Miskolc Mathematical Notes
Vol. 20 (2019), No. 2, pp. 925–939

HU e-ISSN 1787-2413
DOI: 10.18514/MMN.2019.2620

ANALYSIS OF PRIORITY QUEUE WITH REPEATED ATTEMPTS USING GENERALIZED STOCHASTIC PETRI NETS

SEDDA HAKMI, OUIZA LEKADIR, AND DJAMIL AÏSSANI

Received 29 April, 2018

Abstract. The paper deals with the modeling and analysis of a single server queue with repeated attempts and two priority customers involving generalized stochastic Petri nets. Indeed, the consideration of the recalls and priority introduces great analytical difficulties. Therefore, by using the GSPN, we show how this high level formalism allows us to cope with the complexity of this system. The paper extends previous works on this topic and evaluate different performance characteristics of the system. The Markov chain is obtained and some numerical results are presented to illustrate the effect of the system parameters on the developed performance measures.

2010 *Mathematics Subject Classification:* 68M20; 90B22; 97K60; 60K20; 60J20

Keywords: retrial queueing systems, non-preemptive priority, generalized stochastic Petri nets, modeling, performance evaluation

1. INTRODUCTION

In recent years there have been significant contributions concerning retrial queues which are characterized by the feature for which arriving customers who find all servers not available (blocked), will join the retrial group (Orbit) to insist their demand after a random period.

Retrial queues have been widely used to model many problems in telephone switching systems, telecommunication networks and computer systems. For detailed overviews of the main results and methods on retrial queues, the reader can consult [2,9]. There are many applications of retrial queueing models for which it's necessary to assign priority to customers. This queuing system and its variants are used to model disk memory systems, star-like local area networks, and recently in wireless communication networks where resources are to be shared between many users in this case, introduce priority can cater to the necessity of some classes of customers (see [1,10]).

Several authors have studied retrial queues with priorities, high priority customers are queued and served according to a specific discipline. In the case of blocking, low priority customers leave the system and retry until they find the server free. In these systems the high priority customers have either preemptive or non-preemptive priority over the low priority customers. The presence of the repeated customers and priority on the queueing systems makes their analysis more difficult. Hence, different approximating algorithms and approaches are proposed. Krishna Kumar et al. [13] considered an $M/G/1$ retrial queueing system with two-phase service and possible preemptive resume service discipline. Drekic [8] extended the classical preemptive resume model by enabling previously rendered service to be time-constrained. The $M_1, M_2/G/1$ retrial queue with two types of customers, infinite priority queue for type-1 calls and infinite retrial group of type-2 calls, is investigated by Choi and Park [6]. They obtained the joint generating function of queue lengths by a supplementary variable method. Later Falin et al. [9] extended Choi and Park's model to the case where two types of calls may have different service time distributions. The case where the priority queue has finite capacity has been investigated by Bocharov et al. [5]. Choi et al. in [7] investigated an $M_1, M_2/G/l/K+1$ retrial queue and obtained the joint generating function of queue lengths. Moutzoukis and Langaris [15] extended Choi and Park's model in which there are multiple types of calls, structured batch arrivals and single vacation. They obtained the joint generating function of queue lengths. For a detailed review of the main results and the literature on this topic the reader can consult [3,7,9,10,15,18].

A review of the literature reveals the remarkable fact that the non-homogeneity caused by the flow of repeated attempts, the consideration of the priorities and the finiteness of the sources of customers are the keys to understanding the analytical difficulties that arise in the study of the system that we consider. It is important to indicate that most works cited in the literature about the retrial systems with priorities are analyzed only by the queueing theory. Even, though using this theory, it is difficult to obtain analytical results and the last resort remains approximation methods, and simulation (see [1,10,17]). However, using generalized stochastic Petri nets (GSPN), it is possible to obtain performance indices either with analytical means or by numerical algorithms.

To the best of the authors' knowledge, and after a bibliographic research, there is no such study of finite sources retrial queueing systems with priority customers using the GSPN formalism. Nevertheless, the existing works in the area of retrial queues with priorities concern systems with infinite sources. In general, the queueing analysis of finite-source systems is more difficult than the infinite ones. However, in many practical systems, the number of users who access the system is finite, and it is often important to take into account the fact that the rate of customers arrivals decreases, as the number of customers in the system increases. This can be done with the help of finite source or quasi-random input models.

The difficulty in the queueing analysis of finite sources queues arises on the obtaining of the markov chain associated to theses systems. In this case, the GSPNs formalism are convenient for generating these corresponding Markov chains and consider features that may be hard to obtain directly by queueing theory. Moreover, the behavior of complex systems can be easily and efficiently represented by using GSPN, rather than using Markov chains directly [11]. Thus, in this paper we analyze the $M_2/M_2/1//(N_1,N_2)$ retrial queueing system with a finite population and non preemptive priority. Gomez in [12] studied the finite population retrial queue with quasi-random input and Non-preemptive Priority and obtain some performances indices. However, these performance characteristics have been provided through supplementary variable method which have made the expressions cumbersome and the obtained results cannot be put into practice.

In the present paper, we prove how we apply a simple method, based on generalized stochastic Petri nets, to obtain easily the different performance indices of this system. GSPN are a high-level graphical formalism, which allows an easier description of complex systems behavior [11].

Let's outline the structure of the paper. After the Introduction in Section 1, Section 2 is devoted to a brief introduction to GSPN formalism. In Section 3, we describe the basic model of retrial queueing system with non preemptive priority. In Section 5, we present the associated GSPN model for $M_2/M_2/1//(N_1,N_2)$ retrial queueing system, and we detail the proposed stochastic analysis approach. Furthermore, we derive the computational formulas for evaluating exact performance indices of our system. Finally, Section 5 provides some numerical examples and concluding remarks.

2. An overview of Petri Nets

Petri Nets (PN), in their various shapes and sizes, have been used for the study of the qualitative properties of systems exhibiting concurrency and synchronization characteristics. They are directed bipartite graph [14]consisting of places (used to represent conditions) and transitions (used to describe events that occur in the system), that are connected by directed arcs. The places of the net can contain tokens moving from place to place by the firing of the transition representing an event in the system [4, 16]. When a transition fires tokens from input places are absorbed and tokens are created on each of the output places.

Many extensions to Petri Nets have been proposed to enlarge the class of problems that can be represented and, in order to model real systems. Among these extensions we have Generalized Stochastic Petri nets (GSPN), introduced by Marsan et al are the modeling formalism that can be conveniently used for the analysis of complex models of Discrete Event Dynami Systems (DEDS) and for their performance and reliability evaluation [4, 16]. They have been widely and successfully used as a good modeling tool for the qualitative and quantitative analysis of asynchronous concurrent systems with synchronization, nondeterminism, conflicts and sequencing. In GSPN some transitions are timed, while others are immediate. The immediate transitions are drawn as thin bars and fire without delay. The timed transitions are drawn as empty bars fire after an exponentially distributed delay. A marking in which at least one immediate transition is enabled, is called vanishing marking while a marking in which only timed transitions are enabled is known as a tangible marking. Molloy [16] shows that the stochastic Petri nets are isomorphic to continues time Markov process with discrete space states. Thus, the aspect which contributed significantly to the development of Generalized Stochastic Petri nets is the fact that their performance analysis is based upon Markov theory. The states of the CTMC are the markings in the tangible reachability graph, and the state transition rates are the exponential firing rates of timed transitions in the GSPN. In this case, the performance measures of this GSPN model can be evaluated by computing the steady-state distribution π, using the following linear system:

$$\begin{cases} \pi.Q = 0; \\ \sum\limits_{i \in E} \pi_i = 1; \end{cases} \tag{2.1}$$

where: E is the set of the tangible markings; π_i denotes the steady-state probability that the process is in state M_i, Q is the infinitesimal generator of the Markov process and its elements are given as a function of the timed transitions firing rates.

3. Description of $M_2/M_2/1//(N_1, N_2)$ retrial queueing system

We consider retrial queueing system with two priority classes in which two classes of customers arrive independently from two independent finite-sources of sizes N_1 and N_2, following a quasi-random input rate. Customers from the first source are called high priority customers and customers from second sources are called low priority customers. The time between the service completion and the next service requirement of a customer in class i is an independent exponentially distributed random variable with rate λ_i, i=$\overline{1,2}$. The service facility has a single server that serves customers of all the priority classes. The service times of class i are exponential, independent and identically distributed random variables with rate μ_i, i=$\overline{1,2}$. If the server is free at the time of arrival of a primary customer, then the customer starts to be served. Any high priority customer which, upon arrival, finds the server busy is queued up in an ordinary queue. Upon blocking, low priority customers immediately join a pool of unsatisfied customers, called the orbit. Any orbiting customer try to connect with the server with classical retrial policy after an exponential time period with rate $\nu > 0$, independently to the other customers in orbit, until it finds the server free.

4. The $M_2/M_2/1//(N_1, N_2)$ retrial queueing system GSPN model

In what follows, we present the GSPN model describing $M_2/M_2/1//(N_1, N_2)$ retrial queueing systems (Fig. 1).

- The place $P._{Sour1}$ (respectively $P._{Sour2}$) contains the high priority (respectively low priority) customers, represented by N_1 (respectively N_2) tokens, which represents the condition that none of the N_1 and N_2 customers has arrived for service;

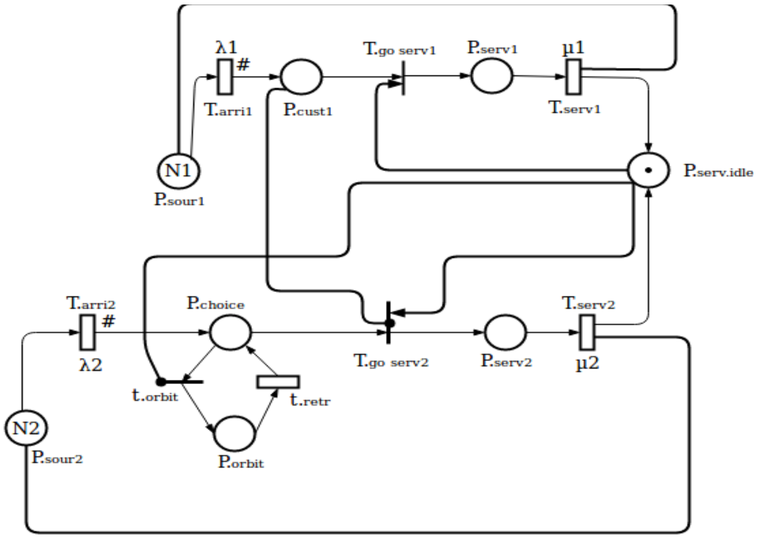


FIGURE 1. GSPN model for the $M_2/M_2/1//(N_1,N_2)$ retrial queueing system.

- The place $P._{Cust1}$ contains the high priority customers;
- The place $P._{Choice}$ represents the condition that a primary or a repeated call is ready for service;
- The place $P._{Orbit}$ represents the orbit;
- The place $P._{serv1}$ (respectively $P._{serv2}$) represents the condition that 'the server is busy by the high priority (respectively low priority) customer';
- The place $P._{serv.Idle}$ represents the condition 'the server is idle', represented by one token.
- Then, the initial marking of the net is :
  $M_o = M(P._{Sour1}), M(P._{Cust1}), M(P._{serv1}), M(P._{Sour2}), M(P._{Choice}),$
  $M(P._{Orbit}), M(P._{serv2}), M(P._{serv.Idle})$
  $M_o = (N_1, 0, 0, N_2, 0, 0, 0, 1).$

When the transition t_{Arri1} fires, one token is taken from P_{Sour1} and is deposited in P_{Cust1}. The firing of t_{Arri1} indicates the arrival of a high priority customer. This firing is marking dependent. Thus, the firing rate of t_{Arri1} depends on the number of tokens in P_{Sour1}. If we have N_1 tokens in P_{Sour1}, the firing rate is $N_1\lambda_1$. The condition of marking dependent firing is represented by the symbol # placed next to the transition t_{Arr1}.

If the arrived customer is a low priority one, the transition t_{Arri2} will fire, then the place P_{Choice} receives a token. Because the transition t_{Arri2} is a marking dependent, so the firing rate is $N_2\lambda_2$.

The immediate transition $tgo.serv1$ is enabled when the place $P_{Serv.Idle}$ contains one token (i.e. the server is idle), and the place P_{Cust1} is not empty (i.e there is at least one priority customer). Once the transition $tgo.serv1$ is fired, a token is removed from each of the two places $P_{Serv.Idle}$ and P_{Cust1}, then it's deposited in P_{Serv1}. This token represents a high priority customer in service.

The immediate transition $t.Orbit$ fires at the arrival of a low priority customer who finds no operational free server i.e. $P_{Serv.idle}$ is empty. Hence, he joins immediately the orbit represented by the place P_{Orbit}. Once in orbit, the customer starts generation of a flow of repeated calls exponentially distributed with rate ν. The firing of transition t_{Retr} represents the arrival of a repeated call from the orbit.

The immediate transition $t_{go.serv2}$ fires if the place P_{Cus1} is empty (This condition is expressed by the inhibitor arc from place P_{Cust1} to the transition $t_{go.serv2}$.), the place $P_{Serv.idle}$ contains one token which represents the idle server and the place P_{Choice} contains one token. So, the place P_{Serv2} receives a token representing a low priority customer in service.

The timed transition t_{Serv2} (respectively, the timed transition t_{Serv1}) is fired to determine the end of the low priority customer period service (respectively, the high priority customer period service). So, one token is deposited in P_{sour2} (respectively in P_{sour1}) which represents the condition that a low priority customer or a priority one will be returned to be idle and a second token is deposited in $P_{Srv.idle}$ which represents the condition that the server is ready to serve another customer.

4.1. *Stochastic analysis and performance measures*

The obtained GSPN model of Figure 2 is bounded and the initial marking is a home states then their steady-state probability distributions exist. In this case, several performance indices can be computed by the formulas given in the following subsections.

► **The mean arrival rate of the high priority requests $\bar{\lambda_1}$ (resp. low priority requests $\bar{\lambda_2}$) are:**

$$\bar{\lambda_1} = \sum_{j\in(SM_j)_1} \lambda_1(M_j)\pi_j, \quad \bar{\lambda_2} = \sum_{j\in(SM_j)_2} \lambda_2(M_j)\pi_j;$$

with : $(SM_j)_k$ is the set of markings where the transition t_{Arri_k} is enabled, and $\lambda_k(M_j)$ is the firing rate associated with the transition t_{Arri_k} in the marking M_j, with $k = \overline{1,2}$.

► **The mean retrial rate of low priority requests:**

The throughput of the transition t_{Retr} gives the mean retrial rate of low priority requests:

$$\bar{\eta} = \sum_{j\in(SM_j)o} \nu(M_j).\pi_j;$$

with : $(SM_j)_o$ is the set of markings where the transition t_{Aretr} is enabled, and $\nu(M_j)$ is the firing rate associated with the transition t_{retr} in the marking M_j.

► **The mean number of the high priority requests η_1 (resp. low priority requests η_2) in the queue:**

$$\eta_1 = \sum_j M_j(P_{Cust1})\pi_j;, \quad \eta_2 = \sum_j M_j(P_{Orbit}).\pi_j;$$

where, $M_j(P_{Cust1})$ is the number of tokens in place P_{Cust1} in the marking M_j and $M_j(P_{Orbit})$ is the number of tokens in place P_{Orbit} in the marking M_j. The sum in this formula is made on all the accessible markings.

► **The mean number of high priority requests η_{S1} (resp. low priority requests η_{S2}) in the system:**

$$\eta_{S1} = \sum_j [M_j(P_{Cust1}) + M_j(P_{Serv1})]\pi_j;$$
$$\eta_{S2} = \sum_j [M_j(P_{Orbit}) + M_j(P_{Serv2})]\pi_j.$$

The sum in this formula is made on all the accessible markings.

► **The mean waiting time of high priority W_1 (resp. low priority W_2) the requests:**

$$W_1 = \frac{\eta_1}{\bar{\lambda}_1}; \qquad W_2 = \frac{\eta_2}{\bar{\lambda}_2}.$$

► **The mean response time of high priority τ_1 (resp. low priority τ_2) requests:**

$$R_1 = \frac{\eta_{S1}}{\bar{\lambda}_1}; \qquad R_2 = \frac{\eta_{S2}}{\bar{\lambda}_2};.$$

► **The blocking probability of low priority customers** The blocking probability of low priority customers is:

$$B_p = \sum_i Prob\{M(P_{Orbit}) \geq 1 \ \ and \ \ M(P_{\cdot serv.Idle}) = 0\}.$$

► **The probability that a low priority customers is being served** This corresponds to the probability that the server is busy by low priority customer, it's given by P_s:

$$P_s = \sum_i Prob\{M(P_{serv2}) = 1\}.$$

As a numerical example for the non-preemptive priority queueing system with a single finite population, we consider the $M_2/M_2/1//(2,2)$ retrial queueing system with $N_1 = 2$ and $N_2 = 2$. The reachability graphs of these models are finite and strongly connected. Hence, underlying Markov process is ergodic and the steady-state distribution exists and is unique. We construct the continuous-time Markov chain (CTMC) (see Figure 2) where the states of this CTMC are the tangible markings of the tree. The vanishing markings are merged with their successor tangible markings since it takes zero time to go through a vanishing marking. Hence, the infinitesimal generator Q of underlying Markov chain is derived and the steady-state marking probability distribution

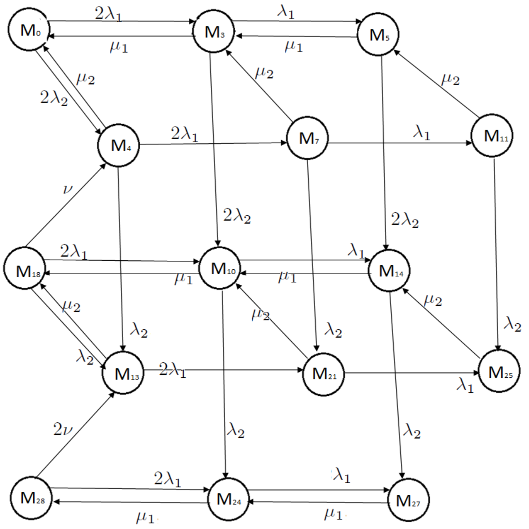


FIGURE 2. The CTMC for the $M_2/M_2/1//(2,2)$ retrial queueing GSPN model.

$$\pi = (\pi_0, \pi_3, \pi_4, \pi_5, \pi_7, \pi_{10}, \pi_{11}, \pi_{13}, \pi_{14}, \pi_{15}, \pi_{18}, \pi_{21}, \pi_{24}, \pi_{25}, \pi_{27}, \pi_{28})$$

is the solution of the system 2.1.

- **The mean arrival rate of the high priority customers**

$$\bar{\lambda_1} = 2\lambda_1(\pi_0 + \pi_4 + \pi_{13} + \pi_{18} + \pi_{28}) + \lambda_1(\pi_3 + \pi_{10} + \pi_{24} + \pi_7 + \pi_{21}).$$

- **The mean arrival rate of the low priority customers**

$$\bar{\lambda_2} = 2\lambda_2(\pi_0 + \pi_3 + \pi_5) + \lambda_2(\pi_4 + \pi_7 + \pi_{11} + \pi_{18} + \pi_{10} + \pi_{14}).$$

- **The mean number of the high priority customers in the queue**

$$\eta_1 = 2(\pi_{11} + \pi_{25}) + \pi_5 + \pi_7 + \pi_{14} + \pi_{21} + \pi_{27}.$$

- **The mean number of the low priority customers in the orbit**

$$\eta_2 = 2(\pi_{24} + \pi_{27} + \pi_{28}) + \pi_{10} + \pi_{13} + \pi_{14} + \pi_{18} + \pi_{21} + \pi_{25}.$$

- **The mean number of the high priority customers in the system**

$$\eta_{S1} = 2(\pi_5 + \pi_{11} + \pi_{14} + \pi_{25} + \pi_{27}) + \pi_3 + \pi_7 + \pi_{10} + \pi_{21} + \pi_{14} + \pi_{24}.$$

- **The mean number of the low priority customers in the system**

$$\eta_{S2} = 2(\pi_{13} + \pi_{21} + \pi_{24} + \pi_{25} + \pi_{27}) + \pi_{28} + \pi_4 + \pi_7 + \pi_{11} + \pi_{10} + \pi_{14} + \pi_{18}.$$

- **The mean waiting time of the customers**

$$W_1 = \frac{2(\pi_{11} + \pi_{25}) + \pi_5 + \pi_7 + \pi_{14} + \pi_{21} + \pi_{27}}{2\lambda_1(\pi_0 + \pi_4 + \pi_{13} + \pi_{18} + \pi_{28}) + \lambda_1(\pi_3 + \pi_{10} + \pi_{24} + \pi_7 + \pi_{21})}.$$

$$W_2 = \frac{2(\pi_{24} + \pi_{27} + \pi_{28}) + \pi_{10} + \pi_{13} + \pi_{14} + \pi_{18} + \pi_{21} + \pi_{25}}{2\lambda_2(\pi_0 + \pi_3 + \pi_5) + \lambda_2(\pi_4 + \pi_7 + \pi_{11} + \pi_{18} + \pi_{10} + \pi_{14})};$$

- **The mean response time of the customers**

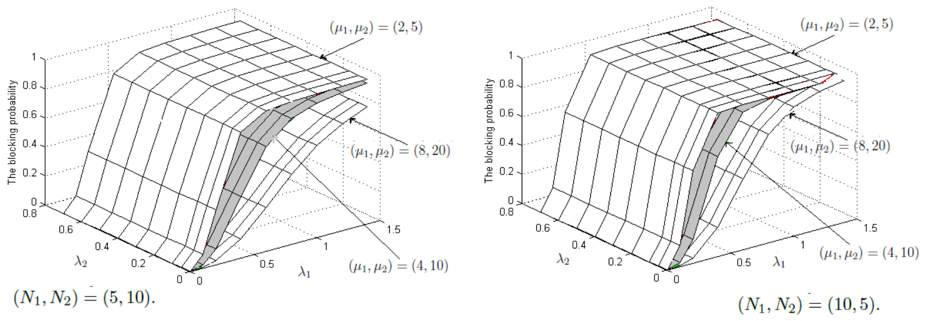
$$R_1 = \frac{2(\pi_5 + \pi_{11} + \pi_{14} + \pi_{25} + \pi_{27}) + \pi_3 + \pi_7 + \pi_{10} + \pi_{21} + \pi_{14} + \pi_{24}}{2\lambda_1(\pi_0 + \pi_4 + \pi_{13} + \pi_{18} + \pi_{28}) + \lambda_1(\pi_3 + \pi_{10} + \pi_{24} + \pi_7 + \pi_{21})}.$$

$$R_2 = \frac{2(\pi_{13} + \pi_{21} + \pi_{24} + \pi_{25} + \pi_{27}) + \pi_{28} + \pi_4 + \pi_7 + \pi_{11} + \pi_{10} + \pi_{14} + \pi_{18}}{2\lambda_2(\pi_0 + \pi_3 + \pi_5) + \lambda_2(\pi_4 + \pi_7 + \pi_{11} + \pi_{18} + \pi_{10} + \pi_{14})};$$

5. Numerical examples

In this section, we present the results of numerical experiments the goal is to demonstrate the feasibility of the proposed approach and to give some insight into quantitative behavior of the system.

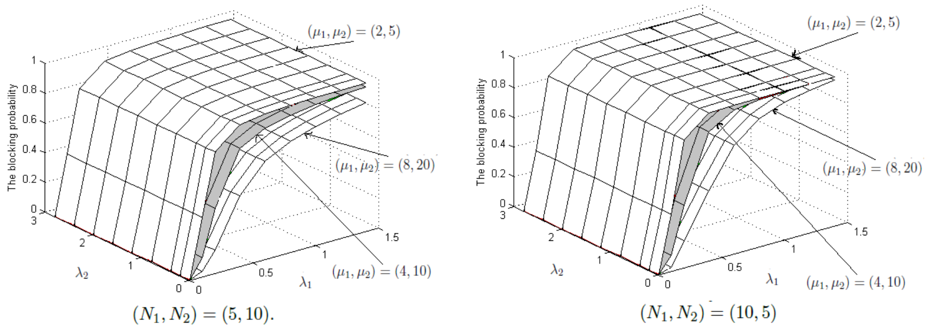
In Figures 3 and 4 we show the influence of the arrival rates λ_1, λ_2 with $\lambda_2 = \frac{\lambda_1}{2}$ on the blocking probability B_p. We have presented three curves which correspond to the different values of service rates, (μ_1, μ_2). In all the numerical experiments, we assume $\nu = 3.5$. From Figures 3 and 4 it is shown that the blocking probability B_p increases as the rates λ_1 increases and approaches one. The increasing of B_p is rapidly for a small value of (μ_1, μ_2) and for $(N_1, N_2) = (10, 5)$. We see also that the blocking probability B_p increases rapidly when $\lambda_2 = 2\lambda_1$. In Figure 4 the blocking probability B_p curves are plotted versus the arrival rate λ_1, λ_2 with $\lambda_2 = 2\lambda_1$ for



$(N_1, N_2) = (5, 10)$.

$(N_1, N_2) = (10, 5)$.

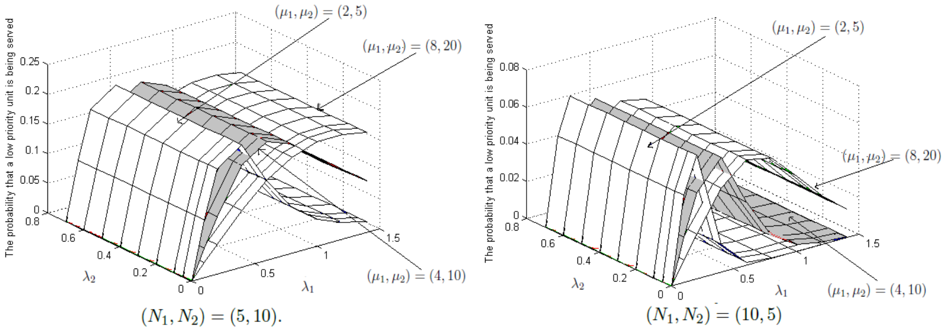
FIGURE 3. B_p versus the arrival rates λ_1 and λ_2 with $\lambda_2 = \frac{\lambda_1}{2}$.



$(N_1, N_2) = (5, 10)$.

$(N_1, N_2) = (10, 5)$

FIGURE 4. B_p versus the arrival rates λ_1 and λ_2 with $\lambda_2 = 2\lambda_1$.



$(N_1, N_2) = (5, 10)$.

$(N_1, N_2) = (10, 5)$

FIGURE 5. P_s versus the arrival rate λ_1 and λ_2 with $\lambda_2 = \frac{\lambda_1}{2}$.

different values of (μ_1, μ_2) and (N_1, N_2). Figure 5 illustrates the behavior of the probability that a low priority customer is being served P_s versus the arrival rate λ_1, λ_2 with $\lambda_2 = \frac{\lambda_1}{2}$. We have presented three curves which correspond to the different values of service rates, (μ_1, μ_2).

Figures 6 shows the effect of the arrival rate λ_1, λ_2 with $\lambda_2 = \frac{\lambda_1}{2}$ on the probability that a low priority customer is being served P_s. We have presented three curves which correspond to the different values of service rates, (μ_1, μ_2). From these figures, it is shown that we see the probability that a low priority customer is being served P_s increases until the maximum and decreases to approach zero. We notice that P_s approaches zero with the increases of λ_1. The zero is reached rapidly for lower values of (μ_1, μ_2) and for a large number of high priority customers $(N_1, N_2) = (10, 5)$.

In the following we show the effect of of the arrival rates λ_1, λ_2 on the mean number of the high priority customers and low priority customers in the queue and in orbit respectively. Figures 7 displays $\bar{Q}_i$ for $i \in \{1, 2\}$ versus the arrival rates λ_1 and λ_2. We see that the mean number of customers in the queue and orbit is

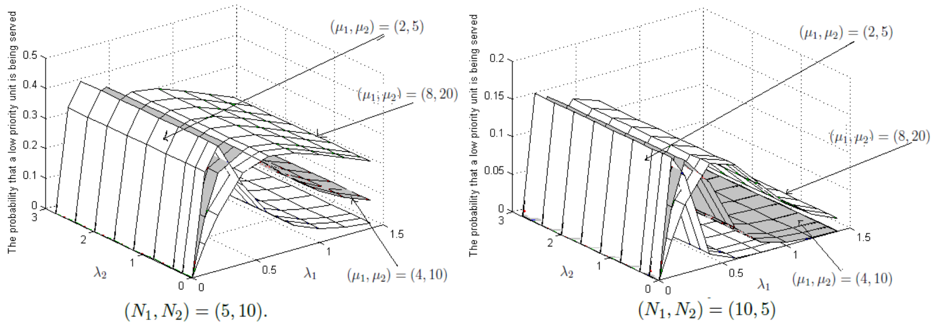


FIGURE 6. P_s versus the arrival rate λ_1 and $\lambda_2 = 2\lambda_1$

an increasing function of arrival rate λ_1. However, when more and more arrival customers arrive, Q_2 approaches N_2 ie the low priority customers are blocked. All

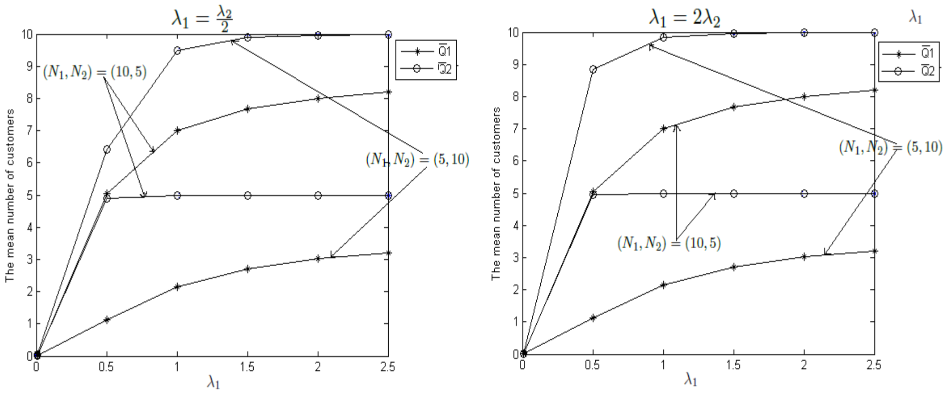


FIGURE 7. $\bar{Q}_1$, $\bar{Q}_2$ versus λ_1.

the above remarks are obtained by considering the parameters $(N_1 = 5, N_2 = 10)$ or $(N_1 = 10, N_2 = 5)$, we checked them with other different values of N_1 and N_2. However, we gave the similar values used by Gomez in order to compare our results obtained via our approach Petri nets and those obtained by him in [12].

In Figure 8 the mean response time is spotted versus the retrial rate ν. We have presented several curves which correspond to the different values of (N_1, N_2) and we have assumed that $(\lambda_1, \lambda_2) = (0.4, 0.8)$, $(\mu_1, \mu_2) = (8, 20)$. From these figures, we see that the mean response time is a decreasing function of retrial rate, for low priority customers and slightly increasing for high priority customers. Moreover, the high priority customers have the best mean response times in all cases. The worst mean response times are given by a high values of N_1 and for a small value of ν.

Furthermore, Figure 8 shows also that the retrial rate has a significant influence on the mean response time for low retrial rate values. This result is logical because the increasing of the retrial rate describes the fact that the repeated requests intensity increases. Hence, customers in orbit have more chance to be served compared to new arrivals,which explains the amelioration of the mean response time. In Tables 1 and 2 we check respectively, the effect of arrival rates λ_1 and λ_2 with $\lambda_2 = \frac{\lambda_1}{2}$ on the mean response time for respectively $(\mu_1,\mu_2) = (8,20)$ and $(\mu_1,\mu_2) = (4,10)$.

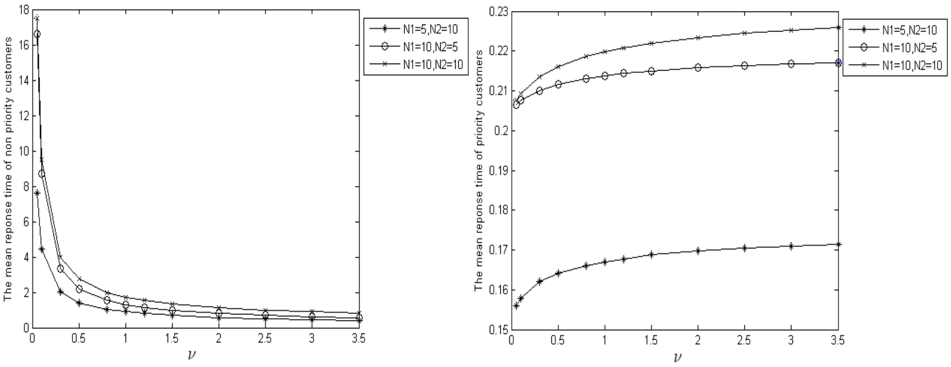


FIGURE 8. Mean response time of customers versus retrial rate.

TABLE 1. The effect of arrival rates on the mean response time for $(\mu_1,\mu_2) = (8,20)$.

	$(N_1,N_2) = (5,10)$		$(N_1,N_2) = (10,5)$	
λ_1	High priority customers	Low priority customers	High priority customers	Low priority customers
0.01	0.1250	0.0533	0.1265	0.0555
0.1	0.1327	0.0861	0.1411	0.1140
0.2	0.1409	0.1288	0.1602	0.2016
0.4	0.1583	0.2384	0.2093	0.4916
0.6	0.1769	0.3889	0.2735	1.0549
0.8	0.1961	0.5898	0.3497	2.1343
1	0.2151	0.8482	0.4313	4.1395
1.2	0.2334	1.1672	0.5114	7.8098

From Tables 1 and 2 we can see that the mean response time is an increasing function of the arrival rates which has a significant influence when they are high. The mean response time of high priority customers is better. Furthermore, the mean response

TABLE 2. The effect of arrival rates on the mean response time for $(\mu_1, \mu_2) = (4, 10)$.

	$(N_1, N_2) = (5, 10)$		$(N_1, N_2) = (10, 5)$	
λ_1	High priority customers	Low priority customers	High priority customers	Low priority customers
0.01	0.2530	0.10856	0.2560	0.1146
0.1	0.2817	0.2013	0.3204	0.3028
0.2	0.3168	0.3450	0.4187	0.7087
0.4	0.3938	0.8246	0.7019	3.0869
0.6	0.4716	1.6716	1.0268	11.3301
0.8	0.5417	2.9668	1.3057	38.3670
1	0.6029	4.6891	1.5171	121.1
1.2	0.6566	7.0179	1.6723	350.57

time of low priority customers increases rapidly with a high values of N_1 and lower values of (μ_1, μ_2).

Finally, we have concluded that for retrial queueing systems with high priority the optimal results are obtained when the arrival rates, service rates are low and the retrial rate is high with a small number of low priority customers.

6. CONCLUSION

The paper presents a technique allowing to obtain exact performance indices, based on Generalized Stochastic Petri nets (GSPN), to analyze $M_2/M_2/1//(N_1, N_2)$ retrial queueing system with finite population. Generalized Stochastic Petri nets (GSPN) have proven to be a powerful and enduring graphically oriented framework for modeling and performance analysing of complex systems. First, We have developed formulas of the main stationary performance indices based on stationary probabilities and network parameters. Furthermore, we have presented numerical examples to illustrate the efficiency of the proposed approach where we study the effect of network parameters on performance indices. Finally, we conclude that the arrival rate of high priority customers, the arrival rate of low priority customers, retrial rate and the number of customers in the sources are the major factors affecting the performance of this system. From these results, we conclude that the Petri net theory can be used to add powerful analysis capabilities to high priority and retrial queueing systems.

REFERENCES

[1] B. Almasi, T. Berczes, A. Kuki, J. Sztrik, and J. Wang, "Performance Modeling of Finite-Source Cognitive Radio Networks." *Acta Cybernetica*, vol. 22, pp. 617–631, 2016, doi: 10.14232/actacyb.22.3.2016.5.

[2] J. Artalejo, "Accessible bibliography on retrial queues: progress in 2000-2009." *Math. Comput. Model.*, vol. 51, no. 9, pp. 1071–1081, 2010, doi: 10.1016/j.mcm.2009.12.011.

[3] Z. Atan, L. V. Snyder, and G. R. Wilson, "Transshipment policies for systems with multiple retailers and two demand classes." *Springer, Berlin/Heidelberg*, vol. 40, no. 1, pp. 159–186, 2018, doi: 10.1007/s00291-017-0497-8.

[4] G. Balbo, "Introduction to generalized stochastic Petri nets," *Springer Verlag, Berlin*, vol. 4486, pp. 83–131, 2007, doi: 10.1007/978-3-540-72522-03.

[5] P. P. Bocharov, O. I. Pavlova, and D. A. Puzikova, "On the $M/G/1/r$ retrial queueing systems with priority of primary customers," *Math. Comput. Modelling*, vol. 30, no. 3-4, pp. 89–98, 1999, doi: 10.1016/S0895-7177(99)00134-X.

[6] B. D. Choi and K. K. Park, "The $M/G/1$ retrial queue with Bernoulli schedule," *Queueing Systems*, vol. 7, no. 2, pp. 219–227, 1990, doi: 10.2307/3214586.

[7] B. D. Choi, K. B. Choi, and Y. W. Lee, "$M/G/1$ retrial queueing systems with two types of calls and finite capacity," *Queueing Systems*, vol. 19, no. 1, pp. 215–229, 1995, doi: 10.1007/BF01148947.

[8] S. Drekic, "A preemptive resume queue with an expiry time for retained service." *Perform. Eval.*, vol. 54, no. 1, pp. 59–74, 2003, doi: 10.1016/S0166-5316(02)00232-8.

[9] G. Falin, J. Artalejo, and J. Martin, "On the single server retrial queue with priority customers," *Queueing System*, vol. 14, no. 3, pp. 439–455, 1993, doi: 10.1007/BF01158878.

[10] S. Gao, "A preemptive priority retrial queue with two classes of customers and general retrial times." *Oper. Res.*, vol. 15, no. 2, pp. 233–251, 2015, doi: 10.1007/s12351-015-0175-z.

[11] N. Gharbi and M. Ioualalen, "Numerical investigation of finite-source multiserver systems with different vacation policies." *Journal of Computational and Applied Mathematics, Elsevier (North-Holland), Amsterdam*, vol. 234, no. 3, pp. 625–635, 2010, doi: 10.1016/j.cam.2009.11.040.

[12] A. Gómez-Corral, "Analysis of a Single-Server Retrial Queue with Quasi-Random Input and Non-preemptive Priority." *Computers and Mathematics with Applications*, vol. 43, no. 6, pp. 767–782, 2002, doi: 10.1016/S0898-1221(01)00320-0.

[13] B. Kumar, A. Vijayakumar, and D. Arivudainambi, "An $M/G/1$ retrial queueing system with two-phase service and preemptive resume." *Queueing Systems*, vol. 113, no. 1, pp. 61–79, 2002, doi: 10.1023/A:1020901710087.

[14] A. R. F. Laali and H. H. S. Javadi, "Spectra Of Some Special Bipartite Graphs," *Miskolc Mathematical Notes HU e-ISSN 1787-2413,DOI: 10.18514/MMN.2017.1587*, vol. 18, no. 1, pp. 295–305, 2017, doi: 10.18514/MMN.2017.1587.

[15] C. Langaris and E. Moutzoukis, "Non-preemptive priorities and vacations in a multiclass retrial queueing system." *Commu. Statis. Stochastic Models*, vol. 12, no. 3, pp. 455–472, 1996, doi: 10.1080/15326349608807394.

[16] M. K. Molloy, "Performance analysis using Stochastic Petri Nets." *IEEE Trans.comp.*, vol. 31, no. 9, pp. 913–917, 1982, doi: 10.1109/TC.1982.1676110.

[17] C. Peng and H. Zeng, "Response time analysis of digraph real-time tasks scheduled with static priority: generalization, approximation, and improvement." *Real-Time Syst. Springer US, New York, NY*, vol. 54, no. 1, pp. 91–131, 2018, doi: 10.1007/s11241-017-9290-7.

[18] F.-F. Wang, A. Bhagat, and T.-M. Chang, "Analysis of priority multi-server retrial queueing inventory systems with MAP arrivals and exponential services." *Springer India, New Delhi, Delhi, India; Operational Research Society of India, Kolkata, West Bengal, India*, vol. 54, no. 1, pp. 44–66, 2017, doi: 10.1007/s12597-016-0270-9.

Authors' addresses

Sedda Hakmi
Research Unit LaMOS (Modeling and Optimization of Systems, Bejaia University 06000, Algeria
E-mail address: sed.hakmi@gmail.com

Ouiza Lekadir
Research Unit LaMOS (Modeling and Optimization of Systems, Bejaia University 06000, Algeria
E-mail address: ouizalekadir@gmail.com

Djamil Aïssani
Research Unit LaMOS (Modeling and Optimization of Systems, Bejaia University 06000, Algeria
E-mail address: lamos_bejaia@hotmail.com