UNIVERSITY OF DEBRECEN

PERFORMANCE ANALYSIS OF FINITE-SOURCE RETRIAL QUEUES
OPERATING IN RANDOM ENVIRONMENTS

JÁNOS ROSZIK, JÁNOS SZTRIK, JORMA VIRTAMO

PREPRINTS NO. 330
(TECHNICAL REPORT NO. 2005/4)

FACULTY OF INFORMATICS

2005

PERFORMANCE ANALYSIS OF FINITE-SOURCE RETRIAL QUEUES OPERATING IN RANDOM ENVIRONMENTS

J. Roszik⋆, J. Sztrik⋆, J. Virtamo‡

⋆University of Debrecen, P.O. Box 12, H-4010, Debrecen, Hungary

{jroszik,jsztrik}@inf.unideb.hu

‡Helsinki University of Technology, P.O. Box 3000, FIN-02015 HUT, Finland

jorma.virtamo@hut.fi

Abstract

This paper is concerned with the performance analysis of finite-source retrial queues with heterogeneous sources operating in random environments, that is, the system parameters are subject to randomly occurring fluctuations. All random variables involved in the model construction are assumed to be exponentially distributed with a parameter depending on the source index and on the state of the corresponding random environment. The novelty of the investigation is the involvement of the random environments, which makes the system rather complicated. The MOSEL tool is used to formulate and solve the problem and the main performance measures are derived and graphically displayed. Several numerical calculations are performed to show the effect of different input parameters on the mean response time of the requests.

Keywords: retrial queueing systems, finite number of heterogeneous sources, random environments, performance tool, performance measures.

1 Introduction

Queues with repeated attempts have been widely used to model many problems in telephone switching systems, computer and telecommunication networks. Retrial queueing systems are characterized by the feature that customers who find the server busy do not wait in a queue, but repeat their requests at pre-determined or random intervals. There is a large literature devoted to retrial queues and their applications in the modeling of computer and communication systems, for some examples see [5, 6, 8, 9, 10, 11]. Finite-source queueing systems operating in random environments, sometimes called Markov-modulated queues have been the interest of recent research, see for example [2, 7].

This paper deals with the performance analysis of a finite-source retrial queue with heterogeneous sources operating in random environments, that is, the system parameters are subject

[1]Research is partially supported by Finnish-Hungarian Bilateral Scientific Cooperation SF-19/03.

to randomly occurring fluctuations. Similar models without repeated attempts were treated in [1, 2]. We also use the MOSEL (Modeling, Specification and Evaluation Language) [4] to formulate the model and to calculate the performance measures. The use of this tool helps us to avoid the very difficult calculations due to the large state space of the describing Markov-chain. The aim of the present paper is to give more realistic models for finite-source retrial queues since the different request arrival, service and retrial rates are subject to random fluctuations.

The rest of the paper is organized as follows. In Section 2 the mathematical description of the model and the performance measures are given, and in Section 3 some numerical examples are presented. Sections 4 an 5 are devoted to the Comments and Conclusions, respectively.

2 The queueing model

Consider a finite-source queue with K sources and a single server, where each source has different parameters and the operation of the sources and the server is influenced by the state of a given random environment.

The server and the sources are collected into M independent groups ($1 \leq M \leq K+1$). The members of a group operate in a common random environment. The environmental changes are reflected in the values of the new and repeated call generation and in the values of the service rates. The members of group m are assumed to operate in a random environment governed by an ergodic Markov chain ($\xi_m(t); t \geq 0$) with state space ($1, ..., r_m$) and with transition density matrix

$$\left(\tau^{(m)}_{i_m j_m}, i_m, j_m = 1, ..., r_m, \tau^{(m)}_{i_m i_m} = -\sum_{k \neq i_m} \tau^{(m)}_{i_m k}\right).$$

The server can be in two states: idle and busy, and each of the sources can be in free, sending repeated calls and under service states. If source i (which is a member of group m) is free at time t and the environmental process $\xi_m(t)$ is in state j_m the probability that this source generates a new request during the time interval $(t, t+dt)$ is $\lambda_i(j_m)dt + o(dt)$, p=$1, ..., M$. If the server is free at the time of arrival of a call then the call starts to be served, that is the source moves into the under service state and the server moves into the busy state. Assuming that the server belongs to group 1 and the environmental process $\xi_1(t)$ is in state j_1 the probability that the service of the request originating from client i is completed in time interval $(t, t+dt)$ is $\mu_i(j_1)dt + o(dt)$. If the server is busy on arrival, then the source starts generation of a Poisson flow of repeated calls with rate $\nu_i(j_m)$ 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 and it can serve a new call. All random variables and the random environments are supposed to be independent of each other.

Because of the exponentiality of the involved random variables the following process will be a Markov chain. The state of the system at time t can be described by the process

$$X(t) = (\xi_1(t), ..., \xi_M(t), \alpha(t), \beta_1(t), ..., \beta_{N(t)}(t)),$$

where $\xi_m(t)$ denotes the states of the background processes (p=$1, ..., M$), and $N(t)$ is the number of sources of repeated calls at time t. The index of the source at the server is denoted by $\alpha(t)$, if there is a customer under service, otherwise this value is 0. Because of the heterogeneity of the sources we need to identify the sources in the sending repeated calls state, so we

denote their indices by $\beta_k(t)$, k=$1, ..., N(t)$, if there is a customer in this state, otherwise this last component is 0.
Since its state space is finite the process $(X(t), t \geq 0)$ is ergodic with the following steady state probabilities.

$$P(j_1, ..., j_M, j, 0) = \lim_{t\to\infty} \mathrm{P}\{\xi_1(t) = j_1, ..., \xi_M(t) = j_M, \alpha(t) = j, N(t) = 0\}$$

$$P(j_1, ..., j_M, j, i_1, ..., i_k) =$$

$$\lim_{t\to\infty} \mathrm{P}\{\xi_1(t) = j_1, ..., \xi_M(t) = j_M, \alpha(t) = j, \beta_1(t) = i_1, ..., \beta_k(t) = i_k\}, \;\; k = 1, ..., K-1.$$

Based on the steady state probabilities the system performance measures can be obtained as:

- *Utilization of the server with respect to source i*

$$U_{Si} = \sum_{j_1,\ldots,j_M} P(j_1, ..., j_M, i, 0) + \sum_{j_1,\ldots,j_M} \sum_{k=1}^{K-1} \sum_{i_1,\ldots,i_k \neq i} P(j_1, ..., j_M, i, i_1, ..., i_k),$$
$$i = 1, ..., K.$$

- *Utilization of the server*

$$U_S = \sum_{i=1}^{K} U_{Si}.$$

- *Probability of source i is sending repeated calls*

$$N_i = \sum_{j_1,\ldots,j_M} \sum_{j=1}^{K} \sum_{k=1}^{K-1} \sum_{\substack{i_1,\ldots,i_k \neq j \\ i \in \{i_1,\ldots,i_k\}}} P(j_1, ..., j_M, j, i_1, ..., i_k), \quad i = 1, ..., K.$$

- *Mean number of repeated calls*

$$N = \sum_{i=1}^{K} N_i.$$

- *Utilization of source i*

$$U_i = 1 - U_{Si} - N_i, \quad i = 1, ..., K.$$

- *Probability of source i is free and its background process is in state j_l*

$$F_i(j_l) = \sum_{\substack{p_1,\ldots,p_M \\ p_l = j_l}} \sum_{\substack{j=1 \\ j \neq i}}^{K} \sum_{k=1}^{K-1} \sum_{\substack{i_1,\ldots,i_k \neq j \\ i \notin \{i_1,\ldots,i_k\}}} P(p_1, ..., p_M, j, i_1, ..., i_k), \quad i = 1, ..., K.$$

- *Throughput of source i*

$$\gamma_i = \sum_{j_l=1}^{r_l} F_i(j_l)\lambda_i(j_l), \quad i = 1, ..., K.$$

- *Mean response time of source* i

$$T_i = \frac{1 - U_i}{\gamma_i}, \quad i = 1, ..., K.$$

The traditional way is to derive the related Kolmogorov equations for the steady state probabilities and using the normalizing condition somehow we have to solve the set of equations. Usually it is not so easy, but in our case these two steps are performed by the help of the tool MOSEL as demonstrated in the next subsections.

3 Numerical examples

In this section we present the validation of the results and some graphically displayed numerical examples. The system parameters for the figures are given in Table 3.

3.1 Validation of results

The calculated performance measures are validated by the results of [1], where an FCFS (First-Come, First-Served) queueing model is studied. Some performance measures are collected in Table 1 and can be compared. We can see that we get back the results of the corresponding queueing model with waiting line, since with very high retrial rates and few sources the difference between the two models is negligible.

	FCFS [1]	retrial
Number of sources:	5	5
Request's generation rate:	0.12, 0.06	0.12, 0.06
Service rate:	1, 1	1, 1
Retrial rate:	–	1e+20
Rate of environment's change:	0.5, 1	0.5, 1
Server queue length:	0.6418	–
Requests in the orbit or in service:	–	0.6418567007
Utilization of the server:	0.4342	0.4342057023
Utilization of the sources:	0.8716	0.8716286599
Mean response time:	1.4782	1.4782319316

Table 1: Validation by the FCFS model

We checked the numerical calculations by the results of the retrial model with a non-reliable server [3], too. We used the model in which the sources are blocked if the server is not operational, and the server continues servicing the interrupted call after it has been repaired. In Table 2 we can see that the results are the same. It can easily be seen that non-reliable model can be considered as a system modulated by a 2-state background process. The system failure can be modeled by setting the rates to 10^{-20} in the second state of the background process.

	non-reliable retrial [3]	retrial in random env.
Number of sources:	5	5
Request's generation rate:	0.10, 0.15, 0.17, 0.19, 0.21	0.10, 0.15, 0.17, 0.19, 0.21
Service rate:	1.0, 1.1, 1.2, 1.5, 1.6	1.0, 1.1, 1.2, 1.5, 1.6
Retrial rate:	0.15, 0.18, 0.21, 0.22, 0.25	0.15, 0.18, 0.21, 0.22, 0.25
Server's failure/repair rate:	0.1, 1	-
Rate of env. change:	-	0.1, 1
Utilization of the server:	0.404265368271	0.404265368271
Utilization of the sources		
Source 1:	0.673069675362	0.673069675362
Source 2:	0.629766856845	0.629766856845
Source 3:	0.634124904383	0.634124904383
Source 4:	0.622974780687	0.622974780687
Source 5:	0.627325387151	0.627325387151
Mean response time		
Source 1:	5.34303316033	5.34303316033
Source 2:	4.31118758383	4.31118758383
Source 3:	3.73337662001	3.73337662001
Source 4:	3.50379767493	3.50379767493
Source 5:	3.11179039958	3.11179039958

Table 2: Validation by the non-reliable retrial model

In Figures 1 and 2 the mean response time is displayed as the primary request generation increases. The difference is that in Figure 1 all operations are stopped if the background process is in the second state, but in Figure 2, only service is interrupted. The results are in agreement with the results of [3], where the same parameters were used.

	K	$\lambda_1(1)...\lambda_5(1)$ $\lambda_1(2)...\lambda_5(2)$	$\mu_1(1)...\mu_5(1)$ $\mu_1(2)...\mu_5(2)$	$\nu_1(1)...\nu_5(1)$ $\nu_1(2)...\nu_5(2)$	$\tau_{12}^{(1)}$	$\tau_{21}^{(1)}$
Fig. 1	5	x axis 1e-20	4.1,4.3,4.5,4.7,4.9 1e-20	0.35,0.4,0.45,0.6,0.7 1e-20	0.05	0.1
Fig. 2	5	x axis x axis	4.1,4.3,4.5,4.7,4.9 1e-20	0.35,0.4,0.45,0.6,0.7 0.35,0.4,0.45,0.6,0.7	0.05	0.1
Fig. 3	5	x axis 1e-20	4.1,4.3,4.5,4.7,4.9 1e-20	0.35,0.4,0.45,0.6,0.7 1e-20	0.1	0.2
Fig. 3 (run 2)	5	x axis $\lambda_1(1)/2...\lambda_5(1)/2$	4.1,4.3,4.5,4.7,4.9 1e-20	0.35,0.4,0.45,0.6,0.7 1e-20	0.1	0.2
Fig. 4	5	x axis x axis	4.1,4.3,4.5,4.7,4.9 1e-20	0.35,0.4,0.45,0.6,0.7 0.35,0.4,0.45,0.6,0.7	0.1	0.2
Fig. 4 (run 2)	5	x axis $\lambda_1(1)/2...\lambda_5(1)/2$	4.1,4.3,4.5,4.7,4.9 1e-20	0.35,0.4,0.45,0.6,0.7 0.35,0.4,0.45,0.6,0.7	0.1	0.2

Table 3: System parameters

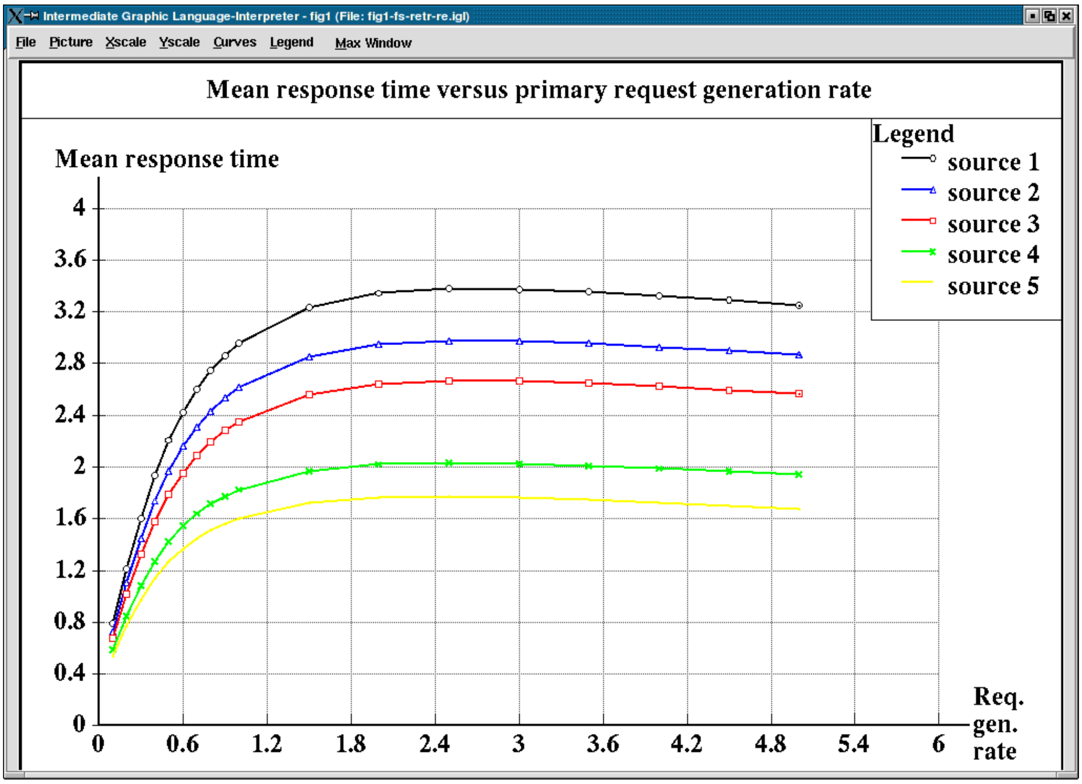

Figure 1: Mean response time versus primary request generation rate

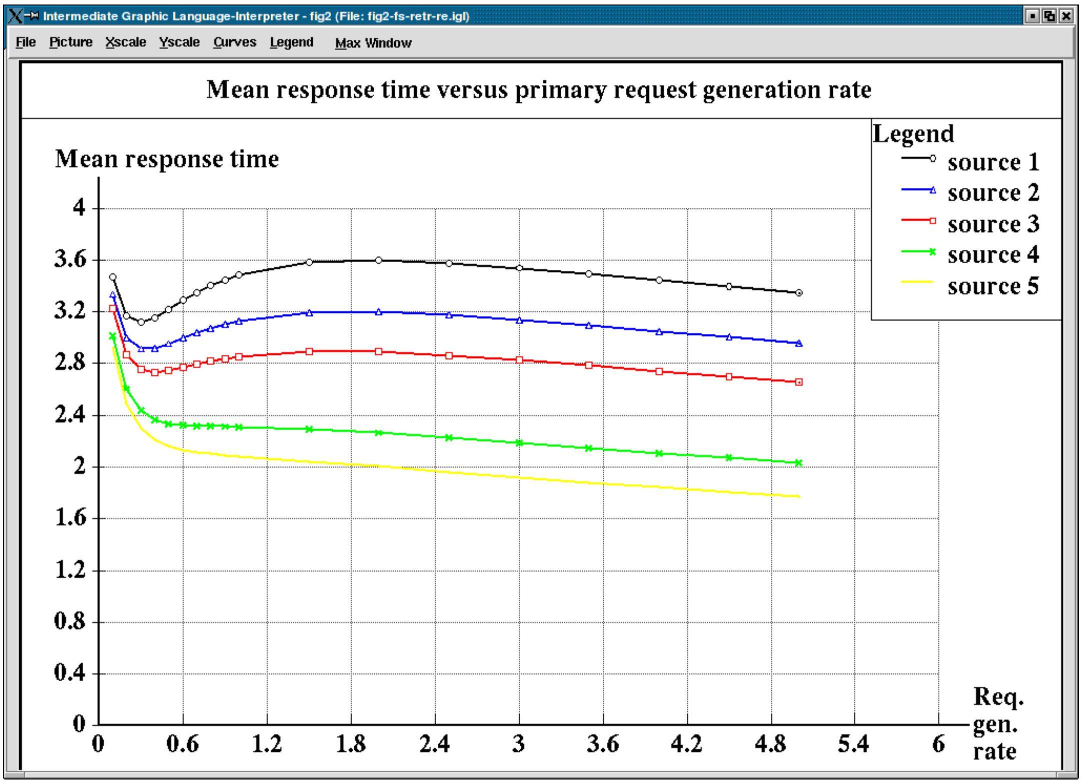

Figure 2: Mean response time versus primary request generation rate

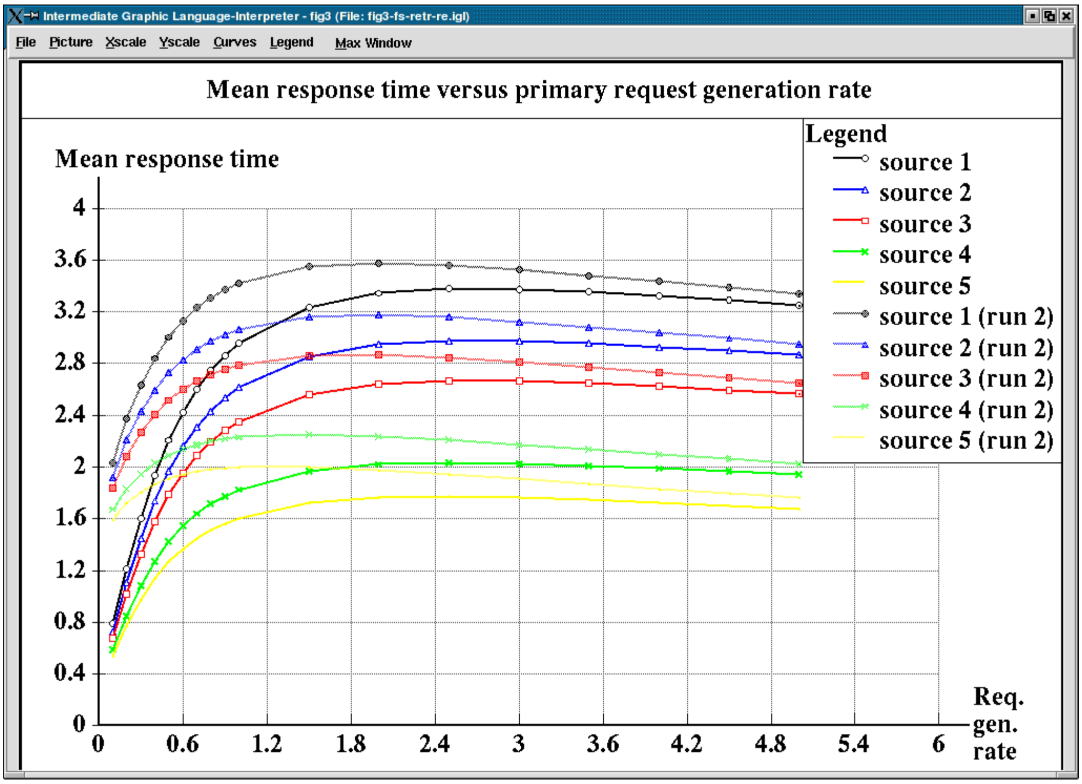

Figure 3: Mean response time versus primary request generation rate

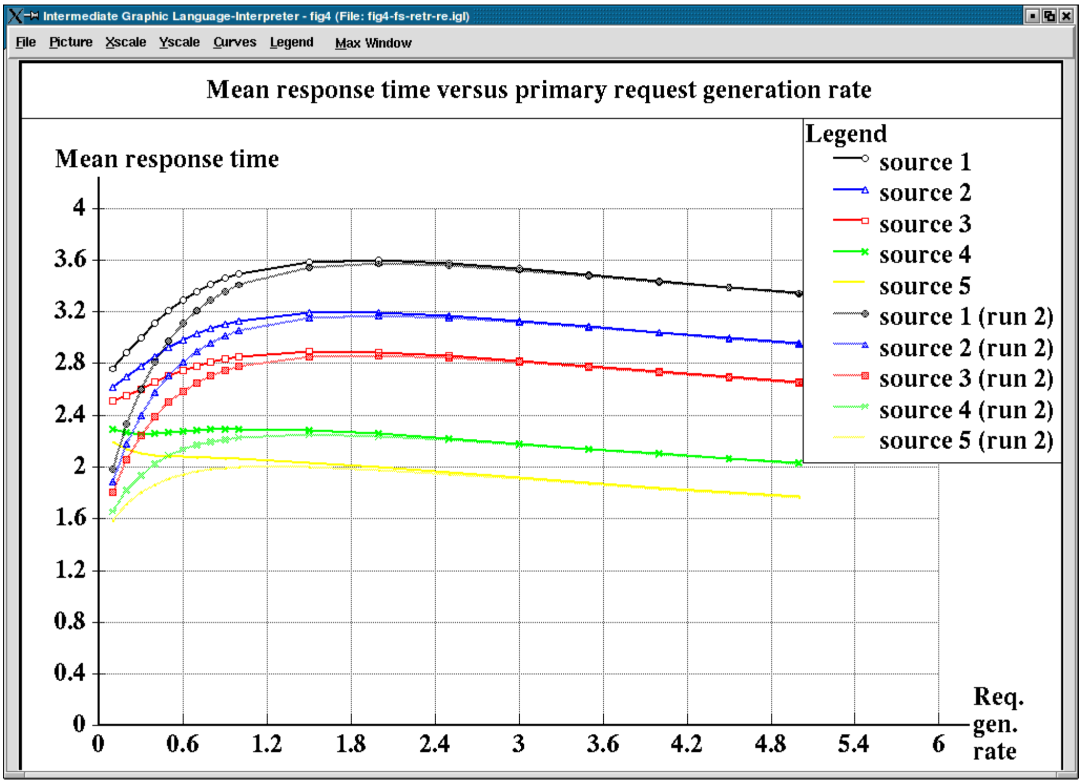

Figure 4: Mean response time versus primary request generation rate

4 Comments

For the easier understanding only simple cases are considered. We used only one random environment with 2 states. The tool is able to deal with systems with several environments.

5 Conclusions

In this paper, a finite-source retrial queue with heterogeneous sources operating in random environments was considered. The numerical results were obtained by the MOSEL tool, and several examples were illustrated graphically.

References

[1] **Almási B., Bolch G., Sztrik J.** (2001) Performability Modeling a Client-Server Communication System with Randomly Changing Parameter Using MOSEL. *5th International Workshop on Performability Modeling of Computer and Communication Systems*, Erlangen, Germany, 37-41.

[2] **Almási B., Bolch G., Sztrik J.** (2003) Analysing Markov-modulated finite-source queueing systems. *Annales Univ. Sci. Budapest., Sect. Comp.*, Vol. 22, 22–33.

[3] **Almási B., Roszik J., Sztrik J.** Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs. *Journal of Mathematical Sciences* (submitted for publication)

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

[5] **Falin G.I., Templeton J.G.C.** (1997) *Retrial queues*. Chapman and Hall, London.

[6] **Falin G.I., Artalejo J.R.** (1998) A finite source retrial queue. *European Journal of Operational Research*, Vol. 108, 409–424.

[7] **Gaver, D. P., Jacobs P. A., Latouche** (1984) Finite birth-and-death models in randomly changing environments. *Advances in Applied Probability*, Vol. 16, 715–731.

[8] **Onur E., Delic H., Ersoy C., Caglayan M.U.** (2002) Measurement-based replanning of cell capacities in GSM networks. *Computer Networks*, Vol. 39, 749–767.

[9] **Roszik J., Sztrik J., Kim C.S.** (2005) Retrial queues in the performance modeling of cellular mobile networks using MOSEL. *International Journal of Simulation: Systems, Science & Technology*, Vol. 6, No. 1-2, 38–46.

[10] **Skelly P., Schwartz M., Dixit S.** (1993) A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer. *IEEE Trans. on Networking*, Vol. 1, 446–459.

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