Journal of Mathematical Sciences, Vol. 146, No. 4, 2007

PERFORMANCE ANALYSIS OF FINITE-SOURCE RETRIAL QUEUES WITH NONRELIABLE HETEROGENOUS SERVERS*

J. Roszik [1] **and J. Sztrik** [2]

UDC 519.2

1. Introduction

Retrial queueing models are often used for the performance and reliability modeling of computer systems and communication networks. The reason is that the return of customers plays a special role in many of these systems as well as in other practical applications, and it has a nonneglectable negative effect on the performance measures. For some applications of retrial queues, see, for example, [1–4], and for some fundamental results on finite-source retrial queueing systems, refer to [5–9].

Usually, the components of computer systems are subject to random breakdowns, which has a substantial influence on the performance measures, so it is of practical importance to investigate nonreliable retrial queueing systems, too. Nonreliable, infinite-source retrial queues were studied in [10–12] and finite-source retrial queues with a single nonreliable server were studied in [13].

The purpose of this paper is to generalize the model of [9, 13] and to give the main stationary performance measures of the nonreliable multiserver model described in the next section. Furthermore, our aim is to illustrate graphically the effect of the nonreliability of the servers on the steady-state systems's measures.

Because of the fact that the state space of the Markov chain described is very large, it is difficult to calculate the system measures in the traditional way of solving the system of steady-state equations. To simplify this procedure, we used the software tool MOSEL (Modeling, Specification, and Evaluation Language) (see [14]) to formulate the model and to obtain the performance measures. With the help of MOSEL, we can use various performance tools (such as SPNP — Stochastic Petri Net Package) to get these characteristics. The results of the tool can graphically be displayed using IGL (Intermediate Graphical Language), which is a part of MOSEL.

The organization of the paper is as follows. Section 2 contains an accurate description of the investigated retrial queueing model and the derivation of the main steady-state performance measures. Section 3 is devoted to the validation of the results of the tool and some graphically displayed numerical results. The paper ends with Comments and Conclusions.

2. The $M\,|\,\vec{M}\,|\,c\,|\,|\,K$ Retrial Queueing Model with Nonreliable Servers

Consider a finite-source retrial queueing system with c servers, where the primary calls are generated by K, $c < K < \infty$, sources. Each server can be in operational (up) or nonoperational (down) states and can be idle or busy. Each source can be in three states: generating a primary call (free), sending repeated calls, and under service by one of the servers. If a source is free at time t, it can generate a primary call during the interval $(t, t + dt)$ with probability $\lambda\,dt + o(dt)$. If one of the servers is up and idle at the moment of arrival of the call, then the service of the call starts. At the arrival of the calls, the availability and idleness of the servers are always examined according to the increasing order of the server indices, resulting in different loads to the servers. The service is finished during the interval $(t, t+dt)$ with probability $\mu_i\,dt + o(dt)$ if the ith server is available.

Server i can fail during the interval $(t, t + dt)$ with probability $\delta_i\,dt + o(dt)$ if it is idle and with probability $\gamma_i\,dt + o(dt)$ if it is busy. If $\delta_i = 0$, $\gamma_i > 0$ or $\delta_i = \gamma_i > 0$, active or independent breakdowns can be discussed, respectively. If the server fails in a busy state, it either continues servicing the interrupted call after it has been repaired or the interrupted request returns to the orbit. In this paper, we only investigate the case where the source moves into the sending-repeated-calls state at the moment of server failure. The repairman follows the FIFO discipline for the server breakdowns, and the repair time of the ith server is exponentially distributed with a finite mean $1/\tau_i$. If all of the servers have failed, two different cases can be treated. Namely, the blocked-sources case, in which all operations are

*Partially supported by the Hungarian Scientific-Research Fund (grant No. OTKA T0-34280/2000) and FKFP (grant No. 0191/2001).

Proceedings of the Seminar on Stability Problems for Stochastic Models, Jurmala, Latvia, 2004, Part I.

stopped except for the repair of the servers. In the unblocked (intelligent) sources case, only service is interrupted but all other operations are continued.

If all of the servers are busy (or have failed in the unblocked case) at the moment of arrival of a call, the source begins the generation of a Poisson flow of repeated calls at a rate ν until it finds an available free server. After service, the source becomes free, and it can generate a new primary call, and the server becomes idle so it can serve a new call. All of the times involved in the model are assumed to be mutually independent of each other.

The state of the system at time t can be described by the process $X(t) = (\alpha_1(t), \dots, \alpha_c(t); N(t))$, where $N(t)$ is the number of sources of repeated calls and $\alpha_i(t)$, $i = 1, \dots, c$, denotes the state of the ith server at time t. If there is a customer under service at the ith server, $\alpha_i(t) = 1$, if it is operational and idle, then $\alpha_i(t) = 0$; otherwise the server has failed and $\alpha_i(t) = -1$.

Because of the exponentiality of the random variables involved, this process is a Markov chain with a finite state space. Since the state space of the process $(X(t), t \geq 0)$ is finite, the process is ergodic for all reasonable values of the rates involved in the model construction. From now on, we assume that the system is in the steady state.

Let us define the stationary probabilities by

$$\mathbf{P}(i_1, \dots, i_c, j) = \lim_{t\to\infty} \mathbf{P}\{\alpha_1(t) = i_1, \dots, \alpha_c(t) = i_c, N(t) = j\}, \quad i_1, \dots, i_c = -1, 0, 1, \quad j = 0, \dots, K^*,$$

where

$$K^* = K - \sum_{i_k, i_k = 1} i_k.$$

Furthermore, let us denote by $C(t)$ the number of busy servers, by $A(t)$ the number of available servers at time t, and by

$$p_{kj} = \lim_{t\to\infty} \mathbf{P}\{C(t) = k, N(t) = j\}$$

the joint distribution of the number of busy servers and the number of repeated calls.

Once we have obtained the above-defined probabilities, the main steady-state system performance measures can be derived as follows:

• The probability that at least one server is available

$$A_S = \mathbf{P}\{\alpha_k > -1, k \in \{1, \dots, c\}\} = 1 - \sum_{j=0}^{K} \mathbf{P}(-1, \dots, -1, j).$$

• Mean number of sources of repeated calls

$$N = \mathbf{E}[N(t)] = \sum_{k=0}^{c} \sum_{j=1}^{K} j p_{kj} = \sum_{i_1, \dots, i_c} \sum_{j=1}^{K^*} j \mathbf{P}(i_1, \dots, i_c, j).$$

• Utilization of the kth server

$$U_k = \sum_{i_1, \dots, i_c, i_k = 1} \sum_{j=0}^{K^*} \mathbf{P}(i_1, \dots, i_c, j), \quad k = 1, \dots, c.$$

• Mean number of busy servers

$$C = \mathbf{E}[C(t)] = \sum_{k=1}^{c} U_k.$$

• Mean number of calls staying in the orbit or in service

$$M = \mathbf{E}[N(t) + C(t)] = N + C.$$

• Utilization of the repairman

$$U_R = \sum_{\substack{i_1, \dots, i_c \\ -1 \in \{i_1, \dots, i_c\}}} \sum_{j=0}^{K^*} \mathbf{P}(i_1, \dots, i_c, j).$$

• Utilization of the sources

$$U_{SO} = \begin{cases} \dfrac{\mathbf{E}[K - C(t) - N(t); A(t) > 0]}{K} & \text{for the blocked case,} \\ \dfrac{\mathbf{E}[K - C(t) - N(t)]}{K} & \text{for the unblocked case.} \end{cases}$$

• Overall utilization of the system

$$U_O = C + KU_{SO} + U_R.$$

• Mean rate of generation of primary calls

$$\overline{\lambda} = \begin{cases} \lambda\mathbf{E}[K - C(t) - N(t); A(t) > 0] & \text{for the blocked case,} \\ \lambda\mathbf{E}[K - C(t) - N(t)] & \text{for the unblocked case.} \end{cases}$$

• Mean waiting time

$$\mathbf{E}[W] = N/\overline{\lambda}.$$

• Mean response time

$$\mathbf{E}[T] = M/\overline{\lambda}.$$

3. Numerical Examples

In this section, we consider some numerical results in the case of homogeneous servers to illustrate graphically the influence of the nonreliable servers on the mean response time $\mathbf{E}[T]$ and on the overall system utilization U_O. We used the SPNP tool with MOSEL, which was able to handle a model with up to 126 sources. In this case, on a PC with a 1.1 GHz processor and 512 MB RAM, the running time with one parameter setup with 2 servers was approximately 1 sec. With 4 servers and 126 sources, in the blocked case it was 2 min 25 sec. The maximum number of servers that the program was able to calculate the system measures in an acceptable time was on this computer 6. With 6 servers and 10 sources, the program finished its run after 20 min, and with 20 sources after 1 hr 15 min.

3.1. Validation of results. The results in the reliable case were validated by the Pascal program given in [5]. In Table 1, we can see that the corresponding performance measures are very close to each other; they are the same at least up to the 8th decimal digit. In the case of nonreliable servers, the results were tested by the $M \mid M \mid 1 \mid\mid K$ retrial model with server breakdowns, which was studied in [13].

TABLE 1. Validations.

	MOSEL	Pascal program [5]
Number of servers:	2	2
Number of sources:	5	5
Request-generation rate:	0.1	0.1
Service rate:	1	1
Retrial rate:	1.1	1.1
Server's failure rate:	$1E-25$	–
Server's repair rate:	$1E+25$	–
Mean waiting time:	0.0653833701	0.0653833729
Mean number of busy servers:	0.4518596260	0.4518596267
Mean number of sources of repeated calls:	0.0295441060	0.0295441065

In what follows, the effect of some parameters is illustrated. The system parameters of the figures are collected in Table 2.

TABLE 2. Input system parameters.

	c	K	λ	μ	ν	δ, γ	τ
Fig. 1	2	5	x axis	1	1.1	0.001	0.01
Fig. 2	2	5	0.2	1	x axis	0.001	0.01
Fig. 3	2	5	0.2	x axis	1.1	0.001	0.01
Figs. 4 and 5	2	5	0.2	1	1.1	x axis	0.01

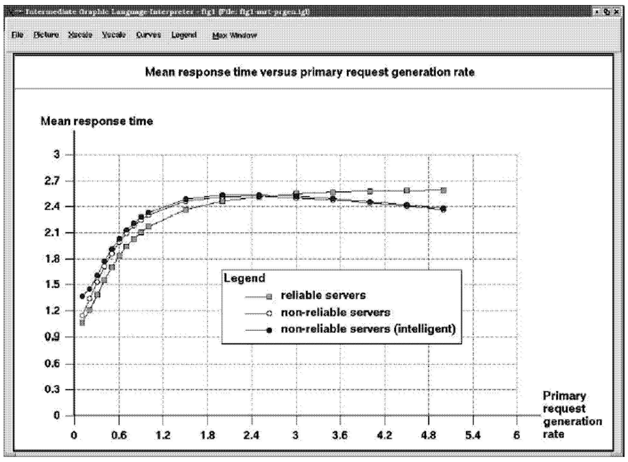

Fig. 1. $\mathbf{E}[T]$ versus primary request generation rate.

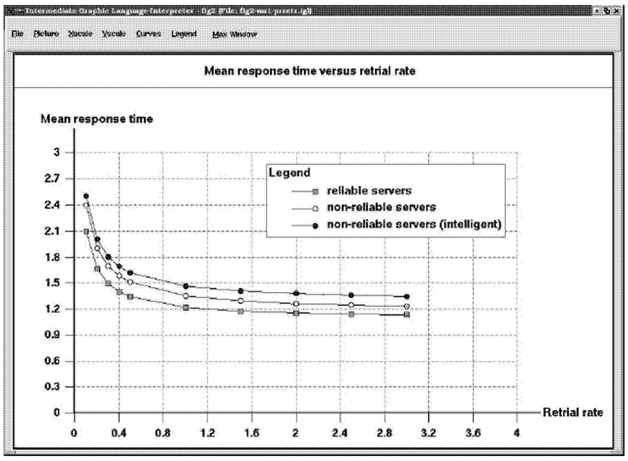

Fig. 2. $\mathbf{E}[T]$ versus retrial rate.

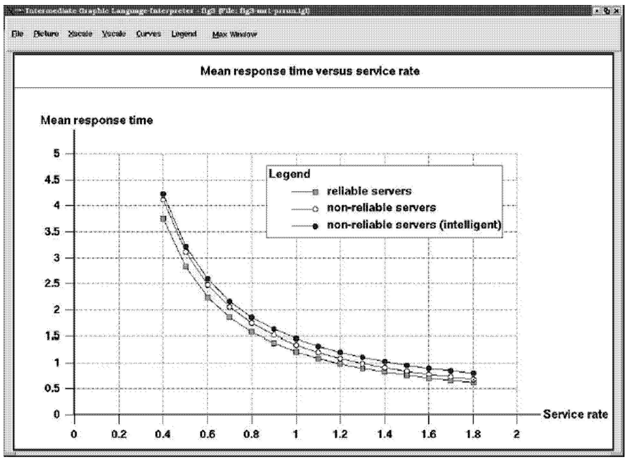

Fig. 3. $\mathbf{E}[T]$ versus service rate.

In Figs. 1–4, the effects of the primary request generation rate, retrial rate, service rate, and server's failure rate on the mean response time are displayed. In Fig. 5, we can see the effect of the server's failure rate on the overall utilization. In each figure, the reliable case and the blocked and unblocked (intelligent) cases are illustrated.

4. Comments

In Fig. 1, we can see that with these parameter setups the difference is very small between the nonintelligent and intelligent cases. An interesting phenomenon, which was mentioned in [7] too, that is, that retrial queues have a

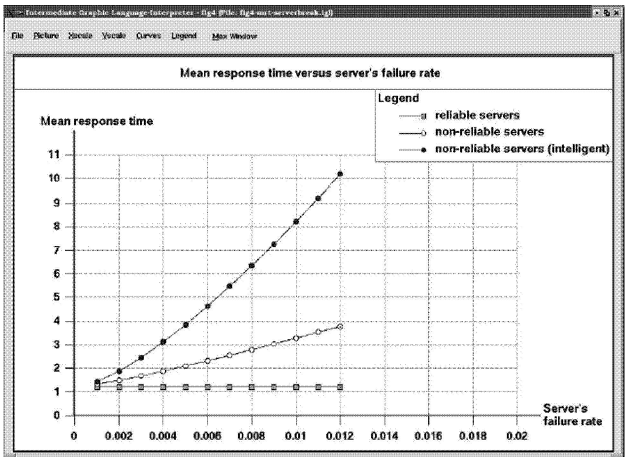

Fig. 4. $\mathbf{E}[T]$ versus server's failure rate.

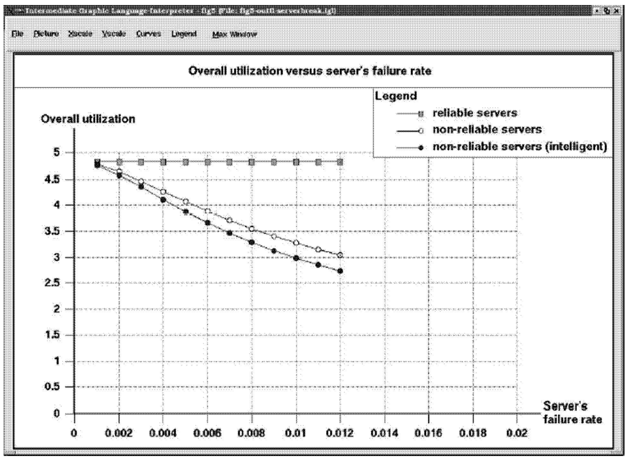

Fig. 5. U_O versus server's failure rate.

maximum of $\mathbf{E}[T]$, is also noted.

In Fig. 2, it is demonstrated how long the retrial rate has a significant influence on the mean response time.

In Fig. 3, we can see that an increase in the service rate has almost the same influence on the reliable and nonreliable systems.

In Fig. 4, it can be observed that an increase in the server failure rate can have a substantial impact on the mean response time, and as it increases the difference between the two nonreliable models increases significantly.

In Fig. 5, it is shown that the overall utilization can be very low if the server failure rate increases and the repair rate is not high enough.

5. Conclusions

In this paper, the performance of homogeneous, finite-source retrial queueing systems with nonreliable heterogeneous servers is studied. The novelty of the investigation is the nonreliability and heterogeneity of the servers. The MOSEL language and software package was used to formulate the model and to calculate the system performance measures, which were graphically displayed to show the effect of the nonreliability of the servers on the mean response times of the calls and on the overall system utilization.

REFERENCES

1. H. Li and T. Yang, "A single server retrial queue with server vacations and a finite number of input sources," *Europ. J. Oper. Res.*, **85**, 149–160 (1995).
2. G. K. Janssens, "The quasi-random input queueing system with repeated attempts as a model for collision-avoidance star local area network," *IEEE Trans. Comm.*, **45**, 360–364 (1997).

3. P. Tran-Gia and M. Mandjes, “Modeling of customer retrial phenomenon in cellular mobile networks,” *IEEE J. Select. Areas Comm.*, **15**, 1406–1414 (1997).
4. E. Onur, H. Delic, C. Ersoy, and M. U. Caglayan, “Measurement-based replanning of cell capacities in GSM networks,” *Comp. Net.*, **39**, 749–767 (2002).
5. G. I. Falin and J. G. C. Templeton, *Retrial Queues*, Chapman and Hall, London (1997).
6. J. R. Artalejo, “Retrial queues with a finite number of sources,” *J. Korean Math. Soc.*, **35**, 503–525 (1998).
7. G. I. Falin and J. R. Artalejo, “A finite source retrial queue,” *Europ. J. Oper. Res.*, **108**, 409–424 (1998).
8. J. R. Artalejo, “Accessible bibliography on retrial queues,” *Math. Comp. Mod.*, **30**, 1–6 (1999).
9. G. I. Falin, “A multiserver retrial queue with a finite number of sources of primary calls,” *Math. Comp. Mod.*, **30**, 33–49 (1999).
10. J. R. Artalejo, “New results in retrial queueing systems with breakdown of the servers,” *Statist. Neerlandica*, **48**, 23–36 (1994).
11. A. Aissani and J. R. Artalejo, “On the single server retrial queue subject to breakdowns,” *Queue. Syst.*, **30**, 309–321 (1998).
12. J. Wang, J. Cao, and Q. Li, “Reliability analysis of the retrial queue with server breakdowns and repairs,” *Queue. Syst.*, **38**, 363–380 (2001).
13. B. Almási, J. Roszik, and J. Sztrik, “Homogeneous finite-source retrial queues with server subject to breakdowns and repairs,” *Comp. Math. Appl.* (submitted for publication).
14. K. Begain, G. Bolch, and H. Herold, *Practical Performance Modeling, Application of the MOSEL Language*, Kluwer Academic, Boston (2001).

[1]*Department of Informatics Systems and Networks, University of Debrecen, P.O. Box 12, H-4010 Debrecen, Hungary. E-mail: jroszik@inf.unideb.hu.*

[2]*Department of Informatics Systems and Networks, University of Debrecen, P.O. Box 12, H-4010 Debrecen, Hungary. E-mail: jsztrik@inf.unideb.hu.*