = 1, \ldots, C$) customer by $\theta_{1,j}$.

$$\theta_{1,j} = P(\text{customer of type 1 and class } j \text{ is blocked}) = P(\text{at least } C - j + 1 \text{ servers are busy serving type 1 customers}) = \sum_{l=1}^{\infty} \sum_{n_{2l} = 0}^{C} \sum_{r=0}^{\infty} P(n_{11}, \ldots, n_{i_C}, n_{21}, \ldots, n_{2C}, r)$$

Condition 1 $j$

where Condition 1 $j$ refers to that the number of servers occupied by the type 1 and class $j$ customers is at least $C - j + 1$, i.e.,

$$\sum_{k=1}^{C} \sum_{i_{kl} = 0}^{\infty} kn_{ik} \geq C - j + 1.$$

2 Probability that a customer of type 2 undergoes retrial: Such a situation occurs when an arriving customer of type 2 and class $j$, ($j = 1, \ldots, C$) finds at least $C - j + 1$ servers busy serving the customers (both, types 1 and 2). Let $\theta_{2,j}$ denote the probability that the customer of type 2 and class $j$ does not find desired number of servers. Then:

$$\theta_{2,j} = P(\text{customer of type 2 and class } j \text{ does not find desired number of servers}) = P(\text{at least } c - j + 1 \text{ servers are busy serving type 1 and type 2 customers}) = \sum_{l=1}^{\infty} \sum_{n_{2l} = 0}^{C} \sum_{r=0}^{\infty} P(n_{11}, \ldots, n_{i_C}, n_{21}, \ldots, n_{2C}, r)$$

Condition 2 $j$

where Condition 2 $j$ refers to that the total number of servers occupied in serving the customers, is at least $C - j + 1$, i.e.,

$$\sum_{j=1}^{\infty} \sum_{k=1}^{C} \sum_{n_{lk} = 0}^{\infty} kn_{lk} \geq C - j + 1.$$

3 Average response time of a customer in the main system: The waiting time of any customer (i.e., type 1 or type 2 customer) in the main system is equal to its average service time. Denoting the average response time of a customer by $E(W_S)$, we get:

$$E(W_S) = \frac{1}{\mu}.$$
Average number of customers in the main system: We use the Little’s formula to evaluate the average number of type 1 and type 2 customers in the main system. For this, we first evaluate the effective arrival rate of customer of type $i$, $(i = 1, 2)$ and class $j$, $(j = 1, ..., C)$. Denoting the same by $\lambda_{i,j,\text{eff}}$, we get:

$$\lambda_{i,j,\text{eff}} = \lambda_{i,j} (1 - \theta_{i,j}), \ i = 1, 2, \ j = 1, ..., C$$

(5)

where $\theta_{i,j}$ is the blocking probability of customer of type $i$, $(i = 1, 2)$ and class $j$, $(j = 1, ..., C)$. Hence, the average number of customers of type 1 in the main system, denoted by $E(N_1)$, is:

$$E(N_1) = \frac{1}{\mu} \sum_{j=1}^{C} \lambda_{1,j} (1 - \theta_{1,j})$$

(6)

And the average number of customers of type 2 in the main system, denoted by $E(N_2)$ is:

$$E(N_2) = \frac{1}{\mu} \sum_{j=1}^{C} \lambda_{2,j} (1 - \theta_{2,j})$$

(7)

Average number of customers in the virtual orbit: We observe that the virtual orbit mimics the behaviour of an $M/M/\infty$ queue. Denoting the average number of customers in the virtual orbit by $E(N_V)$, we get:

$$E(N_V) = \frac{\lambda_{R,\text{eff}}}{\alpha}$$

where $\lambda_{R,\text{eff}}$ is the effective arrival rate of the type 2 customers into the virtual orbit. A customer of type 2 and class $j$ is directed into the virtual orbit if it does not find desired number of servers. Hence, the effective arrival rate into the virtual orbit is:

$$\lambda_{R,\text{eff}} = \sum_{j=1}^{C} \lambda_{2,j} \theta_{2,j}.$$ 

Throughput: The average throughput of a system is the rate at which customers are serviced successfully. We shall obtain the throughput of the system using a GSPN. In a GSPN the throughput of a timed transition $T$ is interpreted as the average rate at which tokens are deposited in its output places. The software package SHARPE (Sahner et al., 1996) enables to obtain the average throughput of a transition. It is calculated as follows: if $Y(t)$ is the average rate at which tokens are deposited by the transition $T$ in all its output places, up to time $t$, then the throughput $\eta_T$ of the transition $T$ is:

$$\eta_T = \lim_{t \to +\infty} \frac{Y(t)}{t}.$$ 

In the next section, we discuss each of these measures for the special cases of the proposed model.
5 Special cases

In this section, special cases of the model proposed in Section 3 are discussed, by relaxing few assumptions.

5.1 Multi-server queuing system with retrial and preemptive priority

In this case, we relax the condition of a customer requiring random number of servers, i.e., each customer requires single server. As shown in Figure 2, the customer of type 2 continues to avail the liberty of retrial, if it does not find an idle server. Further, the effect of preemptive priority is seen only when there are $C$ type 2 customers and a type 1 customer arrives, in which case, a type 2 customer is dropped and the type 1 customer is accommodated. Since, the requirement of random number of servers is relaxed; hence, the stochastic process is equivalently expressed as:

$$M(t) = \{(X_1(t), X_2(t), R(t)) : t \geq 0\}$$

where

$X_i(t)$ number of customers, of type $i$, in the main system at time $t$, $i = 1, 2$

$R(t)$ number of customers in the virtual orbit at time $t$.

Figure 2 Multi-server queuing system with retrial and preemptive priority

Accordingly, the state space becomes:

$$\{(n_1, n_2, r) : 0 \leq n_i \leq C, r \leq 0\}$$

$r$ being the number of customers in the virtual orbit and $n_i$, $i = 1, 2$ being the number of customer of type $i$ in the main system. With the assumption of random number of servers required for simultaneous service, being relaxed, we get that $\theta_{ij} = \theta_{ik} = \theta_i$, for $i = 1, 2$ and $1 \leq j, k \leq C$. For convenience, we denote the steady state probability of the state $(n_1, n_2, r)$
by $P(n_1, n_2, r)$. The steady state measures obtained for this queueing system have the same expression as obtained in Section 4. A change is observed only in the quantities $\theta_1$ and $\theta_2$. From equations (2) and (3), we obtain $\theta_1$ and $\theta_2$ below:

$$\theta_1 = P(\text{Type 1 customer does not get a server}).$$

This happens when, all servers are busy serving type 1 customers. Hence:

$$\theta_1 = \sum_{r=0}^{\infty} P(0, C, r) = p_C$$

and

$$\theta_2 = P(\text{Type 2 customer does not find an idle server}).$$

A type 2 customer will not find an idle server, if all servers are busy serving customers of types 1 and 2. Therefore:

$$\theta_2 = \sum_{n_1=0}^{C} \sum_{r=0}^{\infty} P(n_1, C-n_1, r).$$

From (5) we obtain the effective arrival rate to the main system as:

$$\lambda_i (1 - p_C) + \sum_{n_1=0}^{C} \sum_{r=0}^{\infty} P(n_1, C-n_1, r).$$

where $\lambda_i, i = 1, 2$, is the arrival rate of customer of type $i$. It can be observed that if the system is reduced to only one type of customer (without retrial), that is only the type 1 customer is allowed to the system, then the system ideally becomes an $M/M/C/C$ system with $\lambda_i (1 - p_C)$ as the effective arrival rate.

5.2 Multi-server queuing system in which a customer requires simultaneous service from random number of servers

In this case, we relax the assumption of retrial and preemptive priority and consider a single type of customer requiring simultaneous service from a random number of servers. As in Section 3, the customers are divided into class $j$, $j = 1, 2, \ldots, C$, where a customer of class $j$ requires $j$ servers. And if the arriving customer does not find desired number of servers, then the customer is dropped. This behaviour is illustrated in Figure 3.

The stochastic process is now equivalently given by:

$$M(t) = \{(X(t), S(t)) : t \geq 0\}$$

where

$X(t)$ number of customers in the system at time $t$

$S(t)$ number of busy servers at time $t$. 


Figure 3  Multi-server queuing system in which a customer requires simultaneous service from a random number of servers

Customer of class $j$, $0 < j < C+1$

is lost if more than $C-j$ servers are busy.

Customers $\xrightarrow{\lambda_1}$ Main system $\xrightarrow{\mu}$ Service $\xrightarrow{}$ $C$ Servers

Figure 4  State transition diagram for the multi-server queuing system in which a customer requires simultaneous service from random number of servers

Then the state space of the system is:

$$\{(i, j) : 0 \leq i \leq j \leq C\}.$$ 

The arrival of the customers is assumed to be a Poisson process with rate $\lambda$. The probability that a customer of class $j$ arrives is assumed to be $q_j$, $1 \leq j \leq C$, $\sum_{j=1}^{C} q_j = 1$.

Then the arrival rate of customers of class $j$ is defined as $\lambda_j = q_j \lambda$, $1 \leq j \leq C$. The state transition diagram for the mentioned stochastic process $M(t)$ is given in Figure 4. As per
our knowledge, this special case has not been dealt with, analytically. We here present an
analytical solution for the same. The steady state probability of the states \((i, j)\) is denoted
by \(v_{ij}\). By solving the balance equations for this system, we get the following relation:

\[
v_{0,0} = C^i \left( \frac{\mu}{\lambda_2} \right)^C v_{C,C}.
\]

\[
v_{i,j} = \frac{C^i}{i!} \left( \frac{\lambda_j}{\lambda_2} \right)^{C-i} \frac{\mu^C}{\lambda_2^C} v_{C,C}, \quad 1 \leq i \leq j \leq C
\]

Since the sum of probabilities is 1, hence:

\[
v_{C,C} = \left[ C^i \left( \frac{\mu}{\lambda_2} \right)^C + \sum_{i=1}^{C} \sum_{j=1}^{C} \frac{C^i}{i!} \left( \frac{\lambda_j}{\lambda_2} \right)^{C-i} \frac{\mu^C}{\lambda_2^C} \right]^{-1}
\]

### 5.3 Multi-server queuing system with retrial

In this case, we relax the condition of a customer requiring simultaneous service from a
random number of servers and allow only one type of customer, who is directed into the
virtual orbit on not finding a free server. The system ideally becomes an \(M/M/C/C\) system
with retrial. As per the above notation, only the type 2 arrives into the system, with
arrival following Poisson process with rate \(\lambda_2\). The time that a customer spends in the
virtual orbit is assumed to be \(\text{Exp}(\alpha)\). The stochastic process is now described by:

\[M(t) = \{(X(t), R(t)) : t \geq 0\}\]

where

\[X(t)\] number of customers in the main system at time \(t\)

\[R(t)\] number of customers in the virtual orbit at time \(t\).

Hence, the state space becomes:

\[\{(i, j) : i \geq 0, 0 \leq j \leq C\}.
\]

Let \(v = (n, r)\) and \(v' = (n', r')\). Other than the diagonal entries, the infinitesimal generator
matrix mentioned above, reduces to:

\[
g_{v,v'} = \begin{pmatrix}
\lambda_2 & n' = n+1, r' = r \\
\lambda_2 & r' = r+1, n \geq c \\
n'\mu, & n' = n-1, r' = r \\
\lambda_2 & r' = r-1, n' = n+1
\end{pmatrix}.
\]

In particular, we illustrate such a system for \(C = 2\), in Figure 5. Also, the infinitesimal
generator matrix provided above, matches with the matrix provided by Artalejo et al.

From equation (3), \(\theta_2\) is given by:
\[ \theta_2 = P(\text{Customer does not find an idle server}) = P(\text{all servers are busy}) = \sum_{r=0}^{\infty} P(C, r) \]

and from equation (5):

\[ \lambda_{\text{eff}} = \lambda_2 (1 - \theta_2). \]

The expressions for expected waiting time in the system and the expected number of customers in the system remain the same as obtained in equations (4), (6) and (7).

**Figure 5** State transition diagram for the multi-server queuing system with retrial and \( C = 2 \)

5.4 **Multi-server queuing system with preemptive priority**

In this case, we relax the condition of retrial and the customer requiring simultaneous service from random number of servers. Hence, the queueing system ideally becomes an \( M/M/C/C - /2 \) with preemptive priority queueing discipline. Accordingly, the stochastic process is equivalently expressed as:

\[ X_{1i}(t) = X_{11}(t) = X_1(t) \text{ and } X_{2i}(t) = X_{21}(t) = X_2(t), i = 1, 2, ..., C \]

i.e.,

\[ M(t) = \{(X_{1i}(t), X_{2i}(t)) : t \geq 0\} \]

and the notation \( (n_{11}, \ldots, n_{1C}, n_{21}, \ldots, n_{2C}, r) \) is equivalently written as \( (n_1, n_2) \), where \( n_i \) is the number of customers of type \( i, i = 1, 2 \) in the system. Further, from equations (2) and (3) we obtain:

\[ \theta_i = P(\text{type 1 customer does not get a server}) = P(\text{all servers are busy serving customers of type 1}) = P(0, C) \]
\[ \theta_2 = P(\text{type 2 customer does not find an idle server}) = P(\text{all servers are busy}) = \sum_{n=0}^{C} P(n, C-n). \]

It is observed that for particular values of \( C \), the results so obtained are consistent with the results obtained through direct analytical computation of an \( M/M/C/C - /2 \) system with preemptive priority (Kleinrock, 1976). Abramov (2007) provides a rigorous analysis for multi-server system with retrials and losses. However, the authors assume the capacity of the virtual orbit to be one only. Also, Shin and Choo (2009) provide an algorithm for a \( M/M/C \) queue with balking, reneging and retrials. The algorithm is based on the generalised truncation method and is derived for numerical computation.

### 5.5 The loss system

In this case, we relax the condition of retrial, preemptive priority and a customer requiring simultaneous service from a random number of servers. Hence, the system mentioned in Section 2, typically becomes the \( M/M/C/C \) loss system, with type 1 customer only. The customer arrival follows Poisson process with rate \( \lambda_1 \). The stochastic process is now expressed by \( M(t) = \{X(t); t \geq 0\} \), where \( X(t) \) represents the number of customers in the system at time \( t \). The state space is \( \{0, 1, \ldots, C\} \) and the steady state probability of the states is denoted by \( p_i, i = 0, 1, \ldots, C \). Deducing from equation (2), we get:

\[ \lambda_{\text{eff}} = \lambda_1 (1 - \theta_1) \]

Hence, from equation (5)

\[ \lambda_{\text{eff}} = \lambda_1 (1 - \theta_1) \]

and accordingly from equation (6):

\[ E(N_t) = \frac{\lambda_{\text{eff}}}{\mu} \]

which matches with the result of an \( M/M/C/C \) loss system.

### 6 The GSPN model

The GSPN framework (Ciardo et al., 1991) provides a powerful set of building blocks to deal with the computational complexities of the state-transition mechanism. It integrates the essential features of concurrency, synchronisation and a sequenced presentation of complex systems. The GSPNs have been successfully used (Jayaparvathy et al., 2007; Ciardo et al., 1991; Gharbi and Ioualalen, 2006; Gharbi et al., 2009) to analyse complex CTMC in a concise way, thereby facilitating a graphical visualisation of a multi-tasking scenario and a dynamic behaviour of a system.
In this section, we develop the GSPN for the proposed model, assuming two servers. The flow of customer in the GSPN is illustrated in Figure 6. The guard functions associated with the GSPN model are listed in Table 1.

**Figure 6**  GSPN for the proposed model

**Table 1**  Guard functions

<table>
<thead>
<tr>
<th>Guard function</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g(i,j)$</td>
<td>if $(#(P_{serve,i,j} \leq 0))$, $i = 1, 2; j = 1, 2$</td>
</tr>
<tr>
<td>$gretry$</td>
<td>if $(#(P_{servers} \leq 0))$</td>
</tr>
<tr>
<td>$gpri_i$</td>
<td>if $(#(P_{11} &gt; 0) \text{ or } #(P_{12} &gt; 0))$, $i = 1, 2$</td>
</tr>
</tbody>
</table>

We now describe the model shown in Figure 6. The system begins with two servers available for service. These two servers are represented by the presence of two tokens in the place $P_{servers}$. The firing of the timed transition $T_{ij}$ represents the arrival of the customer of type $i$ and class $j$. The arrival of a customer of type $i$ and class $j$ to the system is depicted by the presence of a token in the place $P_{ij}$. The system under consideration assumes no waiting space. Hence, an inhibitor arc is placed from place $P_{ij}$ to transition $T_{ij}$, in addition to the guard function $[g(i,j)]$. Depending on the class of customer arriving to the system, i.e., class 1 or 2, accordingly 1 or 2 servers are assigned to the arriving customer.
and the customer moves to the place $P_{serve,ij}$. The number of servers assigned to a customer are fixed by assigning suitable multiplicity to the arc from place $P_{servers}$ to the immediate transitions $t_{ij}$. After the assigning the server(s) to the customer, a token is deposited in the place $P_{serve,ij}$. The presence of a token in the place $P_{serve,ij}$ denotes that a customer of type $i$ and class $j$ is under service. The firing of the timed transition $T_{service,ij}$ denotes service completion of the customer of type $i$ and class $j$. It should be noted that the rate of timed transition $T_{service,ij}$, $i = 1, 2$, is place dependent. That is, the service rate of the class 1 customer is dependent upon the number of customers under service. Whereas, when a customer of class 2 is under service, i.e., two servers are busy serving a customer, the time taken for service is the same as that would be taken by a single server, since the servers begin and end service simultaneously. After the service completion, the servers are released and deposited back in the place $P_{servers}$.

In addition, the type 2 customer is preempted if a type 1 customer, requiring more servers, arrives to the system. This is denoted by the transitions $t_{priority,1}$ and $t_{priority,2}$. Suitable guard functions are placed to preempt the customer only when a high priority customer arrives.

Further, if the type 2 customer does not find desired number of servers, then it undergoes retrial. This is represented by the firing of the immediate transition $t_{retry}$. A token is deposited in the place $P_{retry}$ representing that the customer undergoes retrial. The customer undergoing retrial tries after sometime. The timed transition $T_{retry}$ denotes the time after which the customer retries.

Now, we state the performance measures in terms of the places and transitions in the GSPN. Denoting the average number of tokens in place $P$ by $\#(P)$ and throughput of timed transition $T$ by $\eta_T$, the performance measures stated in Section 4 are obtained as follows:

1. Probability that a type 2 customer undergoes retrial:
   \[
   \frac{\#(P_{retry})}{\#(P_{retry}) + \#(P_{service,21}) + \#(P_{service,22})}
   \]

2. Average number of customers in the main system:
   \[
   \#(P_{service,11}) + \#(P_{service,12}) + \#(P_{service,21}) + \#(P_{service,22}).
   \]

3. Average number of customers in the virtual orbit:
   \[
   P_{retry}.
   \]

4. Throughput:
   \[
   \eta_{service,11} + \eta_{service,12} + \eta_{service,21} + \eta_{service,22}.
   \]

The GSPN models for the special cases, discussed in Section 5, are attained by removing few places and transitions from the GSPN explained above. For convenience, we group the places and transitions associated with type $i$ and class $j$ customer, in a rectangular box, as shown in Figure 6. Thereafter, the GSPN for the special case is obtained by removing the appropriate box and the guard functions are altered suitably. Below, we mention the places and transitions to be removed to attain the aforementioned special cases:
Case 1 Multi-server queuing system with retrial and preemptive priority: all transitions and places associated with the type 2, class 2 customer.

Case 2 Multi-server queueing system in which a customer requires simultaneous service from a random number of servers: all transitions and places associated with the type 2 customer, for both class 1 and 2.

Case 3 Multi-server queuing system with retrial: all transitions and places associated with the type 1 customer (class 1 and class 2), and the type 2, class 2 customer.

Case 4 Multi-server queuing system with preemptive priority: all transitions and places associated with the type 1, class 2 customer and the type 2, class 2 customer.

7 Numerical insights

In this section, we analyse and compare the performance of the proposed models by varying the arrival rate of the customers. We use the software package SHARPE to analyse the system performance. The arrival rate of the type 1 customer is varied from 5 to 50 customers per time unit; the arrival rate of the type 2 customer is varied from 10 to 55 customers per unit time. Further, it is assumed that the probability in which a class 1 customer arrives to the system is 0.67 and that class 2 is 0.33. The average service time of a customer is 0.25 time units and the retrial rate is 5 customers per time unit.

Figure 7 Average size of the main system (see online version for colours)
We first analyse the average system size for the proposed queueing system, with respect to the arrival rate of the type 1 and type 2 customers. The result is graphically illustrated in Figure 7. We observe that the average system size increases with the increase in the arrival rate of the type 2 customer. This is because the customers enter the virtual orbit. On the other hand, the average system size reduces with the increase in the arrival rate of the type 1 customer. This is because the type 2 customer is lost more often, since the type 1 customer is given higher priority.

Next, we analyse the average orbit size with respect to the arrival rate of the type 1 and type 2 customers. This is shown in Figure 8. As expected, the average orbit size increases with the increase in the arrival rate of the customer.

In Figure 9, the system throughput, with respect the arrival rate of the customer is presented. The system throughput decreases with the increase in the arrival rate of the type 1 customer. This is because, the system size increases with the increase in the arrival rate, but due to increase in the arrival of the type 1 customer, more of type 1 customer are serviced than the type 2 customer. In contrast, a marginal increase is observed with the increase in arrival rate of the type 2 customer. This is due to the fact that the customer undergoes retrial.

Figure 10 depicts the probability of retrial of type 2 customer as the arrival rate of the customer of type 1 and type 2 vary. A sharp increase in the probability is observed as the arrival rate of the type 1 customer increases. On the other hand, the probability of retrial decreases initially for increasing arrival rate of the type 2 customer, and thereafter increases. This interprets that if the arrival rate of the type 2 customer is controlled for a fixed arrival rate of the type 1 customer, then a low probability of retrial is achievable.
Figure 9  System throughput (see online version for colours)

Figure 10  Arrival rate vs. probability of retrial (see online version for colours)
In Figure 11, the average system size of two queuing systems are compared. Figure 11(a) represents the average system size of ‘multi-server queuing system in which a customer requires simultaneous service from a random number of servers’. In Figure 11(b), the average system size of a multi-server queuing system with retrial is presented. As expected, the latter queuing system performs better, by demonstrating a lower system size.
Figure 12 depicts the throughput for the two queuing systems discussed above. The throughput of a system allowing retrial to the customer is much higher than the throughput of a queuing system in which a customer required simultaneous service from a random number of servers.

Finally, we compare the throughput of three queuing systems:

1. multi-server queueing system in which a customer requires simultaneous service from a random number of servers
2. multi-server queueing system in which a customer requires simultaneous service from a random number of servers together with retrial
3. multi-server queueing system with retrial and preemptive priority.

The throughput of the three systems are presented in Figure 13(a), Figure 13(b) and Figure 13(c), respectively. We observe that the queuing system in which a customer required simultaneous service from a random number of servers, together with retrial performs best, in terms of throughput of the system. The queuing system with retrial and preemptive priority performs stands second in the comparison. The performance of the queuing system in which a customer required simultaneous service from a random number of servers, performs poorly.

Figure 13  Throughput vs. arrival rate (see online version for colours)
8 Conclusions

This paper focuses upon the study of multi-server queuing systems whereby an arriving customer requires simultaneous service from a random number of services. The proposed model is examined along with the queuing disciplines of retrial and preemptive priority. We observe that the analytical solution of the proposed system is complex. Hence, the infinitesimal generator matrix is provided. Thereafter, the steady state performance measures are discussed. In particular, analytical results are obtained for a queuing system in which a customer receives simultaneous service from a random number of servers (i.e., a multi-server queuing system with a single type of customer, without retrial or preemptive priority). Special cases of the proposed model are also discussed and are found in conformance with the existing literature. Further, to overcome the complexity of the system under consideration, we analyse the system using the GSPN technique. The GSPN model is developed for the proposed model and few particular cases of the proposed model. The performance of the proposed model and that of the special cases is analysed and compared with the help of the GSPN models. It is observed that a queuing system in which customer requires service from a random number of servers performs better along with the queuing discipline of retrial in compare to the queuing disciplines of preemptive priority or first in first out.

In the proposed model, we assume that the inter-arrival, service and the waiting time in the virtual orbit follows exponential distribution. In future, we plan to consider non-exponentially distributed inter-arrival and/or service times. Further, we propose to study the system with server breakdown and repair.

Acknowledgements

This research work is partially supported by IIT Delhi, India, through Jaswinder and Taravinder Chadha Chair. One of the authors (R.V.) is thankful to IIT Delhi, India for providing her financial support through GATE Senior Research Fellowship.

References


