6th International Conference on Applied Informatics
Eger, Hungary, January 27–31, 2004.

The effect of server's breakdown on the performance of finite-source retrial queueing systems *

János Roszik, János Sztrik

Department of Informatics Systems and Networks, University of Debrecen
e-mail: {jroszik,jsztrik}@inf.unideb.hu

Abstract

In this paper single server homogeneous finite-source retrial queueing systems are investigated. The server is assumed to be subject to random breakdowns depending on whether it is busy or idle. The failure of the server may block or unblock the system's operations and the service of the interrupted request may be resumed or the call can be transmitted to the orbit. All random variables involved in the model constructions are supposed to be exponentially distributed and independent of each other.
The novelty of investigations is the different type of non-reliability of the server. The MOSEL (Modeling, Specification and Evaluation Language) tool, developed at the University of Erlangen, Germany, was used to formulate and solve the problem and the main performance and reliability measures were derived and graphically displayed. Several numerical calculations were performed to show the effect of the non-reliability of the server on the mean response times of the calls, the overall utilization of the system, and the mean number of calls staying at the server or in the orbit.

Categories and Subject Descriptors: C.4 [Performance of Systems]: Modeling Techniques, Performance Attributes; G.3 [Probability and Statistics]: Queueing Theory, Stochastic Processes; I.6.4 [Model Validation and Analysis];

Key Words and Phrases: Retrial queueing systems, finite number of sources, non-reliable server, performance tool, performance and reliability measures

1 Introduction

Retrial queues with quasi-random input are recent interest in modeling of magnetic disk memory systems [8], cellular mobile networks [9], and local-area networks with nonpersistent CSMA/CD protocols [6] and star topology [5, 7]. Since in practice some components of the systems are subject to random breakdowns it is of basic importance to

*Research is partially supported by Hungarian Scientific Research Fund OTKA T0-34280/2000 and FKFP grant 0191/2001.

study reliability of retrial queues with server breakdowns and repairs because of limited ability of repairs and heavy influence of the breakdowns on the performance measures of the system. For related literature the reader is referred to the works [2, 3, 10] where infinite-source non-reliable retrial queues were treated.

In this paper, finite-source retrial queueing systems with the following assumptions are investigated. Consider a single server queueing system, where the primary calls are generated by K, $1 < K < \infty$ homogeneous sources. The server can be in three states: idle, busy and failed. If the server is idle, it can serve the calls of the sources. Each of the sources can be in three states: free, sending repeated calls and under service. If a source is free at time t it can generate a primary call during interval $(t, t + dt)$ with probability $\lambda dt + o(dt)$. If the server is free at the time of arrival of a call then the call starts to be served immediately, the source moves into the under service state and the server moves into busy state. The service is finished during the interval $(t, t + dt)$ with probability $\mu dt + o(dt)$ if the server is available. If the server is busy at the time of arrival of a call, then the source starts generation of a Poisson flow of repeated calls with rate ν until it finds the server free. 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. The server can fail during the interval $(t, t+dt)$ with probability $\delta dt+o(dt)$ if it is idle, and with probability $\gamma dt+o(dt)$ if it is busy. If $\delta = 0, \gamma > 0$ or $\delta = \gamma > 0$ *active or independent breakdowns* can be discussed, respectively. If the server fails in busy state, it either *continues servicing* the interrupted call after it has been repaired or the interrupted request *transmitted to the orbit*. The repair time is exponentially distributed with a finite mean $1/\tau$. If the server is failed two different cases can be treated. Namely, *blocked sources* case when all the operations are stopped, that is neither new primary calls nor repeated calls are generated. In the *unblocked (intelligent) sources* case only service is interrupted but all the other operations are continued (primary and repeated calls can be generated). All the times involved in the model are assumed to be mutually independent of each other.

Our objective is to continue the investigations which were started in [1] but because of page limitations only some results were presented. The mean number of requests staying in the orbit or in the service, overall utilization of the system and the mean response time of calls are displayed as the function of server's failure and repair rates. To achieve this goal a performance tool called MOSEL (Modeling, Specification and Evaluation Language), see [4], is used to formulate and solve the problem.

The paper is organized as follows. In Section 2 the full description of the model by the help of the corresponding Markov chain is given. Then, the main performance and reliability measures are derived that can be obtained using MOSEL tool. In Section 3 several numerical examples are presented and some comments are made. Finally, the paper ends with a Conclusion.

2 The $M/M/1//K$ retrial queue with unreliable server

The system state at time t can be described with the process $X(t) = (Y(t); C(t); N(t))$, where $Y(t) = 0$ if the server is up, $Y(t) = 1$ if the server is failed, $C(t) = 0$ if the server is idle, $C(t) = 1$ if the server is busy, $N(t)$ is the number of sources of repeated calls at time t. Because of the exponentiality of the involved random variables 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, hence from now on we will assume that the system is in the steady state. We define the stationary probabilities:

$$P(q;r;j) = \lim_{t\to\infty} P(Y(t)=q, C(t)=r, N(t)=j), \;\; q=0,1, \;\; r=0,1, \;\; j=0,...,K^*,$$

$$\text{where } K^* = \begin{cases} K-1 & \text{for blocked case,} \\ K-r & \text{for unblocked case.} \end{cases}$$

Knowing these quantities the main performance measures can be obtained as follows:

- *Utilization of the server*

$$U_S = \sum_{j=0}^{K-1} P(0,1,j).$$

- *Utilization of the repairman*

$$U_R = \sum_{r=0}^{1}\sum_{j=0}^{K^*} P(1,r,j).$$

- *Availability of the server*

$$A_S = \sum_{r=0}^{1}\sum_{j=0}^{K^*} P(0,r,j) = 1 - U_R.$$

- *Mean number of calls staying in the orbit or in service*

$$M = E[N(t)+C(t)] = \sum_{q=0}^{1}\sum_{r=0}^{1}\sum_{j=0}^{K^*} jP(q,r,j) + \sum_{q=0}^{1}\sum_{j=0}^{K-1} P(q,1,j).$$

- *Utilization of the sources*

$$U_{SO} = \begin{cases} \frac{E[K-C(t)-N(t);Y(t)=0]}{K} & \text{for blocked case,} \\ \frac{K-M}{K} & \text{for unblocked case.} \end{cases}$$

- *Overall utilization*

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

- *Mean rate of generation of primary calls*

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

- *Mean response time*

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

3 Numerical examples

In this section we consider some sample numerical results to illustrate graphically the influence of the non-reliable server on the mean response time, overall utilization of the system and mean number of calls staying in the orbit or in the service. In each case the independent failure is considered and different comparisons are made according to service continuation *(resumed, transmitted)* and system operations *(blocked, unblocked)*.

In Figures 1–3 we can see the mean response time, the overall utilization of the system and mean number of calls staying in the orbit or in the service for the reliable and the non-reliable retrial system when the server's failure rate increases. In Figures 4–6 the same performance measures are displayed as the function of increasing repair rate. The input parameters are collected in Table 1.

	K	λ	μ	ν	δ, γ	τ
Figure 1	6	0.8	4	0.5	x axis	0.1
Figure 2	6	0.1	0.5	0.5	x axis	0.1
Figure 3	6	0.1	0.5	0.05	x axis	0.1
Figure 4	6	0.8	4	0.5	0.05	x axis
Figure 5	6	0.05	0.3	0.2	0.05	x axis
Figure 6	6	0.1	0.5	0.05	0.05	x axis

Table 1: Input system parameters

3.1 Comments

In Figure 1, we can see that in the case when the request returns to the orbit at the breakdown of the server, the sources will have always longer response times. Although the difference is not considerable it increase as the failure rate increase. The almost linear increase in $E[T]$ can be explained as follows. In the blocked (non-intelligent) case the failure of the server blocks all the operations and the response time is the sum of the down time of the server, the service and repeated call generation time of the request (which does not change during the failure) thus the failure has a linear effect on this measure. In the intelligent case the difference is only that the sources send repeated calls during the server is unavailable, so this is not an additional time.

In Figure 2 and Figure 5 it is shown how much the overall utilization is higher in the intelligent case with the given parameters. It is clear that the continued cases have better utilizations, because a request will be at the server when it has been repaired.

In Figure 3 we can see that the mean number of calls staying in the orbit or in service does not depend on the server's failure rate in continuous, non-intelligent case, it coincides with the reliable case. It is because during and after the failure the number of requests in these states remains the same. The almost linear increase in the non-continuous, non-intelligent case can be explained with that if the server failure occurs more often the server will be idle more often after repair until a source repeats his call.

In Figure 4, we can see that if the request returns to the orbit at the breakdown of the server, the sources will have longer response times like in Figure 1. The difference is not considerable too, and as it was expected the curves converge to the reliable case.

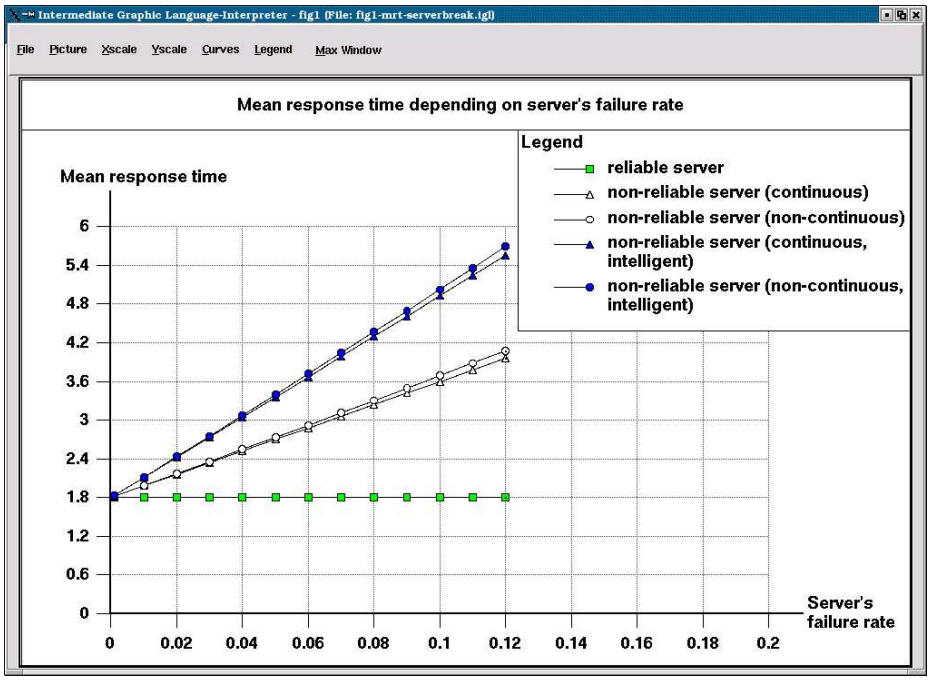

Figure 1: $E[T]$ versus server's failure rate

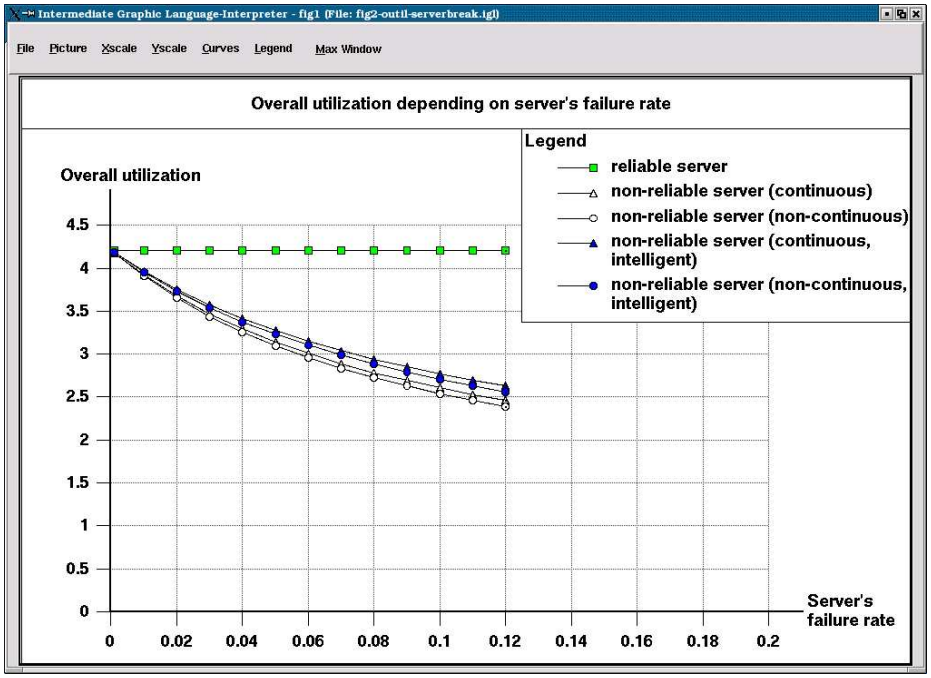

Figure 2: U_O versus server's failure rate

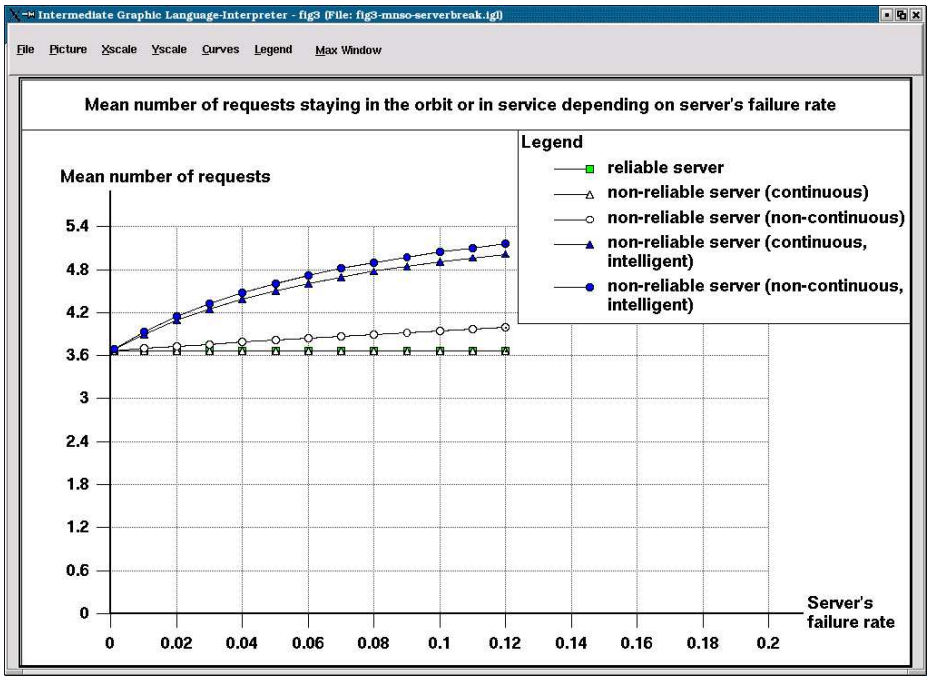

Figure 3: M versus server's failure rate

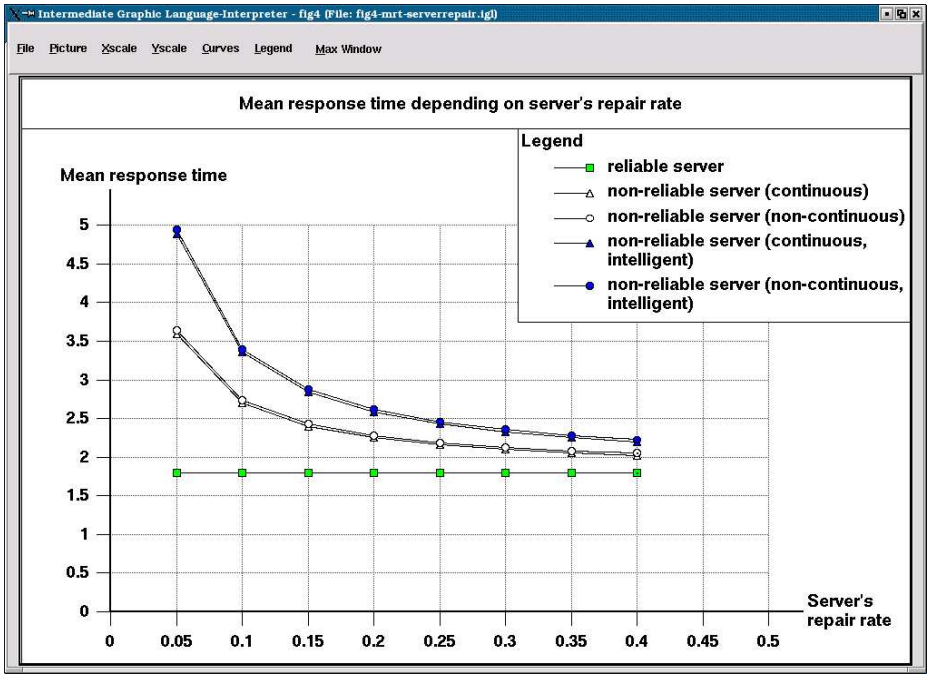

Figure 4: $E[T]$ versus server's repair rate

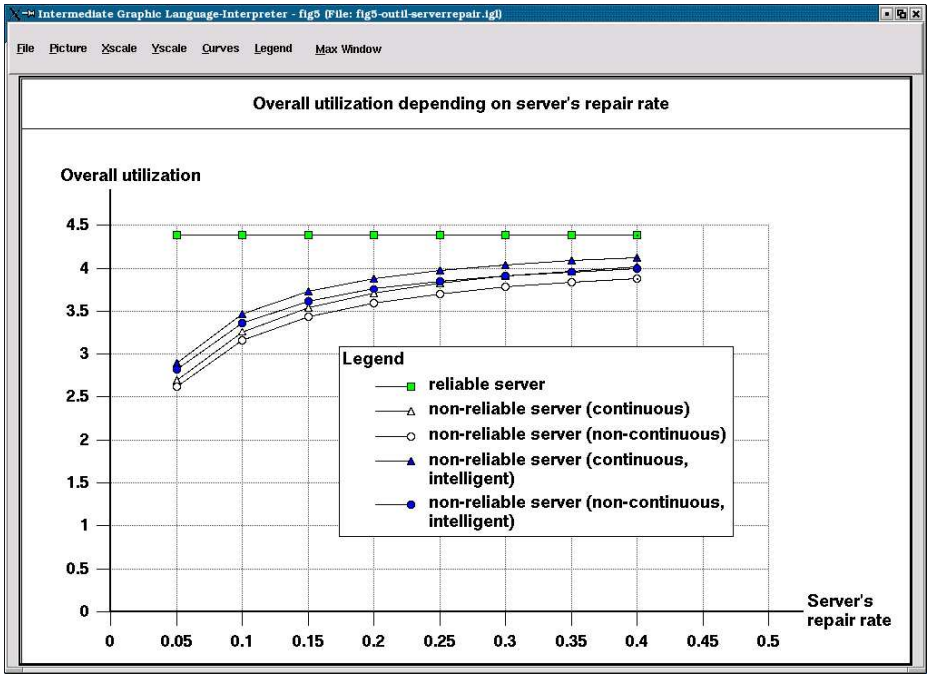

Figure 5: U_O versus server's repair rate

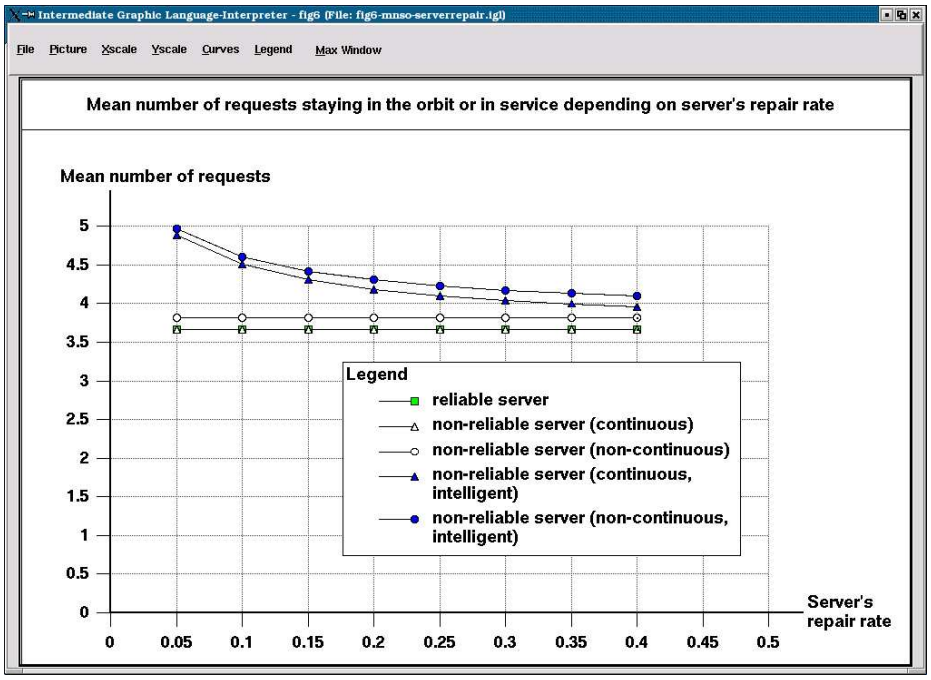

Figure 6: M versus server's repair rate

In Figure 6, it can be seen that the mean number of calls staying in the orbit or in service does not depend on the server's repair rate in continuous, non-intelligent case, it coincides with the reliable case like in Figure 3. It is true for the non-continuous, non-intelligent case too, which has more requests in the orbit on the average because of the non-continuity.

4 Conclusions

In this paper a finite-source homogeneous retrial queueing system is studied with the novelty of the non–reliability of the server. The MOSEL tool was used to formulate and solve the problem, and the main performance and reliability measures were derived and analyzed graphically. Several numerical calculations were performed to show the effect of server's breakdowns and repairs on the mean response times of the calls, on the overall utilization of the system and on the mean number requests staying in the orbit or in service.

References

[1] Almási B., Roszik J., and Sztrik J.: Homogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Computers and Mathematics wit Applications* (submitted for publication).

[2] Artalejo J.R.: New results in retrial queueing systems with breakdown of the servers, *Statistica Neerlandica*, Vol. 48 (1994), 23-36.

[3] Aissani A. and Artalejo J.R.: On the single server retrial queue subject to breakdowns, *Queueing Systems Theory and Applications*, Vol. 30 (1998), 309-321.

[4] Begain K., Bolch G. and Herold H.: *Practical performance modeling, application of the MOSEL language*, Kluwer Academic Publisher, Boston, 2001.

[5] Janssens G.K.: The quasi-random input queueing system with repeated attempts as a model for collision-avoidance star local area network, *IEEE Transactions on Communications*, Vol. 45 (1997), 360-364.

[6] Li H. and Yang T.: A single server retrial queue with server vacations and a finite number of input sources, *European Journal of Operational Research*, Vol. 85 (1995), 149-160.

[7] Mehmet-Ali M.K., Hayes J.F. and Elhakeem A.K.: Traffic analysis of a local area network with star topology, *IEEE Transactions on Communications*, Vol. 36 (1988), 703-712.

[8] Ohmura H. and Takahashi Y.: An analysis of repeated call model with a finite number of sources, *Electronics and Communications in Japan*, Vol. 68 (1985), 112-121.

[9] Tran-Gia P. and Mandjes M.: Modeling of customer retrial phenomenon in cellular mobile networks, *IEEE Journal of Selected Areas in Communications*, Vol. 15 (1997), 1406-1414.

[10] Wang J., Cao J. and Li Q.L.: Reliability analysis of the retrial queue with server breakdowns and repairs, *Queueing Systems Theory and Applications*, Vol. 38 (2001), 363-380.

Postal addresses

János Roszik, János Sztrik
Department of Informatics Systems and Networks
Institute of Informatics, University of Debrecen
P.O. Box 12, H-4010 Debrecen
Hungary