HETEROGENEOUS FINITE-SOURCE RETRIAL QUEUES

B. Almási, * G. Bolch, J. Sztrik
University of Debrecen, Debrecen, Hungary
* University of Erlangen, Erlangen, Germany
{almasi|jsztrik}@math.klte.hu
bolch@informatik.uni-erlangen.de

Abstract

In this paper we investigate a single server retrial queue with a finite number of heterogeneous sources of calls. It is assumed when a given source is idle it will generate a primary call after an exponentially distributed time. If the server is free at the time of the request's arrival then the call starts to be served. The service time is also exponentially distributed. During the service time the source cannot generate a new primary call. After service the source moves into free state and can generate a new call again. If the server is busy at the time of the arrival of a primary call, then the source starts generating so called repeated calls with exponentially distributed times until it finds the server free. As before, after service the source becomes free and can generate a new primary call again. We assume that the primary calls, repeated attempts and service times are mutually independent. This queueing system and its variants could be used to model magnetic disk memory systems, local area networks with CSMA/CD protocols and collision avoidance local area networks.
The novelty of this model is the heterogeneity of the calls, which means that each call is characterized by its own arrival, repeated and service rates. The aim of the paper is to give the usual steady-state performance measures of the system. To do so, an efficient software tool, MOSEL (Modeling, Specification and Evaluation Language) developed at the University of Erlangen, Germany, is used to formulate and solve the problem. Several sample numerical results illustrate the power of the tool showing the effect of different parameters on the system measures.

Keywords: retrial queueing systems, finite number of sources, heterogeneous calls, performance tool, performance measures

[1]Research is partially supported by NATO Scholarship for Senior Researchers 2002, German-Hungarian Bilateral Intergovernmental Scientific Cooperation, OMFB-DLR No. 21-2000

1 Introduction

Retrial queueing systems (or queueing systems with repeated attempts, returning customers, calls or orders) have wide practical use in designing local area networks and telecommunication systems. These queues are characterized by the following feature: a primary request finding all servers and waiting positions busy upon arrival leaves the service area, but after some random time he repeats his demand. Between retrials he is said to be in *'orbit '*. So the repeated attemps for service from the group of blocked customers are superimposed on the normal stream of arrivals of primary requests, thus the retrial queues can be considered as alternative to queues with losses that do not take repeated attemps into account.

In recent years, there has been considerable interest in retrial queues. For a systematic account of the fundamental methods and results, furthermore an accessible classified bibliography on this topic the interested reader is referred to, for example [2], [6], [9], [10], and references therein.

In many practical situations it is important to take into account the fact that the rate of generation of new primary calls 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. Queueing systems without retrials, that is systems with classical waiting lines and finite population have been review in detail by Takagi [27]. Since Kornyisev [20], which was the first paper devoted to finite-source retrial queues, there has been a rapid growth in the literature on this topic. A complete survey on related results can be found in Artalejo [2] for systems of type $M/G/1//K$ and $M/M/c//K$ in Kendall's notation. In addition, in the papers Falin, Artalejo [11], Falin [12] not only the outside observer's distributions of the systems in steady state, but also the stationary performance characteristics are considered on more detail. In particular, all main measures were expressed in terms of the server utilization. Arriving customer's distribution of the system state, busy period and waiting time processes (which is especially complex for retrial queues due to the overtaking) were investigated, too. Further recent results with finite-source of primary requests can be found in [4], [5], [8], [10], [13], [14], [19], [21], [24].

Retrial queues with quasi-random input are recent interest in modelling magnetic disk memory systems [23], cellular mobile networks [28], computer networks [15], and local-area networks with nonpersistent CSMA/CD protocols [21], with star topology [16, 22], with random access protocols [17], and with multiple-access protocols [18].

An examination of these papers shows no investigations on models with heterogenous sources. Thus the aim of the present paper is to analyse a finite-source retrial queues with the following assumptions. There are K different sources of primary calls each consisting a single request. When source i is free at time t (i.e. is not being served and not waiting for service) it may generate a primary call during interval $(t, t+dt)$ with probability $\lambda_i dt + o(dt)$. If the server is idle at time of its arrival then the service starts. The service is finished during the interval $(t, t+dt)$ with probability $\mu_i dt + o(dt)$. During the service time the source cannot generate a new primary request. After service the source moves into a free state and can generate a new call. If the server is busy at time of the arrival of request from the ith source, then the source starts generating a Poisson flow of repeated calls with rate ν_i until it finds the server free. As before, after service the source becomes free and can generate a new primary call. All the times involved in the model are assumed to be mutually independent of each other.

Our objective is to give the main usual stationary performance measures of the system and to show the effect of different parameters on them. To achive this goal a tool called MOSEL (Modeling, Specification and Evaluation Language) developed at the University of Erlangen, Germany, see [7], is used to formulate and solve the problem. We show how this system can be modelled, and how easily performance measures can graphically be represented using IGL (Intermediate Graphical Language). This model is another extention of investigations for heterogeneous finite-source queueing systems that in the case of the classical service disciplines (first-come-first-served, polling, processor-shared, priority processor-shared) were treated, for example in Sztrik [25, 26] and Takagi [27].

The paper is organised as follows. In Section 2 the full description of the model by the help of the correponding multi-component Markov chain is given. Then the main performance measures of the system 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 $\vec{M}/\vec{M}/1//K$ retrial queueing model

2.1 The underlying Markov chain

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 $X(t) = ((\alpha_{C(t)}; \beta_1, ..., \beta_{N(t)}), t \geq 0)$ where $C(t) = 0$ if the server is idle, $C(t) = 1$ if the server is busy, and $\alpha_C(t)$ is the index of the request under service at time t. $N(t)$ is the number of sources of repeated calls at time t, and because of the heterogeneity of the sources we need to identify their indices that are denoted by β_j, $j = 1, ..., N(t)$ if there is a customer in the orbit, otherwise the second component is 0. Since its state space is finite the process $(X(t), t \geq 0)$ is ergodic for all values of the rate of generation of new primary calls, and from now on we assume that the system is in the steady state.

We define the stationary probabilities

$$P(0; 0) = \lim_{t\to\infty} P(C(t) = 0, N(t) = 0)$$

$$P(j; 0) = \lim_{t\to\infty} P(\alpha_1 = j, N(t) = 0), \qquad j = 1, ..., K$$

$$P(0; i_1, ..., i_k) = \lim_{t\to\infty} P(C(t) = 0, \beta_1 = i_1, ..., \beta_k = i_k), \qquad k = 1, ..., K-1$$

$$P(j; i_1, ..., i_k) = \lim_{t\to\infty} P(\alpha_1 = j, \beta_1 = i_1, ..., \beta_k = i_k), \qquad k = 1, ..., K-1.$$

The traditional way is to derive the related Kolmogorov equations for these probabilities and using the norming condition somehow we have to solve the set of equations. In our case these two steps are performed by the help of the tool treated in the next subsection.

Once we have obtained these limiting probabilities the **main system performance measures** can be derived in the following way.

1. The server utilization with respect to source j

$$U_j = \text{P (the server is busy with source j)}$$

that is, we have to summarize all the probabilities where the firsr component is j. Formaly

$$U_j = \sum_{k=0}^{K-1} \sum_{i_1,...,i_k \neq j} P(j; i_1, ..., i_k)$$

Hence the *server utilization*

$$U == E[C(t) = 1] = \sum_{j=1}^{K} U_j.$$

Let us denote by $P_W^{(i)}$ the steady state probability that request i is waiting (staying in the orbit). It is easy to see that

$$P_W^{(i)} = \sum_{j=0, j\neq i}^{K} \sum_{k=1}^{K-1} \sum_{i\epsilon(i_1,...,i_k)} P(j; i_1, ..., i_k).$$

Similarly, it can easily be seen, that the steady state probability $P^{(i)}$ that request i is in the service facility (it is under service or waiting in the orbit) is given by

$$P^{(i)} = P_W^{(i)} + U_i.$$

2. Mean response time of source i

The derivation of the following formulae are based on [27]. Let us denote by $E[T_i]$ the mean response time of customer i, and by γ_i the *throughput* of request i, that is, the mean number of times that request i is served per unit time. These are related by

$$\gamma_i = \frac{1}{E[T_i] + 1/\lambda_i} = \lambda_i(1 - P^{(i)}) = \mu_i U_i, \qquad i = 1, ..., K. \tag{1}$$

For $P^{(i)}$ we have

$$P^{(i)} = \frac{E[T_i]}{E[T_i] + 1/\lambda_i} = \gamma_i E[T_i] = 1 - \frac{\gamma_i}{\lambda_i} \qquad i = 1, ..., K. \tag{2}$$

which represents **Little's theorem** for request i in the service facility. It is easy to see that as a concequence of (1) we have

$$P^{(i)} = 1 - \frac{\mu_i}{\lambda_i} U_i$$

and

$$P_W^{(i)} = P^{(i)} - U_i = 1 - \frac{\mu_i + \lambda_i}{\lambda_i} U_i.$$

Alternatively, by the help of (2) we can express the mean response time $E[T_i]$ for request i in terms of U_i as

$$E[T_i] = \frac{P^{(i)}}{\lambda_i(1 - P^{(i)})} = \frac{1 - \frac{\mu_i}{\lambda_i} U_i}{\mu_i U_i} = \frac{\lambda_i - \mu_i U_i}{\lambda_i \mu_i U_i}.$$

3. Mean waiting time of source i

The mean waiting time of request i is given by

$$E[W_i] = E[T_i] - 1/\mu_i = \frac{1}{\gamma_i} - \frac{1}{\lambda_i} - 1/\mu_i = \frac{\lambda_i - (\mu_i + \lambda_i)U_i}{\lambda_i \mu_i U_i}.$$

At the same time we have another **Little's theorem** for request i waiting for service. Namely

$$P_W^{(i)} = \frac{E[W_i]}{E[T_i] + 1/\lambda_i} = \gamma_i E[W_i] \qquad i = 1, ..., K.$$

4. Mean number of calls staying in the orbit or in service

$$M = E[C(t) + N(t)] = \sum_{i=1}^{K} P^{(i)} = \sum_{i=1}^{K}(1 - \frac{\mu_i}{\lambda_i}U_i) = K - \sum_{i=1}^{K} \frac{\mu_i}{\lambda_i}U_i.$$

5. Mean number of sources of repeated calls

$$N = E[N(t)] = \sum_{i=1}^{K} P_W^{(i)} = \sum_{i=1}^{K}(1 - \frac{\mu_i + \lambda_i}{\lambda_i}U_i) = K - \sum_{i=1}^{K} \frac{\mu_i + \lambda_i}{\lambda_i}U_i.$$

6. Mean rate of generation of primary calls

$$\overline{\lambda} = \sum_{i=1}^{K} \gamma_i = \sum_{i=1}^{K} \lambda_i(1 - P^{(i)}) = \sum_{i=1}^{K} \mu_i U_i.$$

7. Blocking probability of primary call i

$$B_i = \frac{\lambda_i \sum_{j=1, j\neq i}^{K} \sum_{k=0}^{K-1} \sum_{i\neq i_1,...,i_k} P(j; i_1, ..., i_k)}{\overline{\lambda}}.$$

Hence *blocking probability of primary calls*

$$B = \sum_{i=1}^{K} B_i$$

which is the fraction of primary calls which were blocked (i.e. met the server busy).

In particular, in the case of homogeneous calls, that is, when $\lambda_i = \lambda, \quad \mu_i = \mu, \quad \nu_i = \nu, \quad i = 1, ..., K$ the corresponding main performance measures treated in [11, 10] are the following

$$U_i = E[C(t)]/K, \quad i = 1, ..., K, \quad N = K - \frac{(\lambda + \mu)U}{\lambda},$$

$$\overline{\lambda} = \lambda E[K - C(t) - N(t)] = \mu U,$$

$$E[W] = N/\overline{\lambda} = K - (\mu U)^{-1} - \lambda^{-1} - \mu^{-1},$$

$$B = \frac{\lambda E[K - C(t) - N(t); C(t) = 1]}{\lambda E[K - C(t) - N(t)]}$$

as it can be seen in [11] with the appropriate changing of notations.

It should also be mentioned that all the performance measures can be expressed in the terms of the corresponding utilizations U_i, as it was stated in [11]. However, we must admit the distribution function and moments of the waiting times could not be solved.

2.2 MOSEL program

The crutial part of the whole modeling is the formulation of problem and derivation of the main performance measures. This can be done quite easily by the help of a tool called MOSEL (Modeling, Specification and Evaluation Language) developed at the University of Erlangen, Germany, see [7]. It automatically generates the state probabilities and using these a result file containing the performance measures specified in the model description file. It also generates a graphical representation of the results if the input file contains a picture part.

Since in our paper we concentrate on the effect of different parameters of the system on the main performance measures the technical details of programming are not treated. For just an illustration we show a part of the program file. The main emphasis is on the graphical representations of the desired measures which are automatically generated by the tool.

```
retrial2.msl begins */

/* Reliable terminal-system with retrying jobs for NT  terminals */

/*------------------------------- Definitions -----------*/
#define  NT   3     // change it to 4, 5, 6, 7, ...

/*============== No changes required below =======================*/

<1..NT> VAR double prgen#  ;
<1..NT> VAR double prretr# ;
<1..NT> VAR double prrun#  ;
enum cpu_states {cpu_busy, cpu_idle};
enum terminal_states {term_busy, term_retrying, term_waiting};
```

```
/*---------------------------------------- Node definitions -------*/
<1..NT>  NODE terminal#[terminal_states] = term_busy;
         NODE cpu[cpu_states]            = cpu_idle;


/*---------------------------------------------- Transitions -------*/
<1..NT>     FROM cpu_idle, terminal#[term_busy] TO cpu_busy,
 terminal#[term_waiting]
                                            W prgen# ;
<1..NT>     FROM terminal#[term_busy] TO terminal#[term_retrying]

                                            IF cpu==cpu_busy W prgen# ;

<1..NT>     FROM cpu_idle, terminal#[term_retrying] TO cpu_busy,
 terminal#[term_waiting]
                                            W prretr#;
<1..NT      FROM cpu_busy, terminal#[term_waiting] TO cpu_idle,
 terminal#[term_busy]
                                            W prrun# ;


/*---------------------------------------------- Results -------*/
        RESULT>> if(cpu==cpu_busy) cpuutil += PROB                 ;
<1..NT> RESULT>> if(terminal#==term_busy) termutil# += PROB        ;
<1..NT> RESULT>> if(terminal#==term_retrying) retravg += PROB      ;
<1..NT> RESULT>> resptime# = (1-termutil#) / (prgen# * termutil#   ;

/* retrial2.msl ended
```

3 Numerical examples

In this section several sample numerical results illustrate the power of the tool showing the influence of different parameters on the mean response time $E[T_i]$. In homogeneous case the results were validated by the Pascal program given in [10] in pages 272-274. For the easier representation of graphics we deals with 3 sources, but in the different setups the request generation, retrial and service rates play different roles. All the time we display the mean response time $E[T_i]$ of call i as the function of the above mentioned rates. First we consider totally homogenous systems then systems with mixed group of rates are investigated. Finally rather complex situation is analyzed showing the unpredictable operational behaviour of the finite-source retrial system.

3.1 Comments

In Figure 1 we can see how the mean reponse time $E[T]$ for the retrial system approaches the mean time for the classical system without retrial as the retrial rate increases. This is not suprising since in this case the retrial system becomes a FCFS finite-source queue. In Figures 2-3 the expected effects of arrival and service rate are demonstrated. In Figure 4 we present 3 curves with the same different retrial rates as it was considered in [11] showing a suprising

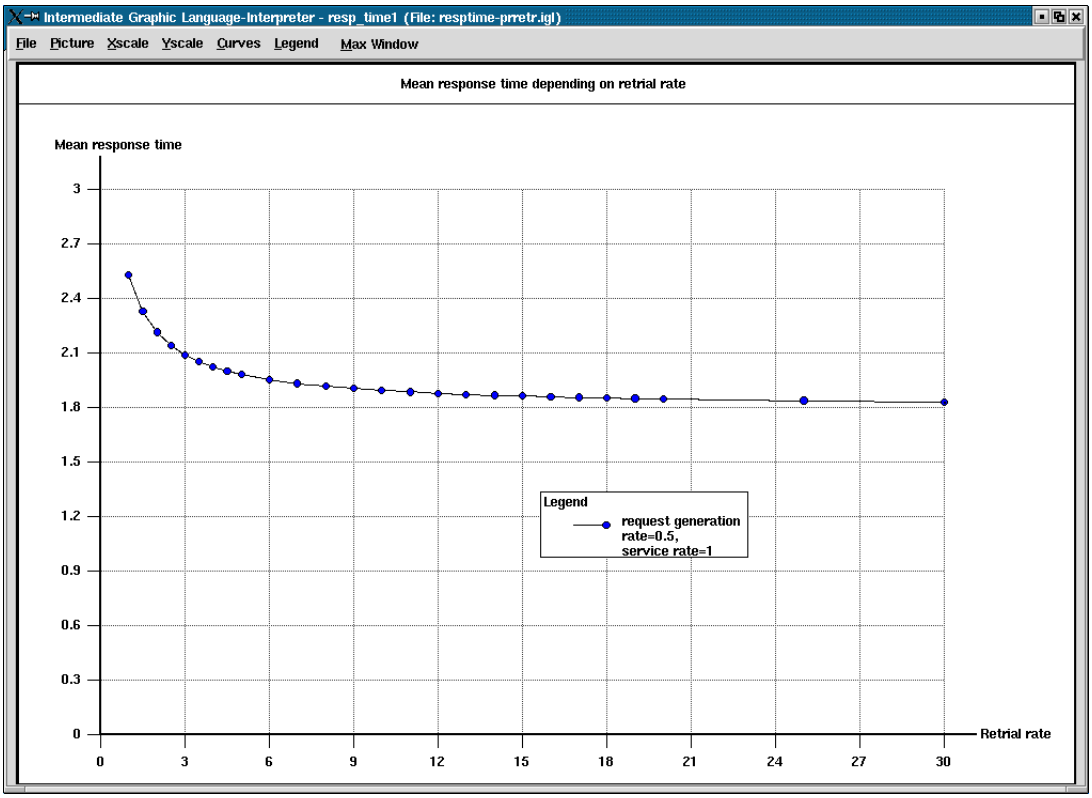

Figure 1: $E[T]$ versus retrial rate

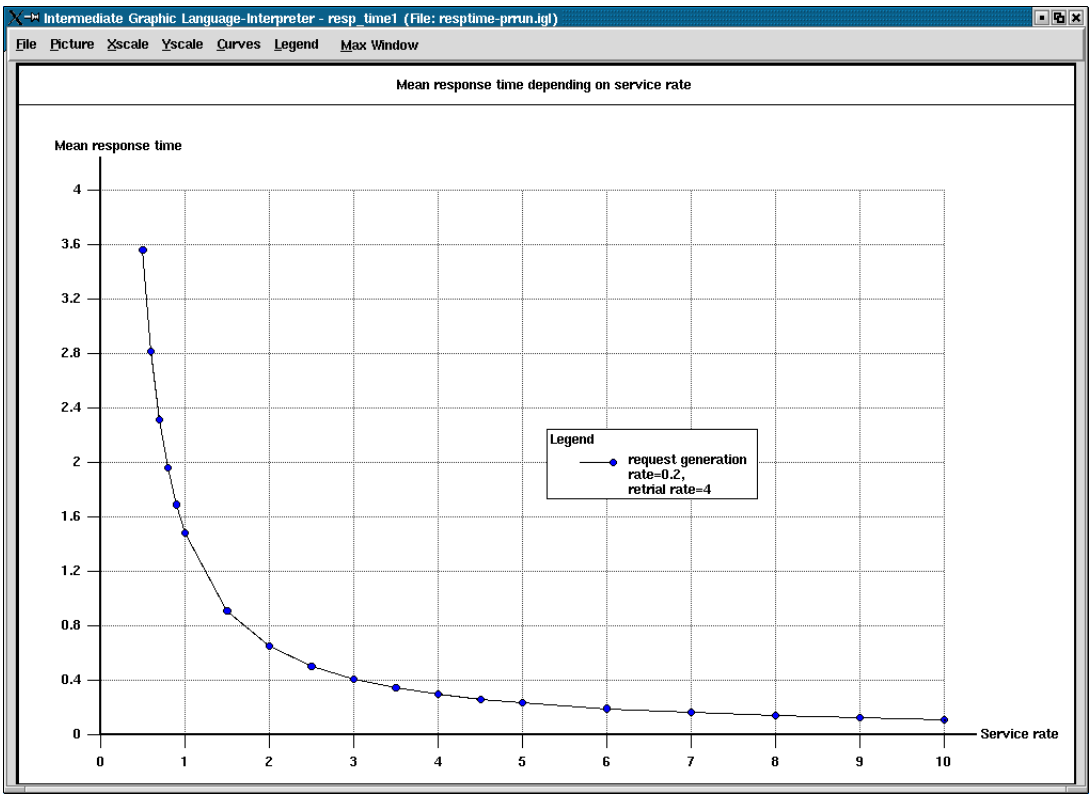

Figure 2: $E[T]$ versus service rate

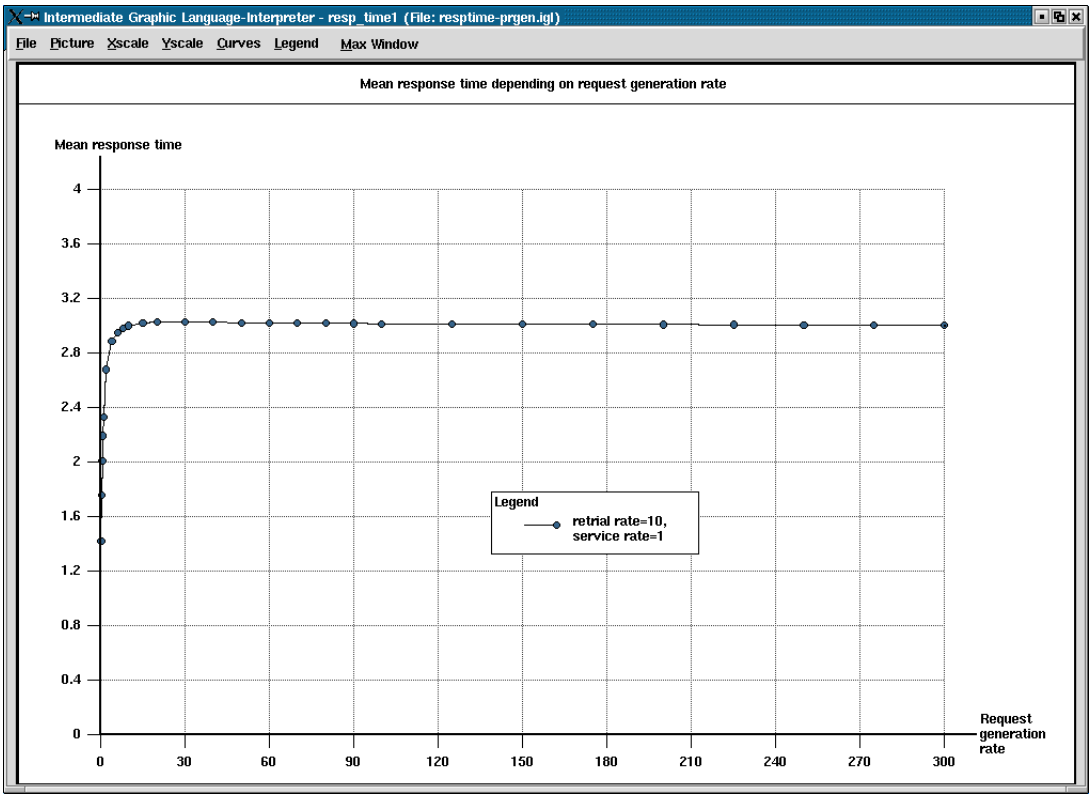

Figure 3: $E[T]$ versus primary request generation rate

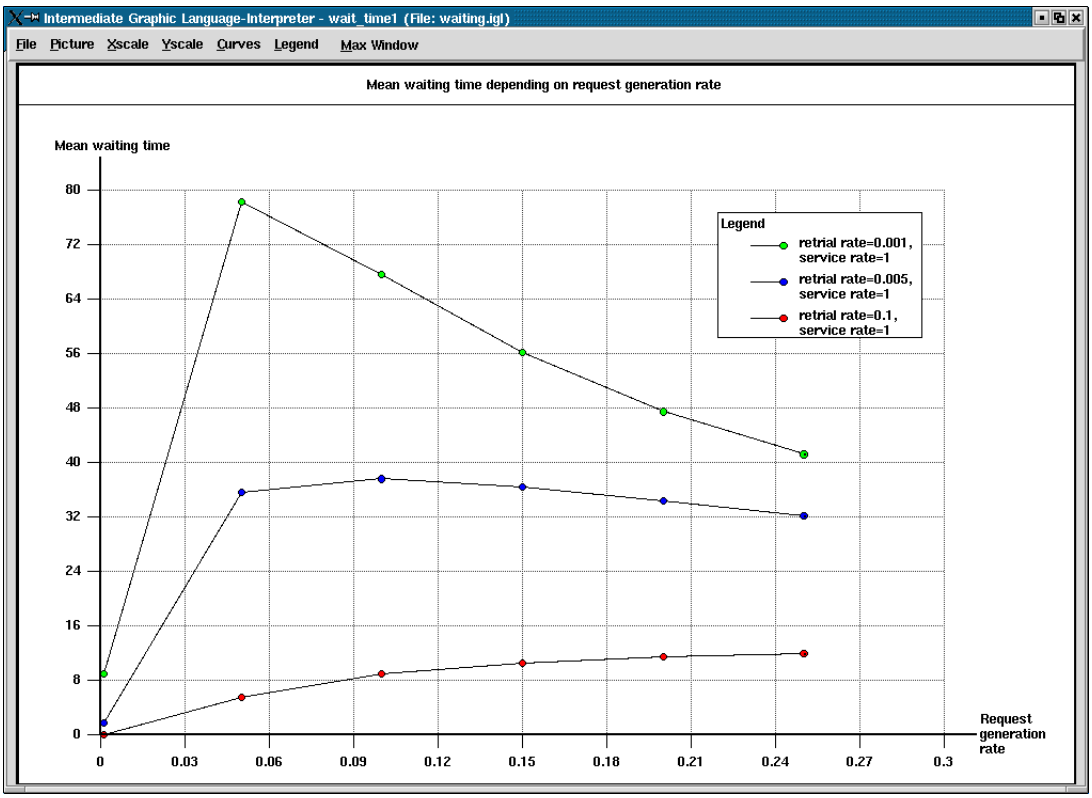

Figure 4: $E[T]$ versus primary request generation rate

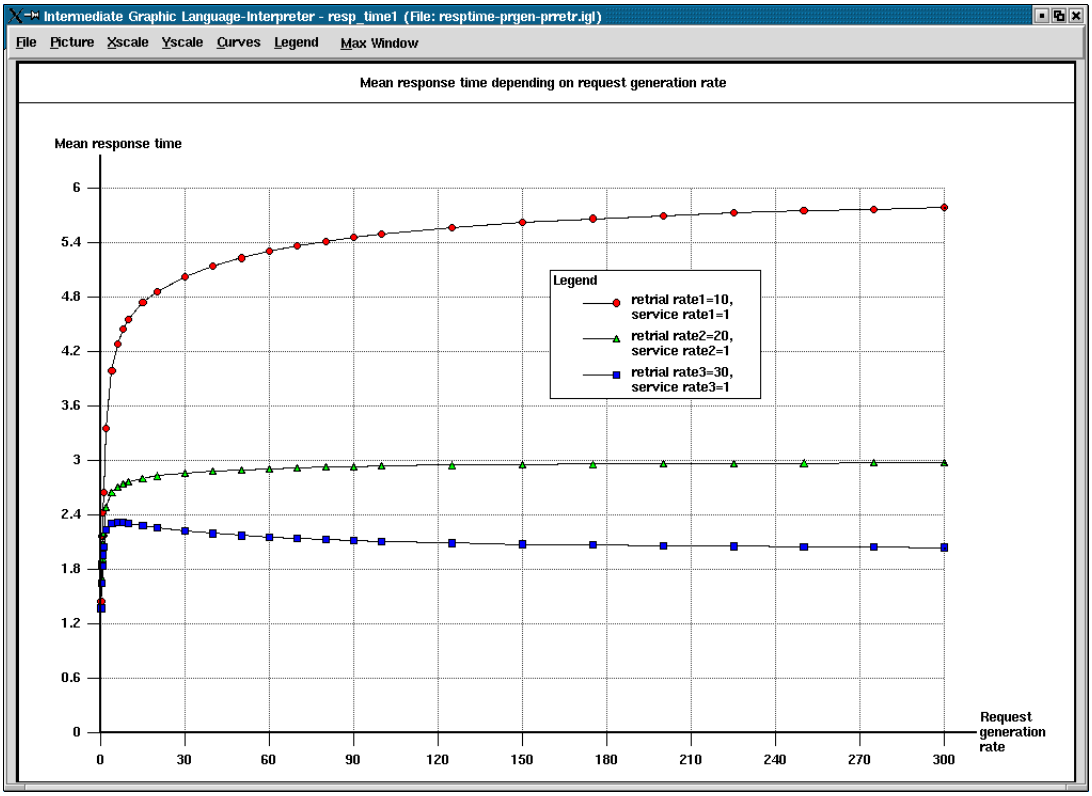

Figure 5: $E[T]$ versus primary request generation rate with homogeneous service and heterogeneous retrial

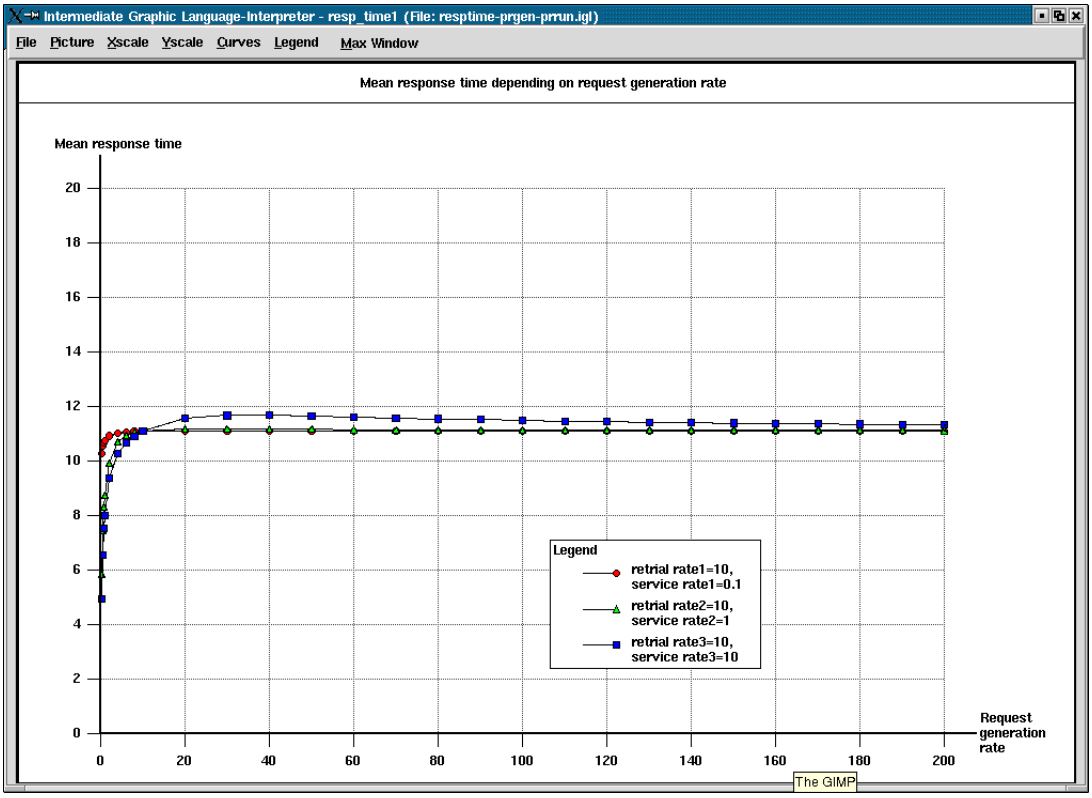

Figure 6: $E[T]$ versus primary request generation rate with homogeneous retrial and heterogeneous service

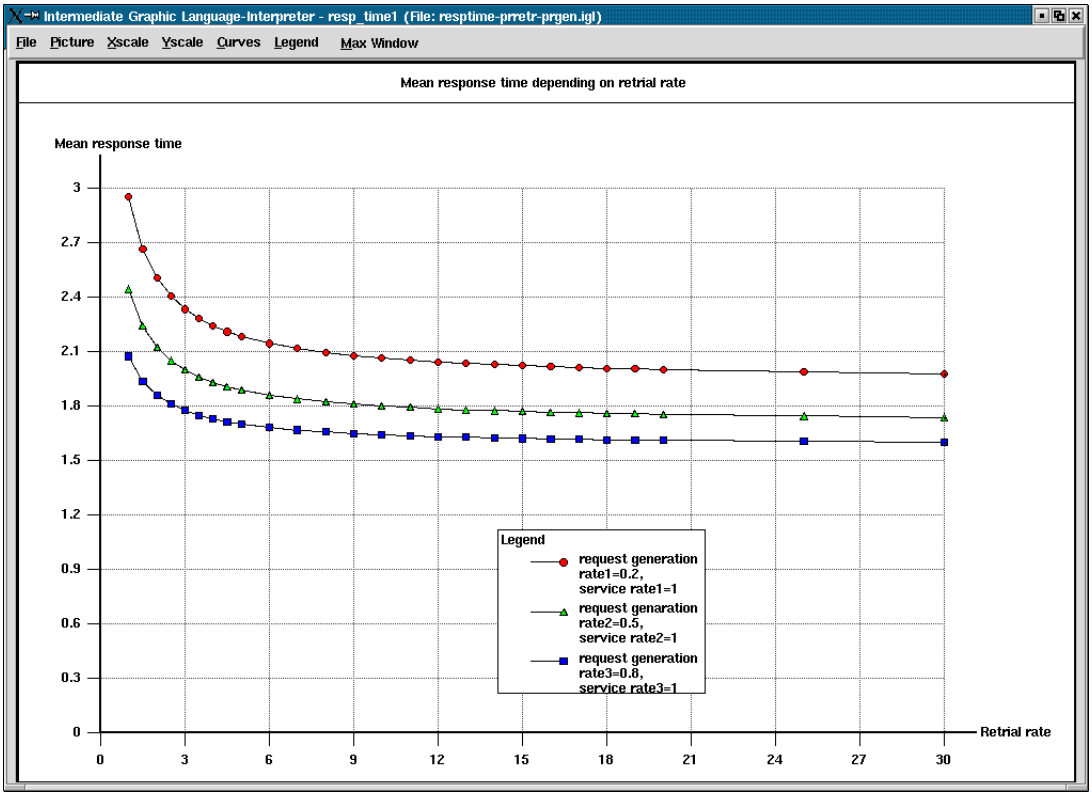

Figure 7: $E[T]$ versus retrial rate with homogeneous service and heterogeneous primary request generation

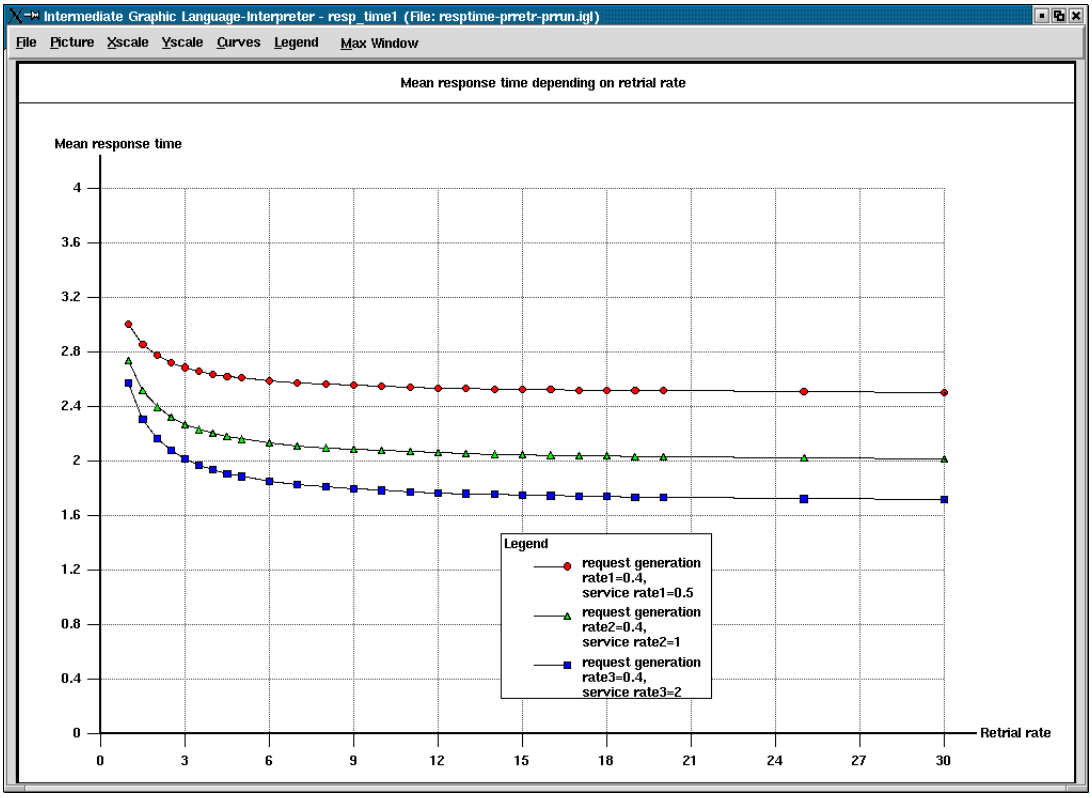

Figure 8: $E[T]$ versus retrial rate with homogeneous primary request generation and heterogeneous service

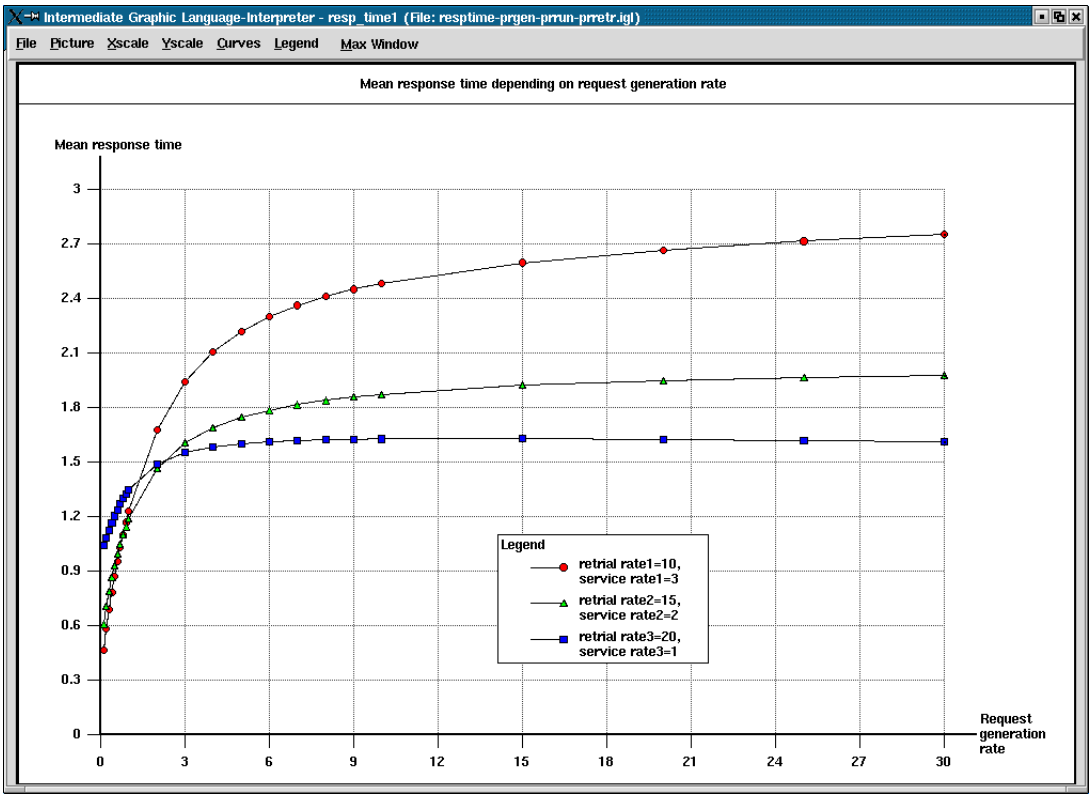

Figure 9: $E[T]$ versus primary request generation rate with heterogeneous service and heterogeneous retrial

phenomenon of retrial queues having a maximum of $E[T]$. In Figures 5-8 different systems with mixed group of rates are investigated. Finally in Figure 9 a rather complex situation is considered as the function of primary request generation rate with fixed heterogeneous service and heterogeneous retrials.

4 Conclusions

In this paper a finite-source retrial queueing system with heterogeneous requests is studied. The novelty of the investigation is the heterogenity of calls which makes the system rather complicated. A tool MOSEL was used to formulate and solve the problem and the main performance measures were derived. Several numerical calculations were performed to show the effect of different parameters on the mean reponse times of the calls.

Acknowledgements

We thank János Roszik for his programming work with MOSEL.

References

[1] **Almási B., Bolch G., Sztrik J.** Performability Modeling of Non-homogeneous Terminal Systems Using MOSEL, *5th International Workshop on Performnability Modeling of Computer and Communication Systems, Erlangen, Germany, 2001* 37-41.

[2] **Artalejo J.R.** Retrial queues with a finite number of sources, *J. Korean Math. Soc.* 35(1998) 503-525.

[3] **Artalejo J.R.** Accessible bibliography on retrial queues, *Math. Comput. Modeling* 30(1999) 1-6.

[4] **Artalejo, J.R. and Gomez-Corral, A.** Information theoretic analysis for queueing systems with quasi-random input, *Math. Comput. Modeling* 22(1995) 65-76.

[5] **Artalejo, J.R., Rajagopalan V., Sivasamy R.** On finite Markovian queues with repeated attemps, *Investigacion Operativa* 9(2000) 83-94.

[6] **Barcelo J., Escudero L., Artalejo J., editors** *Proceedings of the 1st International workshop on retrial queues (WRQ'98),* Madrid, Top 7(1998).

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

[8] **Dragieva V.I.** Single-line queue with finite source and repeated calls, *Problems of Information Transmission* 30(1994) 283-289.

[9] **Falin G.I.** A survey of retrial queues, *Queueing Systems* 7(1990) 127-168.

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

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

[12] **Falin G.I.** A multiserver retrial queue with a finite number of sources of primary calls, *Mathematical and Computer Modelling* 30(1999) 33-49.

[13] **Falin G.I. and Gomez Corral A.** On a bivariate Markov process arising in the theory of single-server retrial queues, *Statistica Neerlandica* 54(2000) 67-78.

[14] **Gomez Corral A.** Analysis of a single-server retrial queue with quasi-random input and nonpreemptive priority, *Computers and Mathematics with Applications* 43(2002) 767-782.

[15] **Houck D.J., Lai W.S.** Traffic modelling and analysis of hybrid fibercoax systems, *Computer Networks and ISDN Systems* 30(1998) 821-834.

[16] **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* 45(1997) 360-364.

[17] **Kalmychkov A.I. and Medvedev G.A.** Probability characteristics of Markov local-area networks with random-access protocols, *Automatic Control and Computer Science* 24(1990) 38-45.

[18] **Khomichkov I.I.** Study of models of local networks with multiple-access protocols, *Automation and Remote Control* 54(1993) 1801-1811.

[19] **Kok A.G.** Algorithmic methods for single server systems with repeated attemps, *Statistica Neerlandica* 38(1984) 23-32.

[20] **Kornyshev Y.N.** Design of a fully accessible switching system with repeated calls, *Telecommunications* 23(1969) 46-52.

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

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

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

[24] **Stepanov S.N.** The analysis of the model with finite number of sources and taking into account the subscriber behaviour, *Automation and Remote Control* 55(1994) 100-113.

[25] **Sztrik J.** Probability model for non-homogeneous multiprogramming computer systems, *Acta Cybernetica* 6(1983) 93-101.

[26] **Sztrik J.** A probability model for priority processor-shared multiprogrammed computer systems, *Acta Cybernetica* 7(1986) 329-340.

[27] **Takagi H.** *Queueing Analysis, A Foundation of Performance Evaluation, Vol. 2., Finite Systems*, North-Holland, Amsterdam, 1993.

[28] **Tran-Gia P. and Mangjes M.** Modeling of customer retrial phenomenon in cellural mobile networks, *IEEE Journal of Selected Areas in Communications* 15(1997) 1406-1414.