DOI 10.1007/s10958-020-04759-4
Journal of Mathematical Sciences, Vol. 246, No. 4, April, 2020

THE SIMULATION OF FINITE-SOURCE RETRIAL QUEUEING SYSTEMS WITH COLLISIONS AND BLOCKING

Á. Tóth[1], T. Bérczes[1], J. Sztrik[1], A. Kuki[1], and W. Schreiner[2]

This paper investigates, using a simulation program, a retrial queuing system with a single server which is subject to random breakdowns. The number of sources of calls is finite, and collisions can take place. We assume that the failure of the server blocks the system's operation such that newly arriving customers cannot enter the system, contrary to an earlier paper where the failure does not affect the arrivals. All the random variables included in the model construction are assumed to be independent of each other, and all times are exponentially distributed except for the service time, which is gamma distributed. The novelty of this analysis is the inspection of the blocking effect on the performance measures using different distributions. Various figures represent the impact of the squared coefficient of the variation of the service time on the main performance measures such as the mean and variance of the number of customers in the system, the mean and variance of the response time, the mean and variance of the time a customer spends in the service, and the mean and variance of the sojourn time in the orbit.

1. Introduction

This paper deals with the investigation of systems with retrial queues; these are effective tools for modeling real life situations emerging in major telecommunication systems, such as telephone switching systems, call centers, computer networks, and computer systems. It is a typical feature of this system that customers who find the server busy or unavailable are directed toward the so-called orbit instead of the service area. Models of retrial queuing systems have also become popular in two-way communication in the last years; see for example [6, 12]. In many application fields, such as cellular mobile networks, computer networks, and local-area networks with random access protocols and with multiple access protocols, researchers can utilize the models of retrial queues like in [3, 9].

In many papers the server is expected to function in all the available time, which is quite unrealistic. In real life applications, reliability cannot be assumed. For example, during wireless communication, several factors can impact the transmission rate, so during packet transmission, interruptions are inevitable. The study of the unreliability of retrial queuing systems has a great importance especially considering the system characteristics and performance measures. Recently several papers investigated finite-source retrial queues where the server is subject to breakdown; see for example [2,5,8,11,18,20,21].

In many practical situations during data transmission, where the number of communication channels is limited, users (sources) usually compete with each other for the available resources. When various sources begin to distribute data towards the same server, collisions are produced, resulting in the loss of the transmission and consequently in the need for retransmission. Creating an efficient method for preventing conflict and corresponding message delay is essential in eliminating the possibility of collision. Recent results on retrial queues with collisions can be found, for example, in [1, 4, 10, 13, 14, 17].

The aim of this paper is to examine the operation of a system in the case of server failure and to analyze the effect of blocking. The considered system contains a nonreliable server with a finite number

[1] University of Debrecen, Debrecen, Hungary, e-mail: toth.adam@inf.unideb.hu, sztrik.janos@inf.unideb.hu, berczes.tamas@inf.unideb.hu, kuki.attila@inf.unideb.hu
[2] Research Institute for Symbolic Computation (RISC), Johannes Kepler University Linz, Linz, Austria, e-mail: Wolfgang.Schreiner@risc.jku.at

Proceedings of the XXXV International Seminar on Stability Problems for Stochastic Models, Perm, Russia, September 24–28, 2018. Part I.

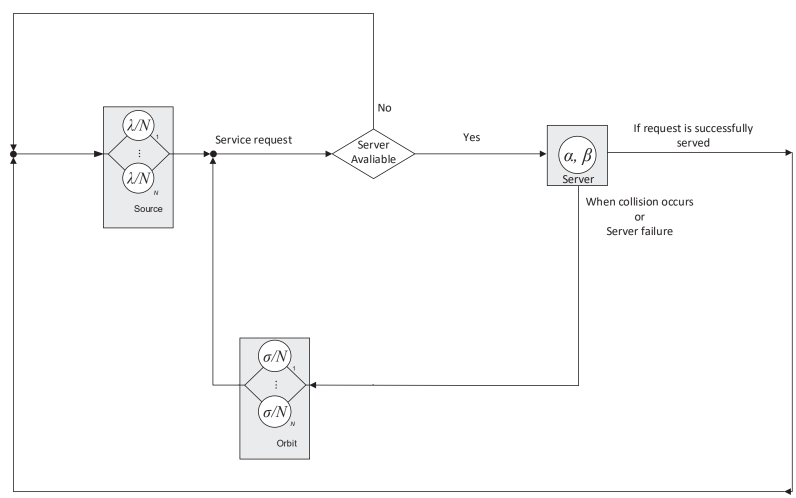

Fig. 1. System model.

of sources where collisions can occur. A simulation model is developed using SimPack [7], a collection of C/C++ libraries and executable programs, in order to obtain the main steady-state performance measures. The SimPack toolkit provides the user with a set of utilities that illustrate the basics of building a working simulation from a model description. One of its main advantages is that it really depends on the user what performance measure to calculate and how the model is built up. The novelty of this work is to provide a possible operation method when the server is not available and to compare the received results with those of [19], where the arrivals of the customers are not blocked by the failure of the server.

2. Model description and notations

This paper investigates a retrial queuing model with one server. The operation of the model can be seen in Fig. 1.

In this model, N customer resides in the source, which is finite such that the system is stable in every moment. Each customer can produce a call (incoming customers in the system) towards the server with rate λ/N, so the inter-request times are exponentially distributed with parameter λ/N. Our model does not contain a queue; therefore the service of an incoming customer starts instantaneously if the server is idle and in operational state (up). The service time follows a gamma distribution with parameters α and β. The customer returns to the source after a successfully executed service. When an arriving customer (from either the source or the orbit) finds the server busy, meaning that another customer is under service, a collision arises such that both requests are sent into the orbit. Customers located in the orbit may retry their requests for service after a random time. The distribution of this period is exponential with parameter σ/N. Because of the unreliability of the server, it breaks down after some exponentially distributed period. The parameter of this distribution is γ_0 when the server is busy and γ_1 when it is idle. The repair mechanism begins immediately upon the failure of the server. The repair time is an exponentially distributed random variable with parameter γ_2. It is assumed that all the random variables involved in the model creation are totally independent of each other. The reason that we deal with rates λ/N and σ/N is that in [15,16] similar systems were treated by an asymptotic method where N tends to infinity, and it was proved that the number of customers in the systems follows a normal distribution. As we can see in Fig. 2, this happens in our case, too.

When a busy server breaks down, the interrupted customer is transmitted to the orbit. Two types of operation mode are distinguished in the case of a server failure:

- without blocking: requests are able to enter the system and these are forwarded to the orbit instantaneously;
- with blocking: the arriving customers cannot enter the system; they return to the source and a new request generation starts.

3. Simulation results

The applied values of the input parameters are presented in Table 1. Tables 2 and 3 indicate the numerical results of the main performance measures comparing the two possible operation methods whenever the server breaks down. Here the following notations are introduced: **E(NS)**: the mean number of customers, **Var(NS)**: the variance of the number of customers, **E(T)**: the mean response time, **Var(T)**: the variance of the response time, **E(W)**: the mean waiting time, **Var(W)**: the variance of the waiting time, **E(S)**: the average successful service time, **Var(S)**: the variance of the successful service time, **E(IS)**: the mean period of the interrupted service time.

Table 1. Numerical values of model parameters

Case studies								
No	N	λ/N	γ_0	γ_1	γ_2	σ/N	α	β
Fig. 2	100	0.01	0.1	0.1	1	0.01	0.5,1,2	0.5,1,2
Fig. 3	100	0.01..6	0.1	0.1	1	0.01	0.5,1,2	0.5,1,2
Fig. 4	100	0.01..6	0.1	0.1	1	0.01	0.5	0.5
Fig. 5	100	0.01..6	0.1	0.1	1	0.01	1	1
Fig. 6	100	0.01..6	0.1	0.1	1	0.01	2	2
Fig. 7	100	0.01	0.05..0.5	0.05..0.5	1	0.01	0.5	0.5
Fig. 8	100	0.01	0.05..0.5	0.05..0.5	1	0.01	1	1
Fig. 9	100	0.01	0.05..0.5	0.05..0.5	1	0.01	2	2
Fig. 10	100	0.01	0.05..0.5	0.05..0.5	1	0.01	0.5	0.5
Fig. 11	100	0.01	0.05..0.5	0.05..0.5	1	0.01	0.5	0.5
Fig. 12	100	0.01	0.05..0.5	0.05..0.5	1	0.01	1	1
Fig. 13	100	0.01	0.05..0.5	0.05..0.5	1	0.01	2	2

Table 2. Numerical results when blocking is not applied

Case	**E(NS)**	**Var(NS)**	**E(T)**	**Var(T)**	**E(W)**	**Var(W)**	**E(S)**	**Var(S)**
1	63.6842	27.9734	175.3073	65657.3454	174.5884	65434.6696	0.3147	0.1979
2	70.5912	24.3012	239.9734	105273.4267	238.9734	104918.6389	0.4784	0.2289
3	75.1825	21.2439	302.8106	151781.1411	301.5377	151277.6006	0.6472	0.2095

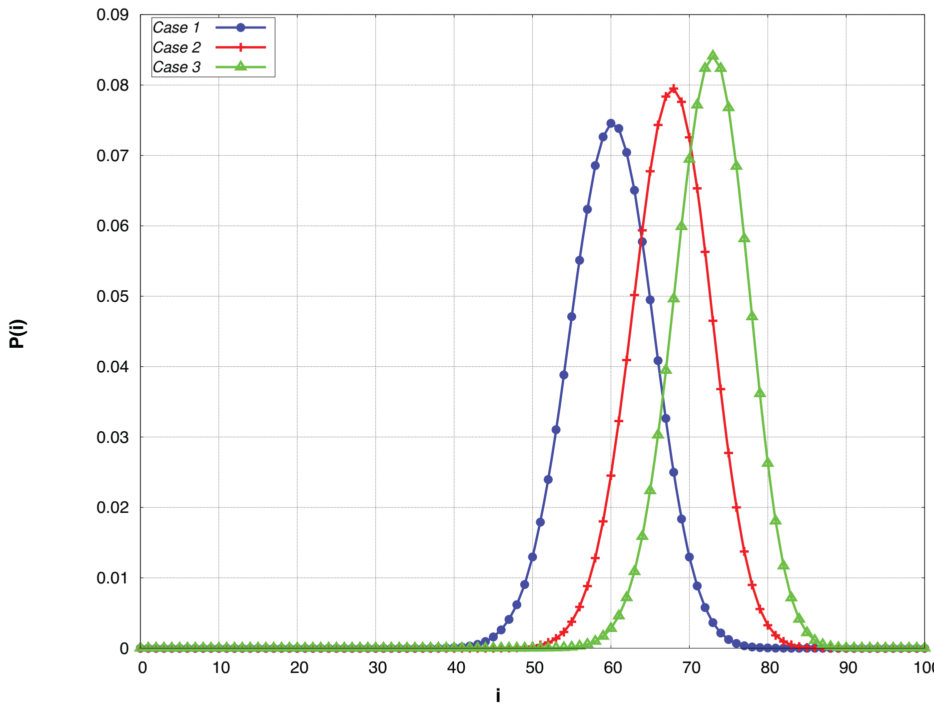

Fig. 2. Comparison of steady-state distributions.

Table 3. Numerical results when blocking is applied

Case	E(NS)	Var(NS)	E(T)	Var(T)	E(W)	Var(W)	E(S)	Var(S)	E(IS)
1	60.0320	28.5106	165.1955	63510.5573	164.4769	63287.935	0.3146	0.1977	0.4040
2	67.64	25.1115	229.8805	103125.2009	228.8805	102770.5034	0.4783	0.2288	0.5217
3	72.69	22.4004	292.6778	149545.4531	291.4051	149042.3599	0.6471	0.2095	0.6256

In Fig. 2 the steady-state distribution of the investigated cases is presented when blocking is applied. The parameters of the service time of Case 2 are unique because for $\alpha = 1$ we have the exponential distribution. When the values of parameters α and β increase, this results in a higher mean number of customers, although the mean of the gamma distribution is the same in all cases ($\alpha/\beta = 1$). Looking carefully at Fig. 2, it can be observed that all cases correspond to the normal distribution. Figure 3 displays the mean waiting time of the customers as a function of the incoming generation rate when blocking is applied. Consequently, because the mean number of customers is higher in Cases 2 and 3 compared to Case 1, the mean waiting time is greater, too. It is a specialty that, when using retrial queues with finite-source, the mean waiting time has a maximum value. This is a general characteristic of retrial queues but depends on the parameter settings used.

In Figs. 4, 5, and 6, the two operation methods are compared when the server is down. These figures ensure the expected behavior when blocking is applied, which results in a lower mean waiting time. This can be explained by the fact that customers cannot enter the system in case of server failure, such that these requests are rejected and directed back towards the source. These figures also show, besides higher values of α and β, that the difference between the operation methods are smaller; in Fig. 6, the results are almost identical.

Figures 7, 8, and 9 present the mean spent time in the source of the customers as a function of the server failure rates. As can be clearly seen, the results are equal, which indicates that the distribution of the service time has no effect on the mean spent time in the source of customers. The proposed behavior is noticeable in case of blocking: the customers spend more and more time in the source if the server is

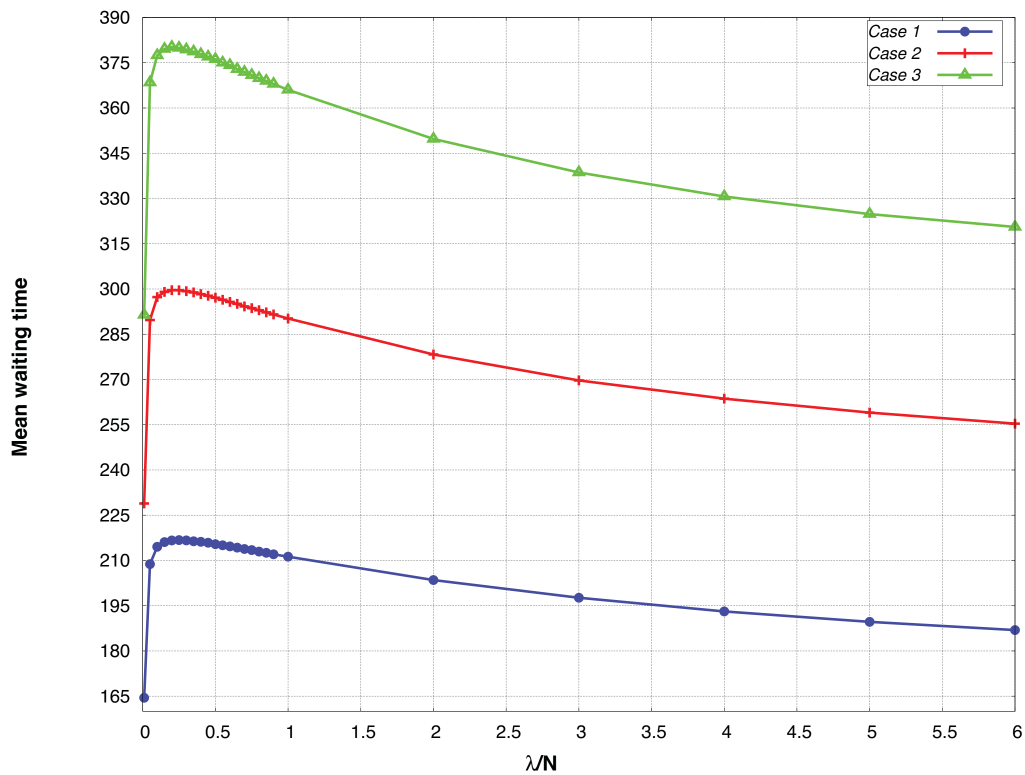

Fig. 3. Mean waiting time vs. intensity of incoming customers.

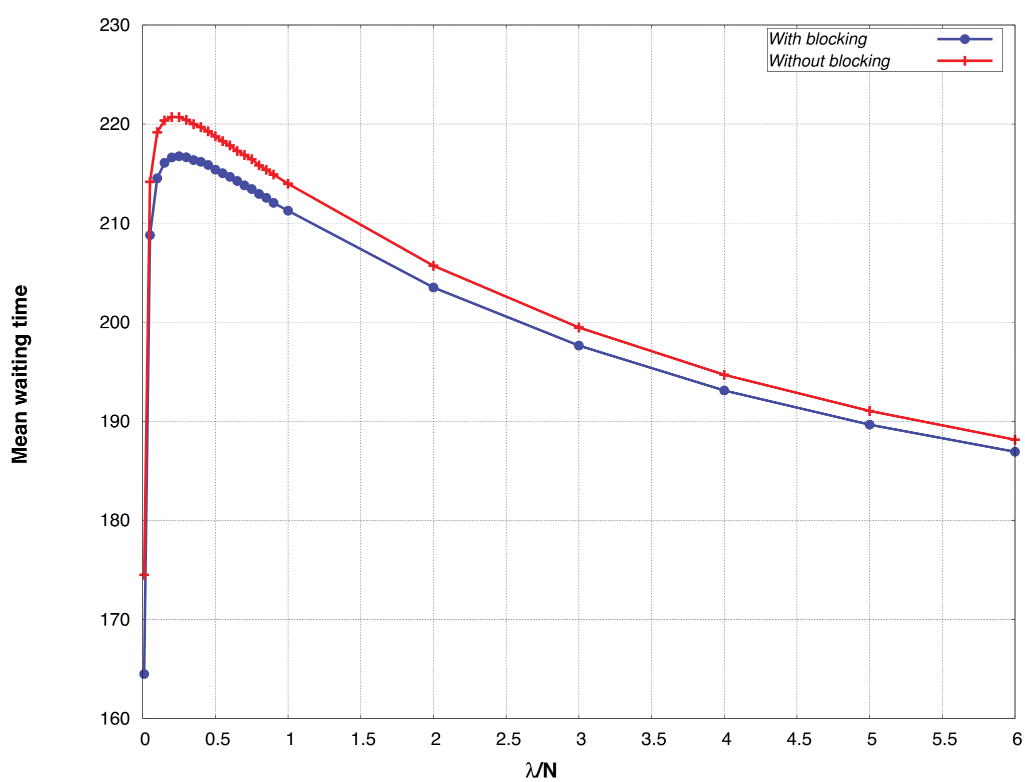

Fig. 4. The effect of blocking; the parameters of service time are $\alpha = \beta = 0.5$.

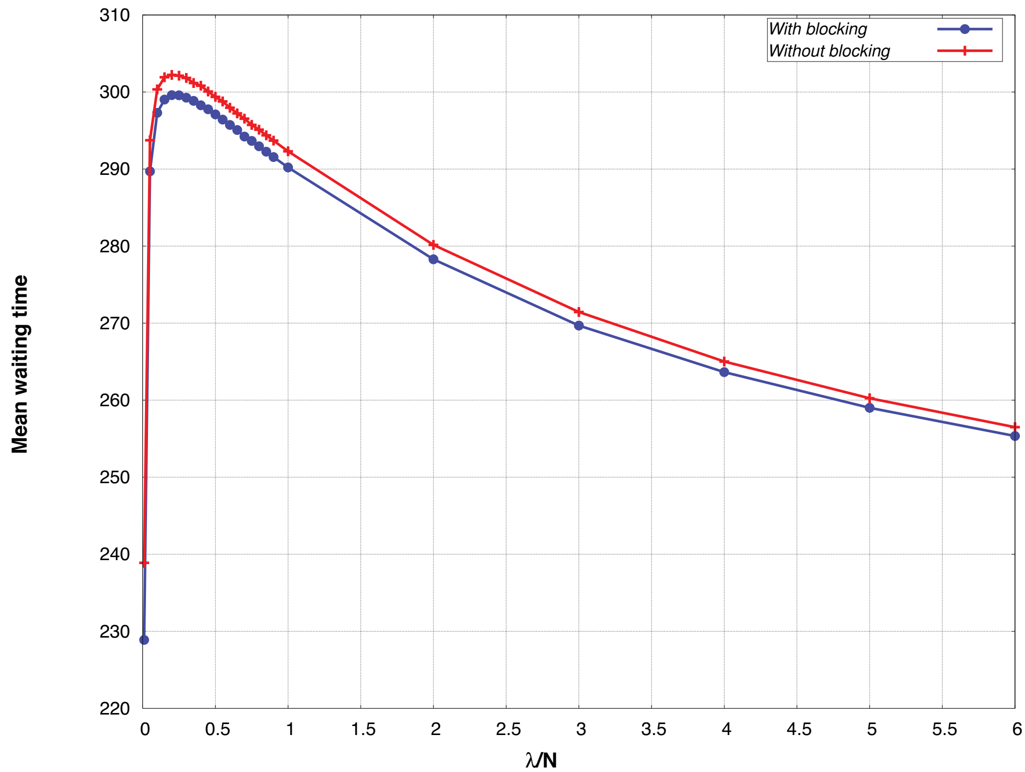

Fig. 5. The effect of blocking; the parameters of service time are $\alpha = \beta = 1$.

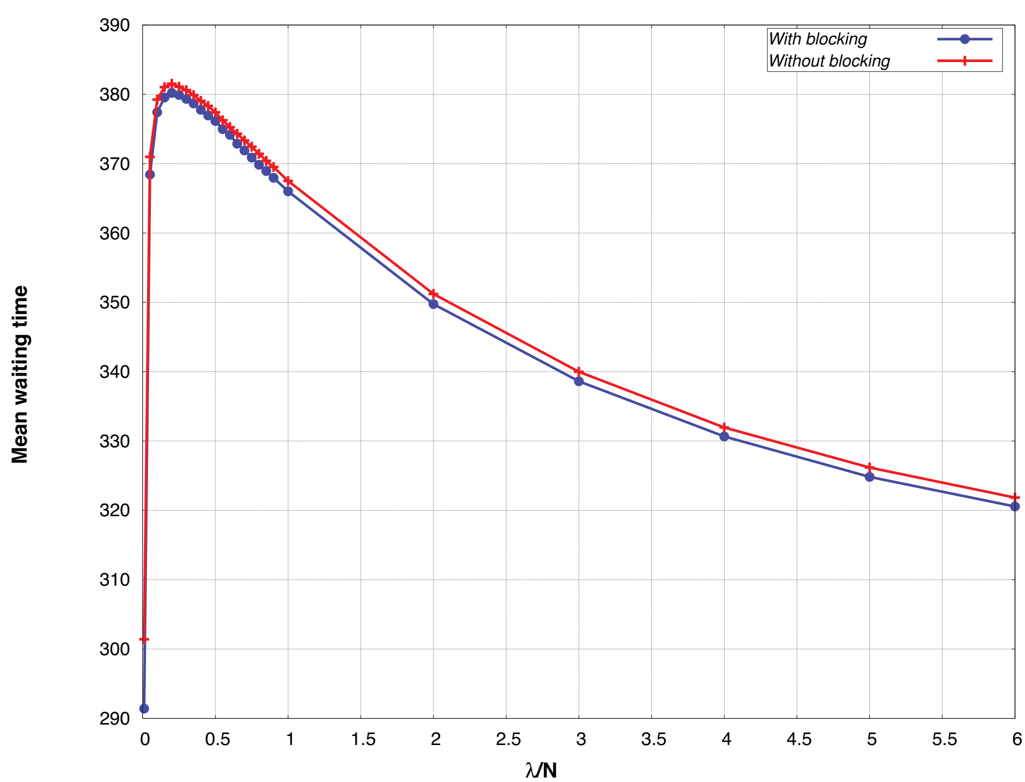

Fig. 6. The effect of blocking; the parameters of service time are $\alpha = \beta = 2$.

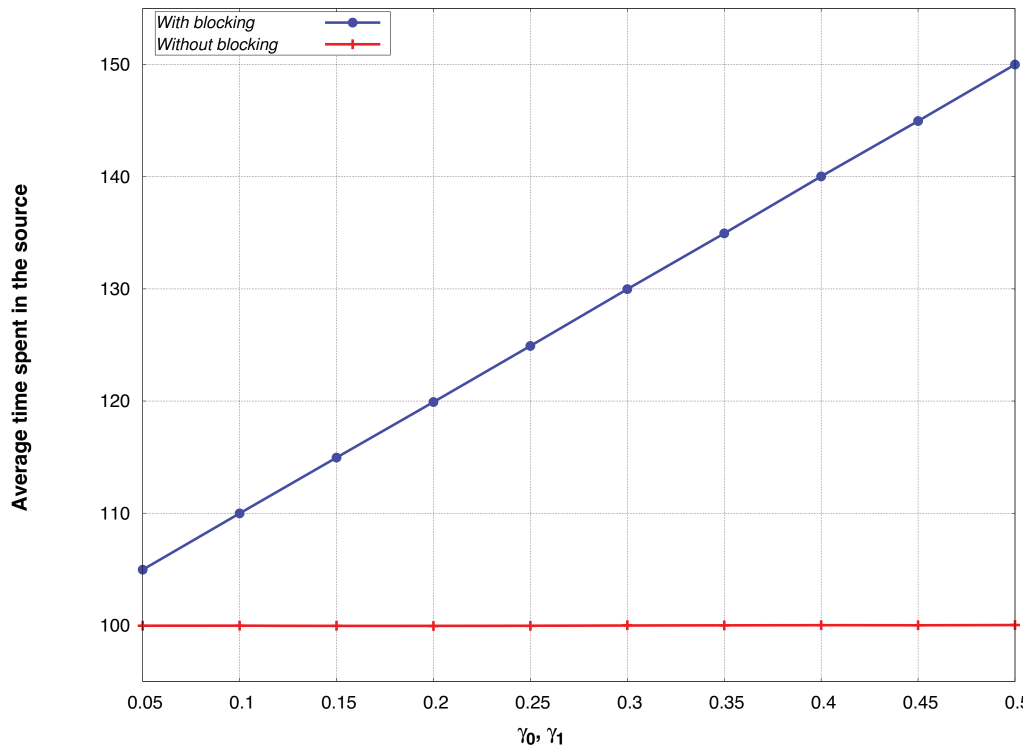

Fig. 7. The effect of blocking; the parameters of service time are $\alpha = \beta = 0.5$.

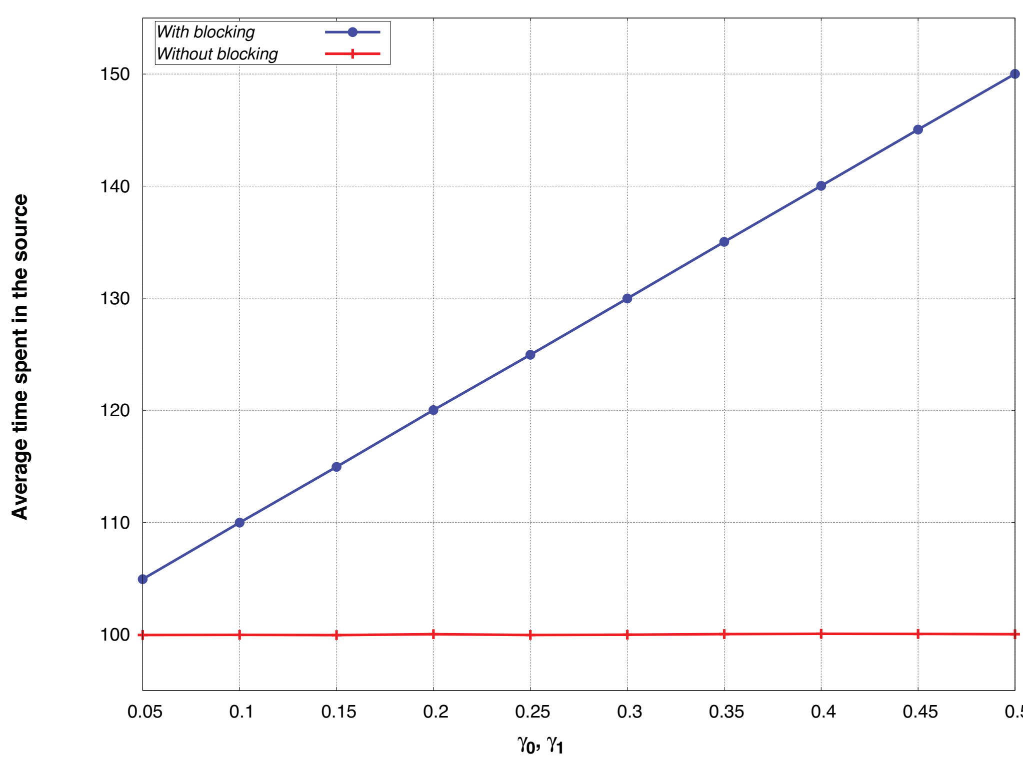

Fig. 8. The effect of blocking; the parameters of service time are $\alpha = \beta = 1$.

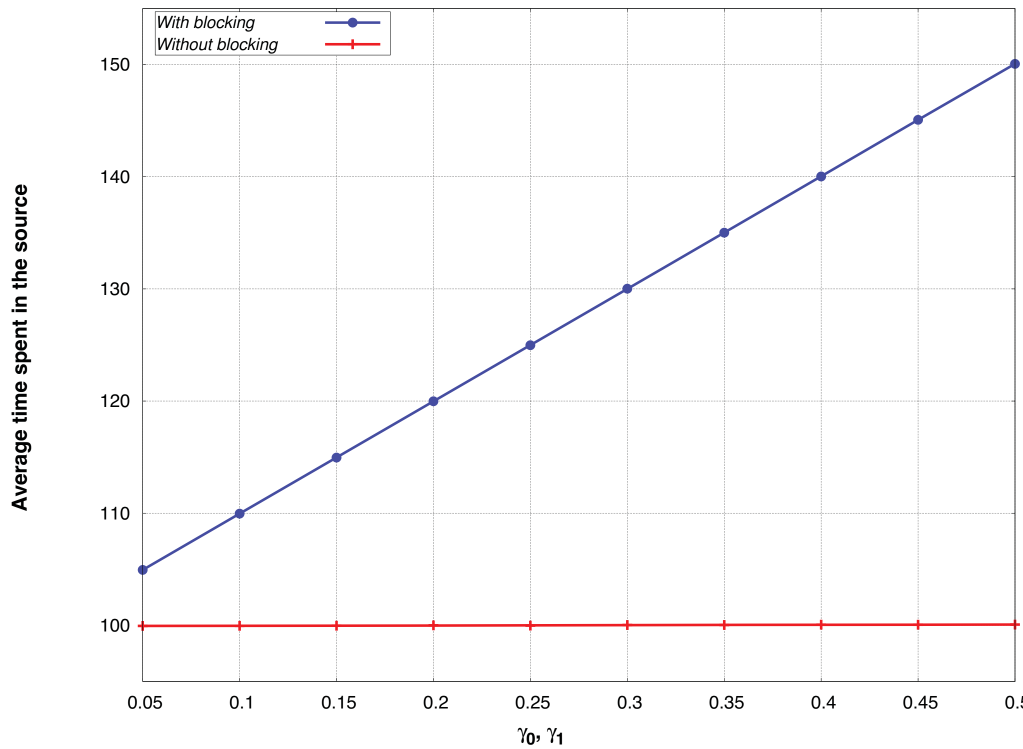

Fig. 9. The effect of blocking; the parameters of service time are $\alpha = \beta = 2$.

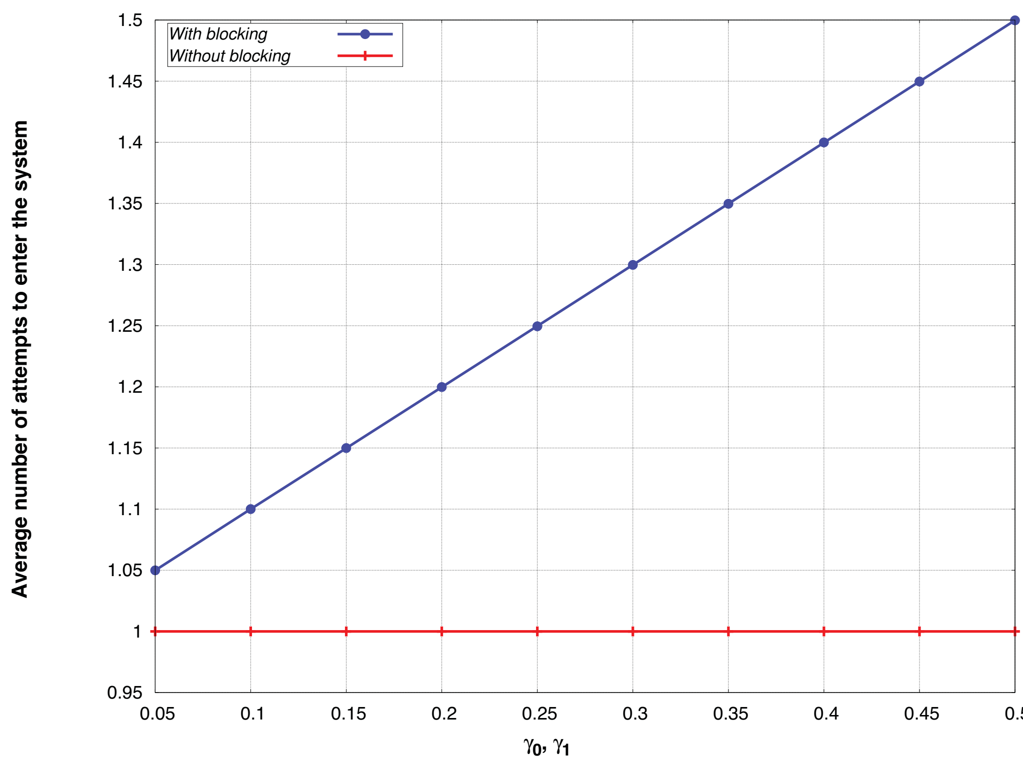

Fig. 10. The effect of blocking; the parameters of service time are $\alpha = \beta = 0.5$.

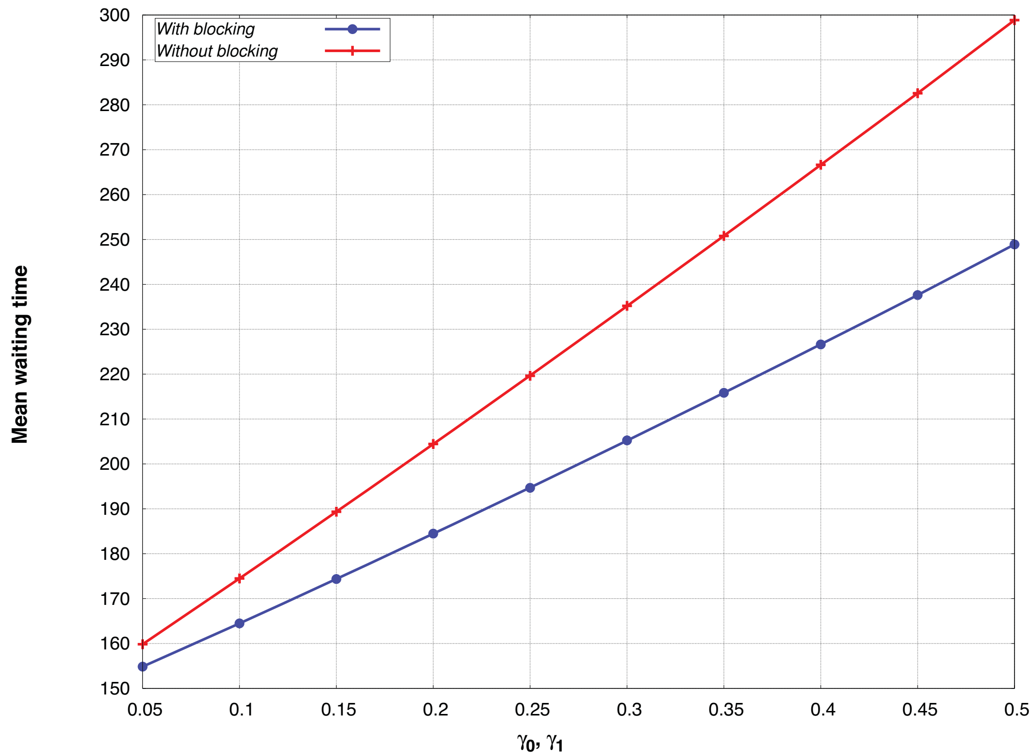

Fig. 11. The effect of blocking; the parameters of service time are $\alpha = \beta = 0.5$.

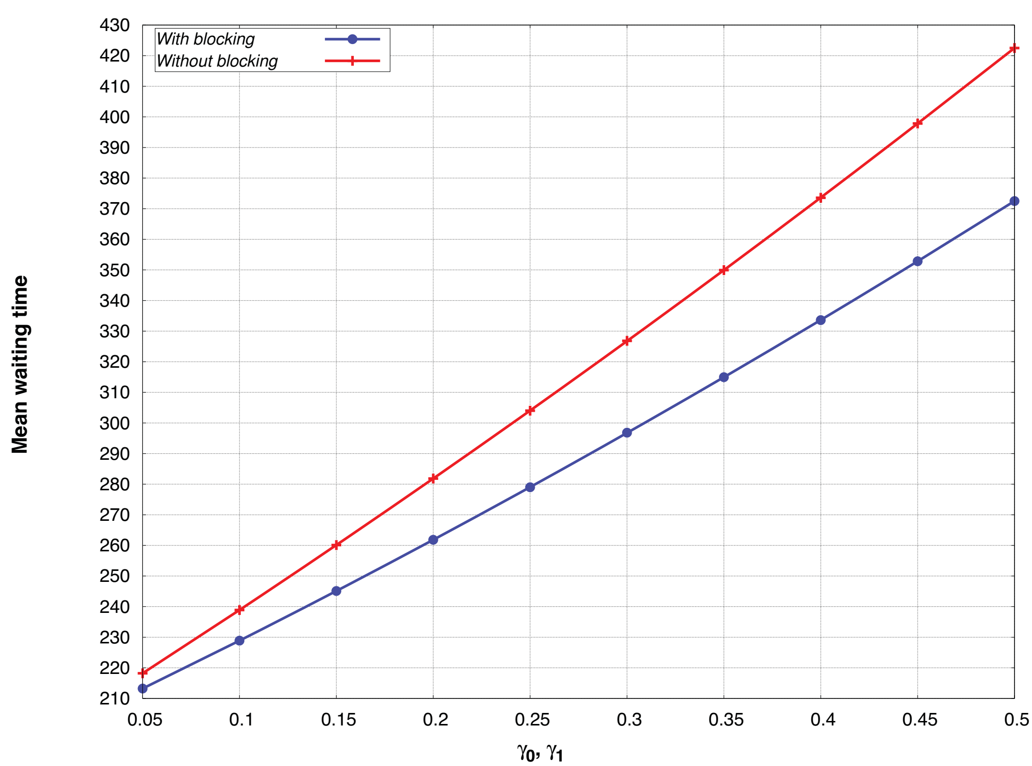

Fig. 12. The effect of blocking; the parameters of service time are $\alpha = \beta = 1$.

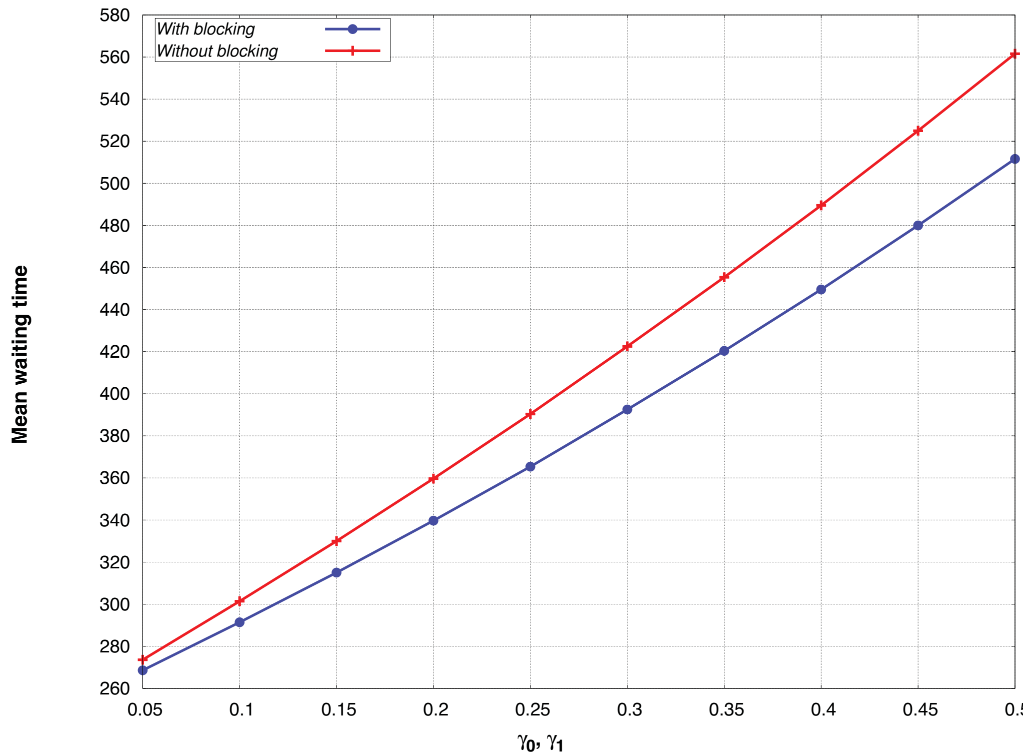

Fig. 13. The effect of blocking; the parameters of service time are $\alpha = \beta = 2$.

more likely subjected to breakdowns. The mean spent time in the source of the customers is constant when requests can enter the system and the server does not function.

Figure 10 shows the average number of attempts of the customers to enter the system. Consequently, if the service unit spends more time in a state of failure, then customers will be rejected more and more times when blocking is applied. Without blocking, the customer needs to appear once at the system and can enter irrespectively of the availability of the server. In the case of blocking, it is not surprising that by increasing parameters γ_0 and γ_1 the average number of attempts increases as well.

Figures 11, 12, and 13 demonstrate the mean waiting time of the customers as a function of γ_0 and γ_1. In all of the three figures, higher parameters of server failure rates result in a greater mean waiting time in both operation methods, but especially in the case of non-blocking. A significant difference can be noted among the values of the mean waiting time using various sets of parameters for the service time. An increasing tendency is observed with the increment in values of α and β. The same phenomenon takes place as in the previous figures in connection with the mean waiting time in that when $\alpha = \beta = 2$, the cases of non-blocking and blocking are closer to each other compared to $\alpha = \beta = 0.5$.

4. Conclusions

A retrial queuing model is presented with finite source and an unreliable server where collisions can occur. The goal of this paper was to provide a sensitivity analysis using various distributions. The results of the blocking case are compared with the results of the non-blocking case. From these, it is evident that when blocking is applied, the customers spend more time in the source instead of the system, resulting in lower mean waiting times. With the help of SimPack the effect of disparate distributions of service times on several main performance measures was investigated.

Acknowledgments

This research was supported by the Austrian-Hungarian Bilateral Cooperation in Science and Technology project 2017-2.2.4-TeT-AT-2017-00010.

REFERENCES

1. A. A. Ali and S. Wei, "Modeling of coupled collision and congestion in finite source wireless access systems," in: *Wireless Communications and Networking Conference (WCNC)* (2015), pp. 1113–1118.
2. B. Almási, J. Roszik, and J. Sztrik, "Homogeneous finite-source retrial queues with server subject to breakdowns and repairs," *Math. Comput. Model.*, **42**, No. 5–6, 673–682 (2005).
3. J.R. Artalejo and A. Gómez-Corral, *Retrial Queueing Systems: a Computational Approach*, Springer, Berlin (2008).
4. S. Balsamo, G.L. Dei Rossi, and A. Marin, "Modelling retrial-upon-conflict systems with product-form stochastic Petri nets," in: *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, Springer (2013), pp. 52–66 .
5. V.I. Dragieva, "Number of retrials in a finite source retrial queue with unreliable server," *Asia-Pac. J. Oper. Res.*, **31**, No. 2 (2014).
6. V.I. Dragieva and T. Phung-Duc, "Two-way communication M/M/1//N retrial queue," in: *International Conference on Analytical and Stochastic Modeling Techniques and Applications*, Springer (2017), pp. 81–94.
7. P. A. Fishwick, "Simpack: Getting started with simulation programming in C and C++," in: *WSC 92 Proceedings of the 24th Conference on Winter Simulation*, ACM, New York (1992), pp. 154–162.
8. N. Gharbi and C. Dutheillet, "An algorithmic approach for analysis of finite-source retrial systems with unreliable servers," *Comput. Math. Appl.*, **62**, No. 6, 2535–2546 (2011).
9. J. Kim and B. Kim, "A survey of retrial queueing systems," *Ann. Oper. Res.*, **247**, No. 1, 3–36 (2016).
10. J. S. Kim, "Retrial queueing system with collision and impatience," *Commun. Korean Math. Soc.*, **25**, No. 4, 647–653 (2010).
11. A. Krishnamoorthy, P. K. Pramod, and S. R. Chakravarthy, "Queues with interruptions: a survey," *TOP*, **22**, No. 1, 290–320 (2014).
12. A. Kuki, J. Sztrik, Á. Tóth, and T. Bérczes, "A contribution to modeling two-way communication with retrial queueing systems," *Commun. Comput. Inform. Sci.*, **912**, 236–247 (2018).
13. A. Kvach and A. Nazarov, "Sojourn time analysis of finite source Markov retrial queuing system with collision," *Commun. Comput. Inform. Sci.*, **564**, 64–72 (2015).
14. A. Nazarov, A. Kvach, and V. Yampolsky, "Asymptotic analysis of closed Markov retrial queuing system with collision," *Commun. Comput. Inform. Sci.*, **487**, 334–341 (2014).
15. A. Nazarov, J. Sztrik, A. Kvach, and T. Bérczes, "Asymptotic analysis of finite-source M/M/1 retrial queueing system with collisions and server subject to breakdowns and repairs," *Ann. Oper. Res.*, **277**, No. 2, 213–239 (2018).
16. A. Nazarov, J. Sztrik, and A. Kvach, "A survey of recent results in finite-source retrial queues with collisions," *Commun. Comput. Inform. Sci.*, **912**, 1–15 (2018).
17. Y. Peng, Z. Liu, and J. Wu, "An M/G/1 retrial G-queue with preemptive resume priority and collisions subject to the server breakdowns and delayed repairs," *J. Appl. Math. Comput.*, **44**, No. 1–2, 187–213 (2014).

18. J. Sztrik, "Tool supported performance modelling of finite-source retrial queues with breakdowns," *Publ. Math.*, **66**, No. 1, 187–213 (2014).

19. Á. Tóth, T. Bérczes, J. Sztrik, and A. Kuki, "Comparison of two operation modes of finite-source retrial queueing systems with collisions and a non-reliable server by using simulation," *J. Math. Sci.*, **237**, No. 6, 846–857 (2019).

20. J. Wang, L. Zhao, and F. Zhang, "Analysis of the finite source retrial queues with server breakdowns and repairs," *J. Ind. Manag. Optim.*, **7**, No. 3, 655–676 (2011).

21. F. Zhang and J. Wang, "Performance analysis of the retrial queues with finite number of sources and service interruptions," *J. Korean Stat. Soc.*, **42**, No. 1, 117–131 (2013)