



Retrial Queues and their Application in Performance Modelling of Communication Networks

**Visszatéréses sorbanállási rendszerek és alkalmazásuk kommunikációs
hálózatok modellezésében**

Doktori (PhD) értekezés téziszfüzete

ROSZIK JÁNOS

Témavezető:

Prof. Dr. Sztrik János

Debreceni Egyetem, Informatikai Kar

Debrecen, 2006

Tartalomjegyzék

1	Introduction	3
1.1	Introduction to Retrial Queues	3
2	Research Objectives	3
3	Methodology	3
4	New Results	4
4.1	Analysis of Retrial Queueing Systems	4
4.1.1	Analysis of Single-server Non-reliable Finite-source Retrial Queues	5
4.1.2	Analysis of Multiserver Non-reliable Finite-source Retrial Queues	7
4.1.3	Analysis of Retrial Queues in Random Environment	10
4.2	Application of Retrial Queues in Performance Modelling of the GSM System	12
5	Application of the Results	14
6.	Bevezetés	17
6.1.	Visszatéréssel sorbanállási rendszerek	17
7.	Kutatási célok	17
8.	Az alkalmazott módszerek	17
9.	Új eredmények	18
9.1.	Visszatéréssel sorbanállási rendszerek elemzése	18
9.1.1.	Egykiszolgálós, nem megbízható, véges forrású visszatéréssel rendszerek elemzése	19
9.1.2.	Többkiszolgálós, nem megbízható, véges forrású visszatéréssel rendszerek elemzése	21
9.1.3.	Véletlen környezetben lévő visszatéréssel rendszerek elemzése	24
9.2.	Visszatéréssel sorbanállási rendszerek alkalmazása a GSM rendszer modellezésében	26
10.	Az eredmények alkalmazása	28

1 Introduction

Performance evaluation plays an important role in the design, analysis and development of practical systems, like computer and telecommunication systems and networks. Queueing models are often used for the performance and reliability modelling of these systems, and retrial queues are more and more frequently applied to certain types of them. For some examples, see [14; 16; 20]. 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 often has a non-neglectable negative effect on the performance measures.

Another important characteristic of real-life systems is non-reliability, which also has a negative influence on these measures, because most of the components of the systems are subject to random breakdowns and require repairs (see for example [15; 17; 21]). Non-reliability has been extensively studied for traditional queues with waiting lines, but only for infinite-source queues with returning customers.

1.1 Introduction to Retrial Queues

Retrial queues (queueing systems with repeated attempts, or queues with returning customers) are characterized by the following feature: a request finding all servers busy upon arrival leaves the service area but after some (random) time repeats his demand. In case of many real-life systems, retrial queues can be applied in the performance modelling, for example in modelling local-area and cellular mobile networks. For more detailed information and results on this type of queueing systems, see for example [4; 9; 11], and a complete survey can be found on queueing systems without retrials in [19]. For related literature about infinite-source non-reliable retrial queues the reader is referred to the works [2; 3; 12; 22].

In the dissertation, two types of retrial queues are introduced. The first one is a finite-source retrial queue, on which the analysis is based in the first part. The second one is a truncated infinite source queue with returning customers. An extension of this model is applied in the second part to analyze the GSM system (Global System for Mobile Communications).

2 Research Objectives

In the first part of the dissertation, the objective is to analyze some non-reliable finite-source retrial models and a reliable retrial queue in random environment (which also can be applied in the performance analysis of non-reliable systems). To the best know of the author, these models have not been treated in the literature before.

The second goal is to use a retrial queue in modelling of a practical system. In the second part, a real-life network is modelled using a retrial queueing model. A modelling way of the GSM system is treated with a modelling environment. This is based on previous works of various authors (for example [1; 16; 20]) and generalized with some model extensions.

3 Methodology

Because of the fact that the state space of the underlying Markov chains of the investigated queueing models is very large and the functioning of the systems is complex, it is quite difficult to calculate the steady state probabilities in the traditional way of solving the system of

steady-state equations. To simplify these calculations and to make these studies more usable in practice, the tool MOSEL (Modeling, Specification and Evaluation Language) [5] is used to formulate the models and to calculate the performance measures. With the tool, we can perform two steps in one, so we do not need to write down and somehow solve the set of steady state equations. The difficulty of modelling lies in the description of the system and its behavior for the performance tool.

MOSEL is a modelling environment with a high-level modelling language which allows us to describe complex real-world systems and to calculate their system measures using other performance evaluation tools. The model description can be translated automatically into the language of various performance tools and then analyzed by the appropriate tools (at present SPNP – Stochastic Petri Net Package and TimeNET are supported and suitable for the investigated models) to get these measures.

This tool has already been used, and it has proved its applicability for the modelling of several computer and communication systems. The functioning and usage of MOSEL is illustrated by Figure 1. The technical details of programming can be found in [5], and in [6; 7], where the new, revised version of MOSEL, called MOSEL-2, is introduced.

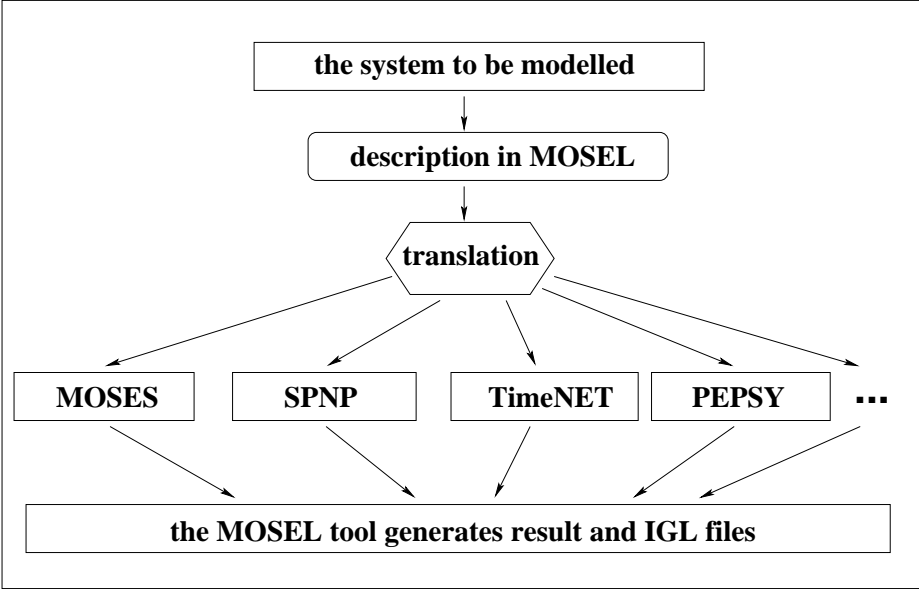


Figure 1: The modelling process in the MOSEL environment

In the first part of the dissertation, the original tool is used for the performance evaluation of finite-source retrial queueing systems. In the second part, the GSM system is modelled with the revised modelling language, and detailed comments are provided about MOSEL-2 programming.

4 New Results

4.1 Analysis of Retrial Queueing Systems

The components of the real systems may be subject to random breakdowns, so it is important to investigate non-reliable queues because of limited ability of repairs and heavy influence of

the breakdowns on the performance measures of the systems. Besides this, the analysis of non-reliable retrial queues (where the sources and the server may be subject to random breakdowns and repairs) is also important.

4.1.1 Analysis of Single-server Non-reliable Finite-source Retrial Queues

Consider a single server queueing system, where the primary calls are generated by K , $1 < K < \infty$ homogeneous sources. The server can be in operational (up) or non-operational (down) states, and it can be idle or busy. If the server is idle and up, 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 and up 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, 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 *returns to the orbit* (i.e. the source starts generation of repeated calls). 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 (new and repeated calls can be generated). All the times involved in the model are assumed to be independent of each other.

This model is another extension of investigations for homogeneous finite-source queueing systems without retrials but with server's breakdowns which were treated in [18]. Similarly, it generalizes the results of the reliable model of [11]. As it can be seen, this system is more complicated than in the reliable case, since it involves two types of failures, continued or repeated service and blocked or unblocked operations during breakdowns.

On the one hand, this model was extended with non-reliable sources. The sources can be non-operational only in free state. If a source is free at time t it can fail during the interval $(t, t + dt)$ with probability $\eta dt + o(dt)$ and then it moves to the repairman who follows FIFO discipline for the source breakdowns and gives preemptive priority to the server failure. The repair time of the sources is exponentially distributed with a finite mean $1/\kappa$.

On the other hand, the model was extended with (reliable) heterogeneous sources. In this case, the i -th source can generate a primary call during interval $(t, t + dt)$ with probability $\lambda_i dt + o(dt)$. If the server is idle and up 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 $\mu_i dt + o(dt)$ if the server is available. If the server is busy, then the source starts generating a Poisson flow of repeated calls with rate ν_i until it finds the server idle.

Thesis 1: *I have analyzed single-server non-reliable finite-source retrial queueing systems. The description of the models and the main stationary performance and reliability measures were given by the help of the corresponding Markov chains. The results were validated by former works, and the effect of changing various parameters on some of these measures were illustrated graphically. These models were published in [J1; J3; J4].*

In the following, we treat some example analysis.

In Figure 2, we can see the mean response time $E[T]$ for the reliable and the non-reliable server systems with continuous, non-continuous service after repair, with blocked and unblocked (intelligent) operations during service failure when the primary request generation rate increases. This figure demonstrates a surprising phenomenon of finite-source retrial queues having a maximum of $E[T]$ which was noticed in [10], too. The difference between continuous, non-continuous service, moreover blocked, unblocked systems' operations is clearly shown.

In Figure 3, the effect of the retrial rate on the mean number of requests staying in the orbit or in service is pictured. We can see the difference between the reliable, the non-reliable server and the non-reliable server and sources cases. If the server fails in busy state, it continues servicing the interrupted call after it has been repaired, and the operation of the sources is stopped or not stopped. It is worth pointing out that the values for the reliable case and the blocked case with reliable sources coincide. However, it is not so surprising since during the failure the number of requests remain the same.

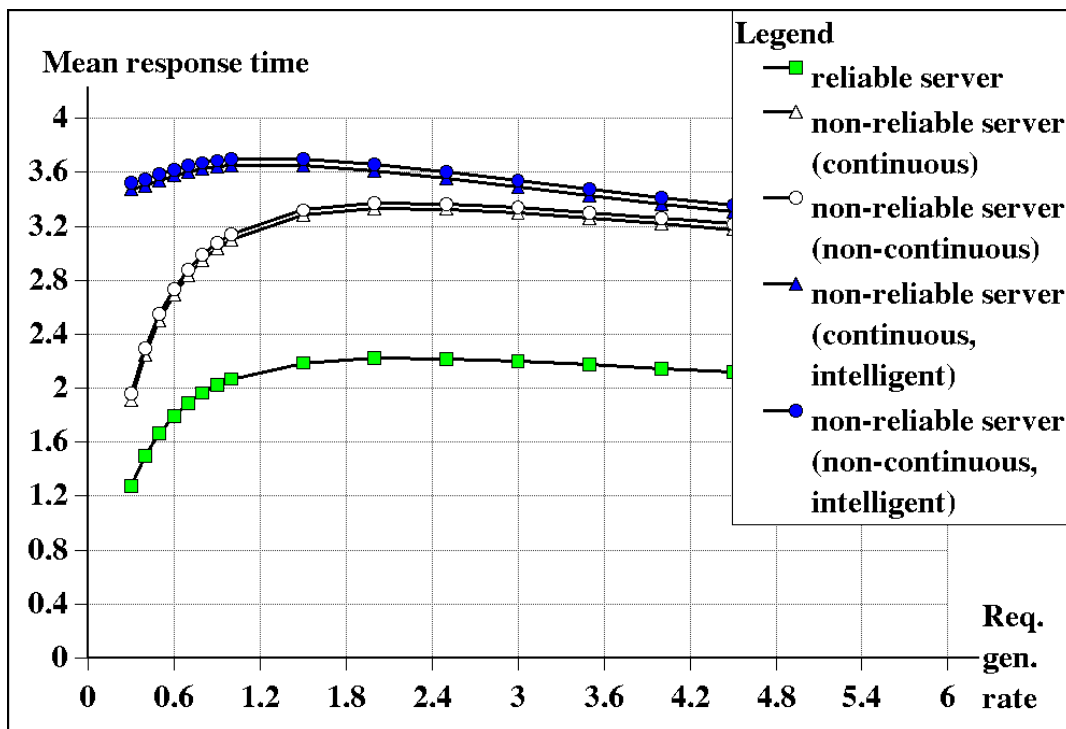


Figure 2: Mean response time versus primary request generation rate

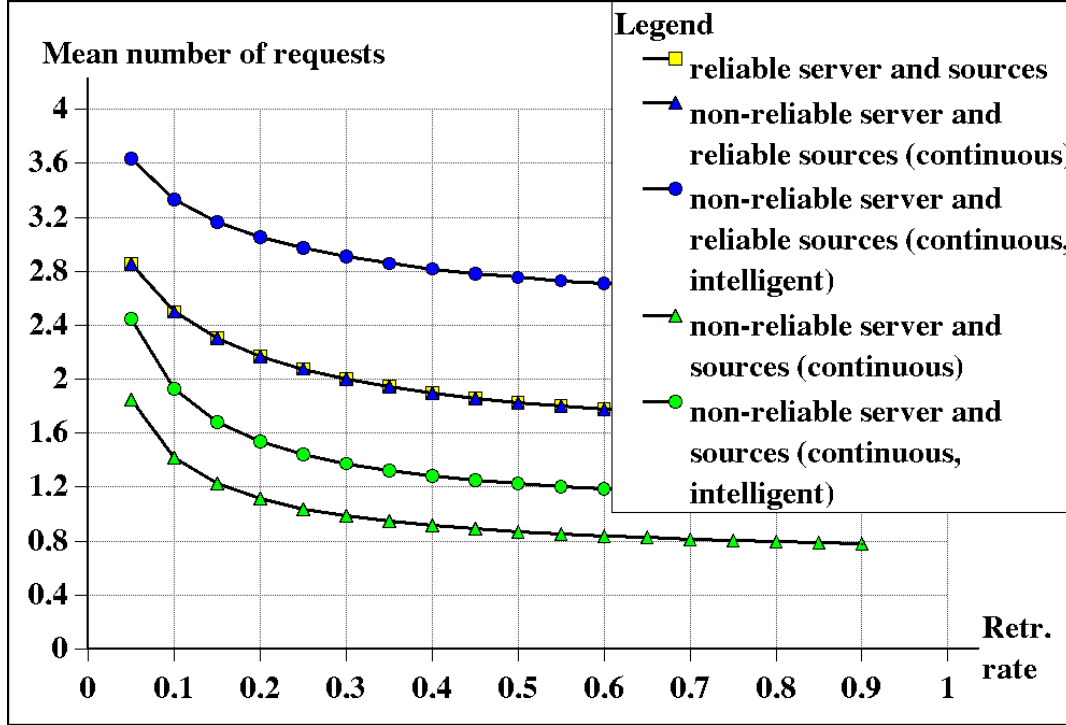


Figure 3: Mean number of requests staying in the orbit or in service versus retrial rate

4.1.2 Analysis of Multiserver Non-reliable Finite-source Retrial Queues

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 or non-operational states, and it 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 the arrival of the call then the service of the call starts. The service is finished during the interval $(t, t + dt)$ with probability $\mu_i dt + o(dt)$ if server i 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 the server fails in busy state, it either continues servicing the interrupted call after it has been repaired or the interrupted request returns to the orbit. We only investigate the case when the source moves into the sending repeated calls state at the moment of server's failure. The repairman follows FIFO discipline for the servers' breakdowns, and the repair time of server i is exponentially distributed with a finite mean $1/\tau_i$. If all the servers fail we treat two different cases. Namely, *blocked sources* case when all the operations are stopped expect from the repair of the first failed server. In the *unblocked (intelligent) sources* case only service is interrupted but all the other operations are continued.

If all the servers are busy (or failed in the unblocked case) at the moment of the arrival of a call the source starts generation of a Poisson flow of repeated calls with 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 the times involved in the model are assumed to be independent of each other.

Thesis 2: *I have analyzed a multiserver non-reliable finite-source retrial queueing system. The system was treated with non-reliable heterogeneous (asymmetric) servers, that is the servers had different parameters in service, failure and repair rates. The description of the model and the main stationary performance and reliability measures were given by the help of the corresponding Markov chain. The results were validated by former works, and the effect of changing various parameters on some of these measures were illustrated graphically. Furthermore, different service policies were compared in this model. These results will be published in [J5; J6].*

In Figure 4, we can see the difference between the Fastest Free Server (referred to as ordered in the legend of the figure), Random and the Averaged Random (where the service rate of the servers is the average of the rates of the heterogeneous case) service policies in the mean response time depending on the primary request generation rate. In the case of Random service disciplines, the requests are assigned to the idle servers randomly. In the other case, the requests are assigned to the fastest available free server. We can see that the difference between the two random cases are not too significant, but the Fastest Free Server case always has better response time, especially when more and more requests arrive.

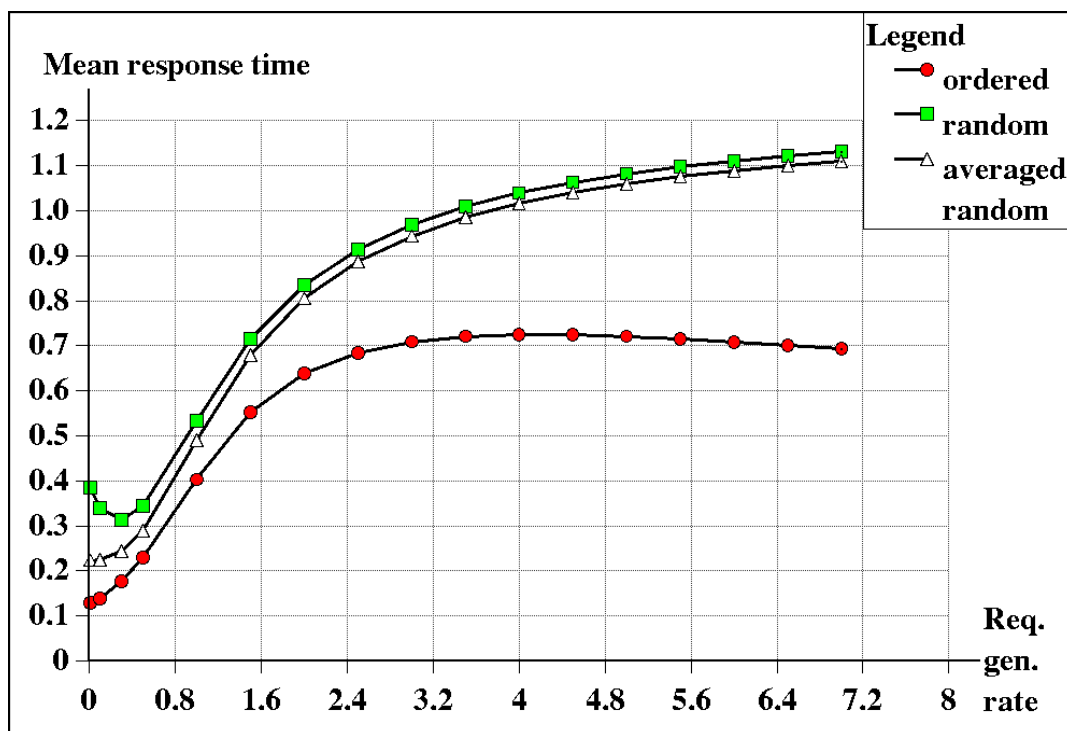


Figure 4: Mean response time versus primary request generation rate

In Figure 5 it is shown how the increase of the servers’ failure rate affects the mean response time. The averaged random case has a little better response time than the not averaged random case like in the former figure. The surprising decrease in the mean response time of the Fastest Free Server case can be explained by the help of Figure 6.

In Figure 6, we can see the server utilizations versus the servers’ failure rate with the same parameter setup as in Figure 5. In the random case, the slowest server has the highest utilization and the fastest has the lowest, since it services the request much faster and the requests are

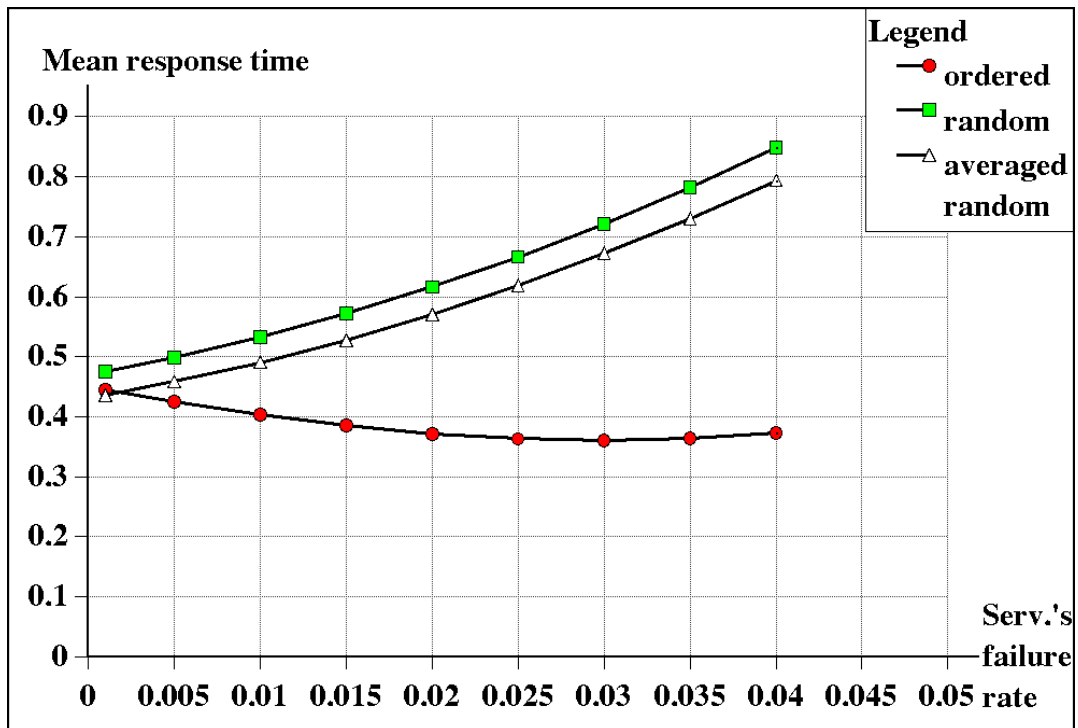


Figure 5: Mean response time versus servers' failure rate

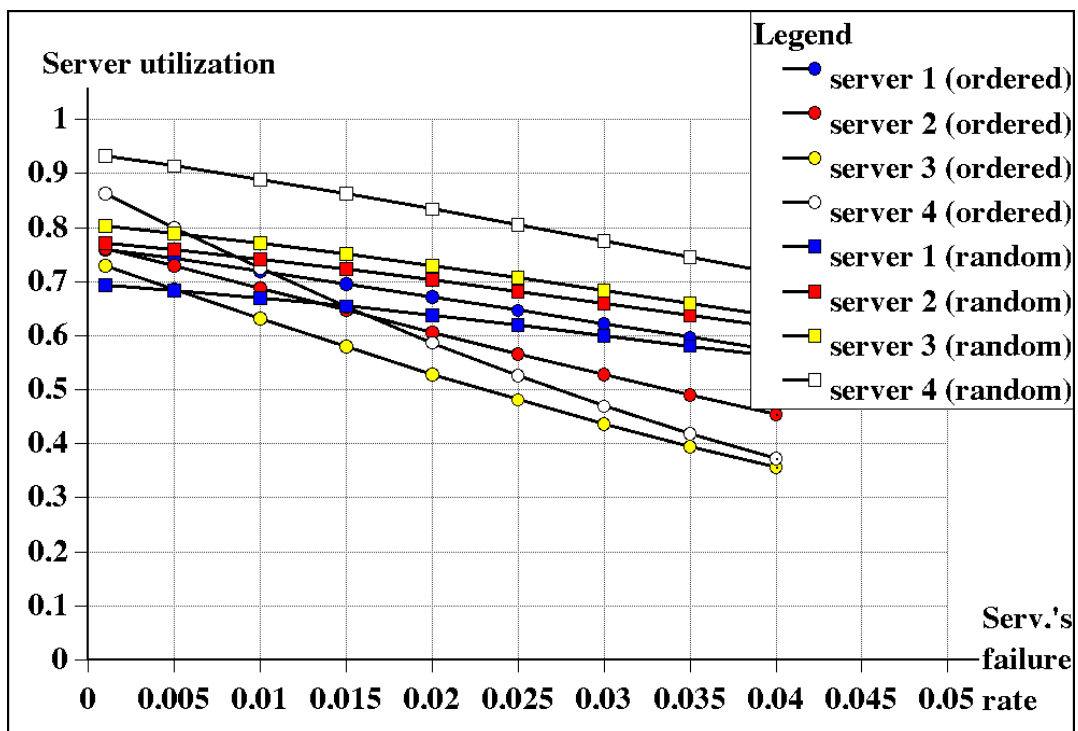


Figure 6: Server utilization versus servers' failure rate

assigned to the available and free servers with the same probability. In the beginning of the ordered case, the slowest server has the highest utilization too, but as it fails more often, its service is interrupted more often and loses from its utilization much faster than the faster servers, since it gets requests to serve only if all the other servers are busy or failed.

4.1.3 Analysis of Retrial Queues in Random Environment

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 $(1, \dots, r_m)$ and with transition density matrix

$$\left(\tau_{i_m j_m}^{(m)} \right), \quad i_m, j_m = 1, \dots, r_m, \quad \text{where} \quad \tau_{i_m i_m}^{(m)} = - \sum_{k \neq i_m} \tau_{i_m k}^{(m)}.$$

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 $\lambda_i(j_m)dt + o(dt)$, $m=1, \dots, M$. If the server is idle 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.

Thesis 3: *I have analyzed a finite-source retrial queueing system with heterogeneous sources operating in random environment, that is, the system parameters are subject to randomly occurring fluctuations. The description of the model and the main stationary performance and reliability measures were given by the help of the corresponding Markov chain. The results were validated by former works, and the effect of changing various parameters on some of these measures were illustrated graphically. Furthermore, the application of this model for the analysis of non-reliable systems was demonstrated. This model was published in [J7].*

In Figures 7 and 8 the mean response time is displayed as the primary request generation increases. The difference is that in Figure 7 all operations are stopped if the background process (which has two states) is in the second state, but in Figure 8, only service is interrupted. The results are in agreement with the results of [13], where the same parameters were used.

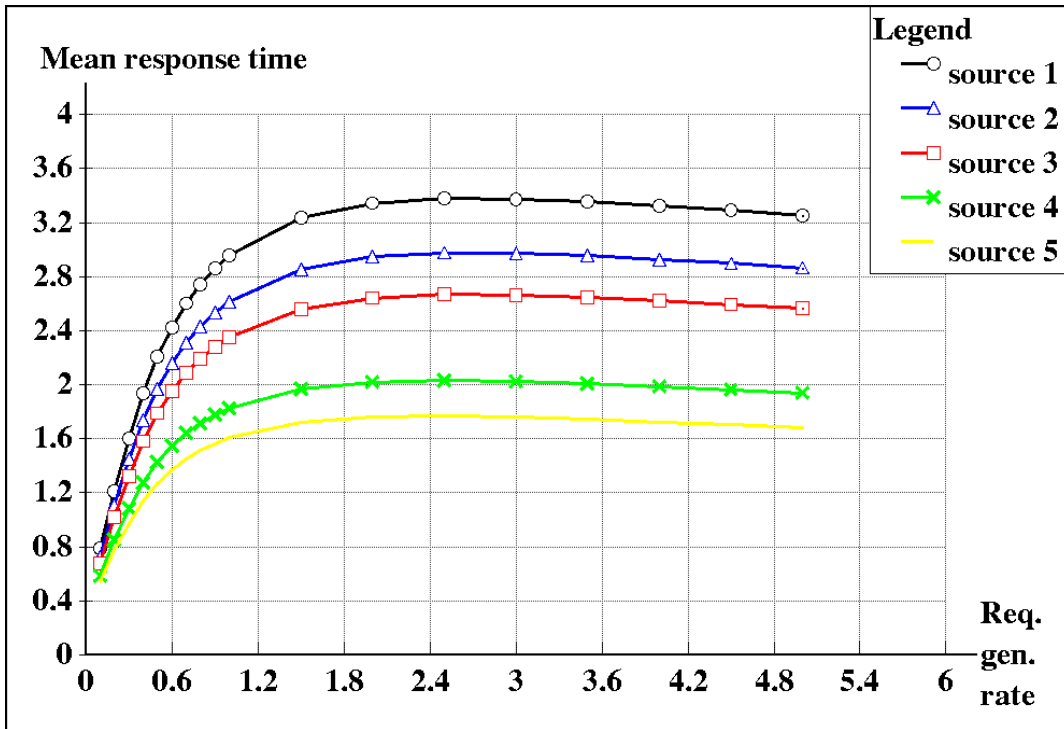


Figure 7: Mean response time versus primary request generation rate

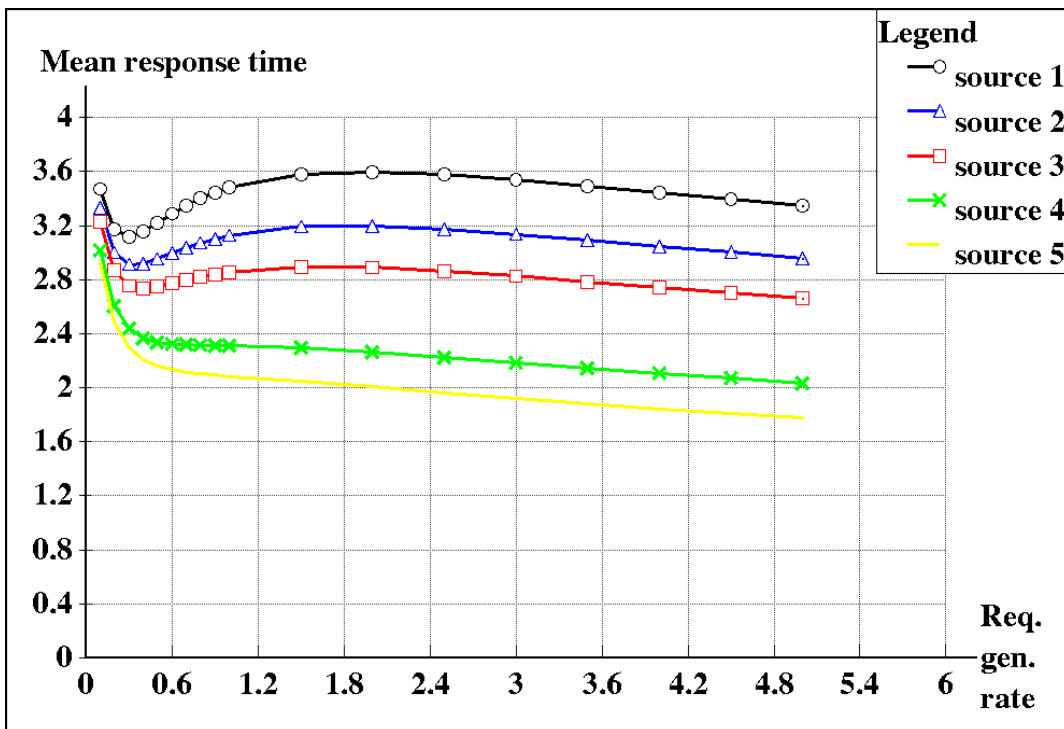


Figure 8: Mean response time versus primary request generation rate

4.2 Application of Retrial Queues in Performance Modelling of the GSM System

Queueing network models are widely used in the traffic modelling of cellular mobile systems, such as GSM (Global System for Mobile Communications), GPRS (General Packet Radio Service) and UMTS (Universal Mobile Telecommunication System). Most of the papers consider queueing systems without retrials, but after the study of Tran-Gia and Mandjes [20], which demonstrated in the context of cellular systems that the retrial phenomenon is not neglectable because of the significant negative influence on the system performance measures, authors more and more take it into consideration.

In the investigated cellular network model only one cell is treated. The cells are considered identical and to have the same traffic parameters, so it is enough to investigate one cell, and the handoff effect from the adjacent cells to this cell and from this cell to adjacent cells is described by handoff processes. Instead of the frequently used single arrival stream model the fresh call and handoff call arrivals are distinguished, what is gainful if we investigate complex call handling policies.

We assume, that the number of channels in the cell is C , and the number of guard channels (which are reserved channels for handoff calls) is g , where $g < C$.

The arrival process of the fresh calls is a Poisson process with rate λ_f . If the number of the active users is smaller than $C - g$, the incoming call starts to be served. Otherwise it is blocked and it starts generation of a Poisson flow of repeated calls (redialing) with probability Θ_1 . A blocked customer repeats his call after a random time which is exponentially distributed with mean $1/\nu_{bl}$, and it can be served or blocked again like the fresh calls. The call duration time is exponentially distributed with mean $1/\mu$.

The arrival process of the handoff calls is a Poisson process with rate λ_h . If the number of active users is smaller than C , the incoming call starts to be served. Otherwise it is dropped (handoff failure) and it starts generation of a Poisson flow of repeated calls with probability Θ_2 . A dropped customer tries to repeat his call after a random time which is exponentially distributed with mean $1/\nu_{dr}$. If it is blocked it continues redialing with probability Θ_2 . The call duration time for handoff calls is also exponentially distributed with mean $1/\mu$.

The active, redialing blocked and dropped customers leave the cell after an exponentially distributed time with mean $1/\mu_a$, $1/\mu_b$ and $1/\mu_d$, respectively.

The number of redialing users because of blocking and dropping is limited to an appropriately large values of N_{bl} and N_{dr} , to make the state space finite in order to make the calculations possible by the tools in the steady state.

Thesis 4: *I have evaluated the performance of the GSM system using a retrial queueing model. The accurate description of the cellular model was given along with the underlying Markov chain. It was shown how the model description can be translated into the description language of MOSEL-2. Some numerical examples were treated, where the results of the calculations were displayed graphically to demonstrate the effect of the changing of various system parameters on the quality of service measures. This model was based on former works of other authors with some model extensions, and published in [J2].*

In Figure 9, the fresh call blocking probability is displayed versus the number of channels with and without user redials. The negative influence of the retrial phenomenon is clearly

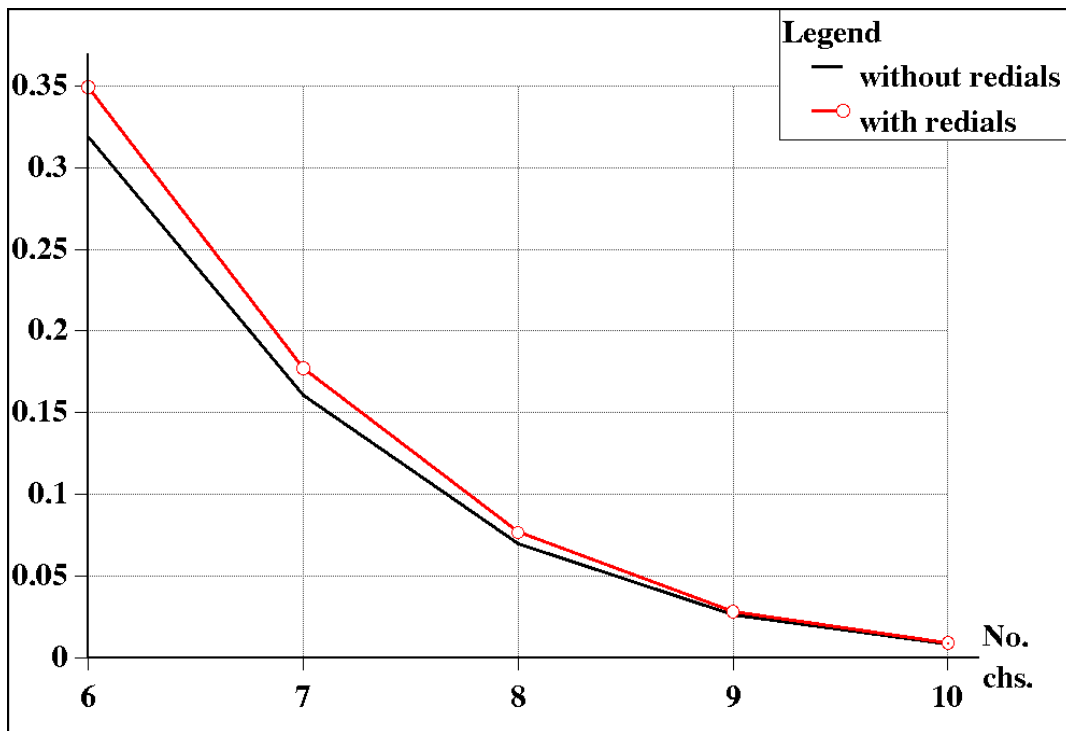


Figure 9: Fresh call blocking probability versus number of channels

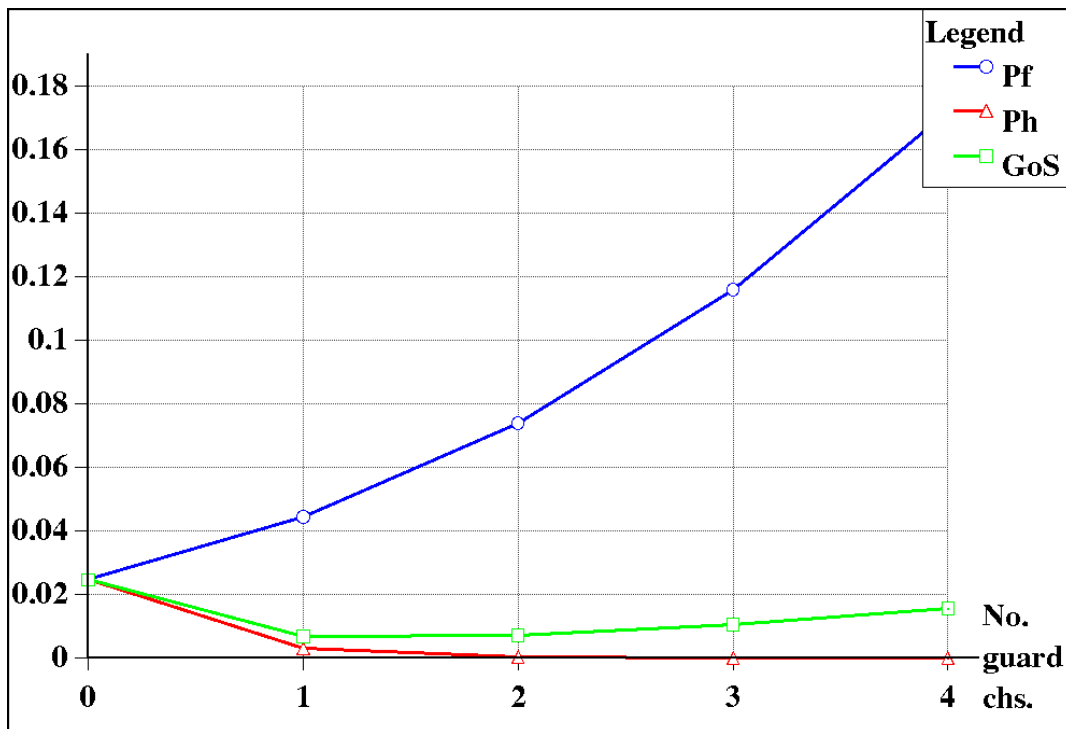


Figure 10: System measures versus number of guard channels

shown. The system input parameters belonging to the curve without redials are the same as in [8], where a similar model is studied without customer redials. These results are in agreement with theirs in the exponential case.

In Figure 10, the fresh call blocking and handoff call dropping probabilities and the GoS (Grade of Service) are displayed versus the number of guard channels. In this figure, we can see how many guard channels can improve the grade of service significantly, and then the other reserved channels can achieve only a very small handoff call dropping advance on the great expense of fresh call blocking probability, and the GoS declines.

5 Application of the Results

The models of the first part can be applied for the validation of more complex simulation models including non-exponential distributions.

The model of the second part can be used for the performance analysis of a real-life cellular mobile network. Besides, this work can be considered as the initial step towards the analysis of more complex third generation systems using MOSEL-2.

References

- [1] **Alfa A. S. and Li W.** PCS networks with correlated arrival process and retrial phenomenon, *IEEE Transactions on Wireless Communications*, Vol. 1 (2002) 630-637.
- [2] **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.
- [3] **Artalejo J.R.** New results in retrial queueing systems with breakdown of the servers, *Statistica Neerlandica*, Vol. 48 (1994) 23-36.
- [4] **Artalejo J.R.** Accessible bibliography on retrial queues, *Math. Comput. Modeling*, Vol. 30 (1999) 1-6.
- [5] **Begain K., Bolch G. and Herold H.** *Practical performance modeling, application of the MOSEL language*, Kluwer Academic Publisher, Boston, 2001.
- [6] **Begain K., Barner J., Bolch G., Zreikat A. I.** The Performance and Reliability Modelling Language MOSEL and its Application, *International Journal of Simulation*, Vol. 3 (2003) 66-80.
- [7] **Beutel B.** Integration of the Petri Net tool TimeNET into the MOSEL modelling environment, *MS Thesis*, Dept. of Computer Science, Univ. of Erlangen, Germany, 2003.
- [8] **Dharmaraja S., Trivedi K.S., Logothetis D.** Performance modeling of wireless networks with generally distributed handoff interarrival times, *Computer Communications*, Vol. 26 (2003) 1747-1755.
- [9] **Falin G.I.** A survey of retrial queues, *Queueing Systems*, Vol. 7 (1990) 127-168.

- [10] **Falin G.I. and Artalejo J.R.** A finite source retrial queue, *European Journal of Operational Research*, Vol. 108 (1998) 409-424.
- [11] **Falin G.I. and Templeton J.G.C.** *Retrial queues*, Chapman and Hall, London, 1997.
- [12] **Kulkarni V. G. and Choi B. D.** Retrial queues with server subject to breakdowns and repairs, *Queueing Systems Theory and Applications*, Vol. 7 (1990) 191-208.
- [13] **Sztrik J., Almási B., Roszik J.** Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Journal of Mathematical Sciences*, Vol. 132 (2006) 677-685.
- [14] **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.
- [15] **Kovalenko I.N., Kuznetsov N.Yu. and Pegg P.A.** *Mathematical theory of reliability of time dependent systems with practical applications*, John Wiley and Sons, Chichester, 1997.
- [16] **Marsan M. A., Carolis G. D., Leonardi E., Cigno R. L., Meo M.** Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials, *IEEE Journal on Selected Areas in Communications*, Vol. 19 (2001) 332-346.
- [17] **Ravichandran N.** *Stochastic methods in reliability theory*, John Wiley and Sons, New York, 1990.
- [18] **Sztrik J. and Gál T.** A recursive solution of a queueing model for a multi-terminal system subject to breakdowns, *Performance Evaluation*, Vol. 11 (1990) 1-7.
- [19] **Takagi H.** *Queueing Analysis, A Foundation of Performance Evaluation, Vol. 2., Finite Systems*, North-Holland, Amsterdam, 1993.
- [20] **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.
- [21] **Trivedi K. S.** *Probability and statistics with reliability, queueing and computer science applications*, Prentice-Hall, Englewood Cliffs, 1982.
- [22] **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.

Publications of the Author

International Journal Papers

- [J1] **Roszik J.** Homogeneous finite-source retrial queues with server and sources subject to breakdowns and repairs, *Annales Univ. Sci. Budapest., Sect. Comp.*, Vol. 23 (2004) 213-227.

- [J2] **Roszik J., Sztrik J., Kim C.S.** Retrial queues in the performance modeling of cellular mobile networks using MOSEL, *International Journal of Simulation: Systems, Science & Technology*, Vol. 6 (2005) 38-46.
- [J3] **Almási B., Roszik J., Sztrik J.** Homogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Mathematical and Computer Modelling*, Vol. 42 (2005) 673-682.
- [J4] **Sztrik J., Almási B., Roszik J.** Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Journal of Mathematical Sciences*, Vol. 132 (2006) 677-685.
- [J5] **Roszik J., Sztrik J.** Performance analysis of finite-source retrial queues with non-reliable heterogeneous servers, *Journal of Mathematical Sciences* (submitted for publication)
- [J6] **Sztrik J., Roszik J.** Finite-source retrial queueing systems with heterogeneous non-reliable servers and different service policies, *Journal of Mathematical Sciences* (submitted for publication)
- [J7] **Roszik J., Sztrik J., Virtamo J.** Performance analysis of finite-source retrial queues operating in random environments, *International Journal of Operations Research* (to appear)

International Conference Papers

- [C1] **Roszik J., Almási B., Sztrik J.** Multiserver retrial queues with finite number of heterogeneous sources, *Proceedings of 6th International Conference on Applied Informatics*, Eger, Hungary, (2004) Vol. II, 19-26.
- [C2] **Roszik J., Sztrik J.** The effect of server's breakdown on the performance of finite-source retrial queueing systems, *Proceedings of 6th International Conference on Applied Informatics*, Eger, Hungary, (2004) Vol. II, 221-229.
- [C3] **Hyytiä E., Lassila P., Penttinen A., Roszik J.** Traffic load in dense wireless multihop network, *Proceedings of The 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, Montreal, Quebec, Canada, (2005) 9-17.

Publications in Hungarian

- [H1] **Roszik J.** Visszatérési sorbanállási rendszerek a telekommunikációs hálózatok modellezésében, *Informatika a Felsőoktatásban 2005*, Debrecen, Hungary, (2005) 6 pages
- [H2] **Roszik J.** Számítógép-hálózatok gyakorlati segédanyag, *mobiDIÁK könyvtár*, Debrecen, Hungary, (2005) 43 pages

6. Bevezetés

A teljesítményelemzés fontos szerepet játszik a gyakorlati rendszerek – mint például a számítógép és telekommunikációs hálózatok – tervezésében, elemzésében és fejlesztésében. Gyakran alkalmaznak különböző sorbanállási modelleket ezen rendszerek teljesítményének és megbízhatóságának vizsgálatában. Bizonyos típusaiknál egyre gyakrabban használják a visszatérési sorbanállási rendszereket is, lásd például: [14; 16; 20]. Ennek az oka az, hogy a visszatérő igények többükénél – akár csak mint más gyakorlati alkalmazásoknál – speciális szerepet játszanak, és nem elhanyagolható negatív hatással vannak a teljesítményjellemzőkre.

A gyakorlati rendszerek egy másik fontos jellemzője a meghibásodhatóság, mely, mivel a rendszerkomponensek többsége általában meghibásodhat és javítást igényel, szintén negatív hatással van ezekre a jellemzőkre (lásd például: [15; 17; 21]). A nem megbízhatóság hatását alaposan tanulmányozták hagyományos, várakozási sorral rendelkező sorbanállási rendszerek-nél, visszatérő igényekkel viszont csak végtelen forrásúak esetén.

6.1. Visszatérési sorbanállási rendszerek

A visszatérő igényeket tartalmazó vagy visszatérési sorbanállási rendszerek olyan rendszerek, melyekben ha egy beérkező igény minden kiszolgálót foglaltnak talál, akkor elhagyja a kiszolgálókat, majd egy (véletlen) idő eltelte után megismétli a kérést. Több valós rendszer teljesítményelemzése esetén is – mint például a helyi és a celluláris mobil hálózatok – alkalmazhatók a visszatérési sorbanállási modellek. Az ilyen típusú sorbanállási rendszerekkel kapcsolatos részletes információkért és eredményekért lásd például: [4; 9; 11], a hagyományos, visszatérő igényeket nem tartalmazó rendszerekkel kapcsolatban pedig: [19]. Végtelen forrású, nem megbízható visszatérési modellekkel kapcsolatos irodalom található továbbá a [2; 3; 12; 22] munkákban.

A disszertációban két visszatérési sorbanállási modellt tekintünk át. Az első az a véges forrású sorbanállási rendszer, amelyen az első részben lévő elemzések alapulnak. A második pedig egy végtelen forrású modell, melynek egy kiterjesztett változatát alkalmazzuk a második részben a GSM rendszer (Global System for Mobile Communications) elemzésére.

7. Kutatási célok

A disszertáció első részének célja néhány nem megbízható, véges forrású visszatérési modellnek, valamint egy megbízható visszatérési rendszernek véletlen környezetben való vizsgálata volt. Az utóbbi szintén alkalmazható nem megbízható rendszerek teljesítményelemzésében. A szerző legjobb tudása szerint ezeket a modelleket nem tárgyalták korábbi munkákban.

A másik cél egy visszatérési sorbanállási modell alkalmazása volt egy gyakorlati rendszer modellezésében. A második részben egy valós hálózatot modelleztem egy ilyen, visszatérő igényeket tartalmazó sorbanállási rendszer segítségével. Itt egy modellezési módot tekintek át a GSM rendszer elemzésére egy modellezőeszköz felhasználásával. Ezt mások korábbi munkáira építve (például [1; 16; 20]), néhány modellkiterjesztést beépítve valósítottam meg.

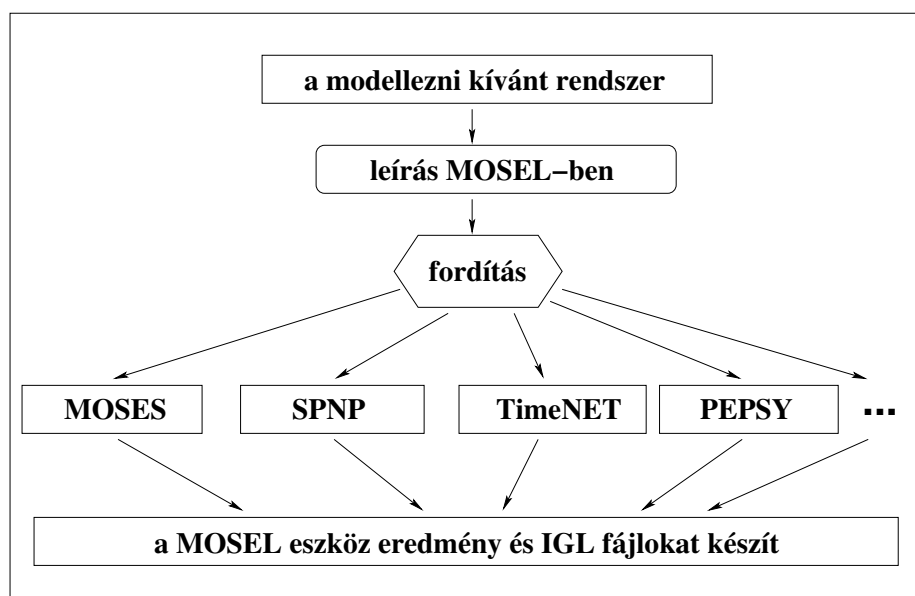
8. Az alkalmazott módszerek

Mivel a vizsgált sorbanállási modelleket leíró Markov-láncok állapottere nagyon nagy, és a

rendszerek működése bonyolult, ezért a hagyományos módon, azaz az egyensúlyi állapotegyenlet-rendszer megoldásával meglehetősen nehéz kiszámítani az egyensúlyi állapotvalószínűségeket. Hogy egyszerűsítsük ezeket a számításokat, továbbá a gyakorlatban használhatóbbá tegyük az elemzéseket, a MOSEL (Modeling, Specification and Evaluation Language) [5] eszközt használjuk a modellek leírására és a jellemzők kiszámítására. Ennek az eszköznek segítségével két lépést egyben oldhatunk meg, így nem szükséges felírunk és valahogyan megoldanunk az egyensúlyi állapotegyenlet-rendszert. A modellezés nehézsége abban rejlik, hogy hogyan írjuk le a rendszert és viselkedését az eszköz számára.

A MOSEL egy olyan magasszintű nyelvvel rendelkező modellezési környezet, mely lehetővé teszi valós, bonyolult rendszerek leírását, és ez alapján kiszámítja a keresett rendszerjellemzőket más hatékonyságvizsgálati eszközök felhasználásával. A MOSEL leírás automatikusan lefordítható a különböző eszközök nyelvére (jelenleg az SPNP – Stochastic Petri Net Package és a TimeNET alkalmazható a vizsgált modellek esetén), majd az adott eszközt felhasználva elemezhetjük a modellt és kaphatjuk meg a keresett rendszerjellemzőket.

A MOSEL eszközt már számos számítógép és kommunikációs rendszer modellezésére használták és bizonyította alkalmazhatóságát. Működését és használatát az itt látható 11. ábra szemlélteti. A programozás technikai részletei megtalálhatók [5]-ben, valamint [6; 7]-ben, ahol az új, átdolgozott verziót (MOSEL-2) ismertetik.



11. ábra. A modellezési folyamat a MOSEL környezetben

A disszertáció első részében az eredeti eszközt használtam véges forrású visszatérési sorbanállási rendszerek elemzésére. A második részben az átdolgozott modellezőnyelvet alkalmaztam – részletes magyarázatokkal a MOSEL-2 programozással kapcsolatban – a GSM rendszer modellezésére.

9. Új eredmények

9.1. Visszatérési sorbanállási rendszerek elemzése

Mivel a valós rendszerek komponensei általában meghibásodhatnak, ezért a korlátozott javítási

lehetőségek, és a meghibásodásoknak a teljesítményjellemzőkre gyakorolt jelentős hatása miatt fontos a nem megbízható sorbanállási rendszerek vizsgálata, így a nem megbízható visszatérési modelleké is.

9.1.1. Egykiszolgálós, nem megbízható, véges forrású visszatérési rendszerek elemzése

Tekintsünk egy egykiszolgálós sorbanállási rendszert, ahol K , $1 < K < \infty$ homogén forrás generál igényeket. A kiszolgáló lehet működő vagy meghibásodott, továbbá szabad vagy foglalt. Szabad és működő állapotban kiszolgálhatja a források igényeit. Az igényforrások állapota a következő lehet: szabad, kiszolgálásra várakozó és kiszolgálás alatt lévő. Ha egy forrás szabad a t időpillanatban, akkor a $(t, t + dt)$ időintervallumban $\lambda dt + o(dt)$ valószínűséggel küld új elsődleges igényt. Ha a kiszolgáló működik és szabad az igény beérkezésekor, akkor a kiszolgálás azonnal elkezdődik, a forrás kiszolgálás alatt lévő, a kiszolgáló pedig foglalt állapotba kerül. Ha a kiszolgáló működik a kiszolgálás a $(t, t + dt)$ időintervallumban $\mu dt + o(dt)$ valószínűséggel fejeződik be. Ha a kiszolgáló foglalt, akkor a forrás egy ν paraméterű Poisson folyamat szerint újraküldi a kérést amíg nem találja azt szabadnak. Kiszolgálás után a forrás szabadabbá válik és új elsődleges igényt generálhat, a szerver pedig egy új kérést szolgálhat ki. A kiszolgáló a $(t, t + dt)$ időintervallumban $\delta dt + o(dt)$ valószínűséggel hibásodik meg ha szabad, és $\gamma dt + o(dt)$ valószínűséggel ha foglalt. Ha $\delta = 0, \gamma > 0$ vagy $\delta = \gamma > 0$ akkor *aktív, illetve független meghibásodásról* beszélünk. Ha a szerver foglalt állapotban hibásodik meg, akkor javítás után vagy *folytatja a megszakadt kiszolgálást*, vagy pedig *az igényforrás kiszolgálásra várakozó állapotba kerül*. A javítási idő exponenciális eloszlású, $1/\tau$ várható értékkel. Ha a kiszolgáló meghibásodott, két különböző esetet tekintünk. Egyrészt a *blokkolt források* esetét, amikor sem új, sem ismétlődő igényeket nem lehet küldeni, valamint a *nem blokkolt (intelligens) forrásokét*, ahol csak a kiszolgálás szünetel. A modellben szereplő időket egymástól függetleneknek tekintjük.

Ez a modell a [18] által vizsgált homogén, véges forrású, nem visszatérési de kiszolgáló-meghibásodást tartalmazó sorbanállási rendszer kiterjesztése, továbbá [11] megbízható modellel kapcsolatos eredményeit általánosítja. Látható, hogy ez a rendszer jóval bonyolultabb a megbízható esetnél, mivel kétfajta meghibásodást, valamint meghibásodás esetén folytatásos és ismétlődő kiszolgálást, továbbá blokkolt és nem blokkolt működést is tartalmaz.

Egyrészt, ez a modell ki lett egészítve nem megbízható forrásokkal is, melyek szabad állapotban hibásodhatnak meg. Ha egy igényforrás szabad a t időpillanatban, akkor a $(t, t + dt)$ időintervallumban $\eta dt + o(dt)$ valószínűséggel hibásodik meg. Ekkor a javítóegységhez kerül, ami FIFO elv szerint javítja a forrásokat, és megszakításos elsőbbséget ad a kiszolgáló javításának. A források javítási ideje exponenciális eloszlású $1/\kappa$ várható értékkel.

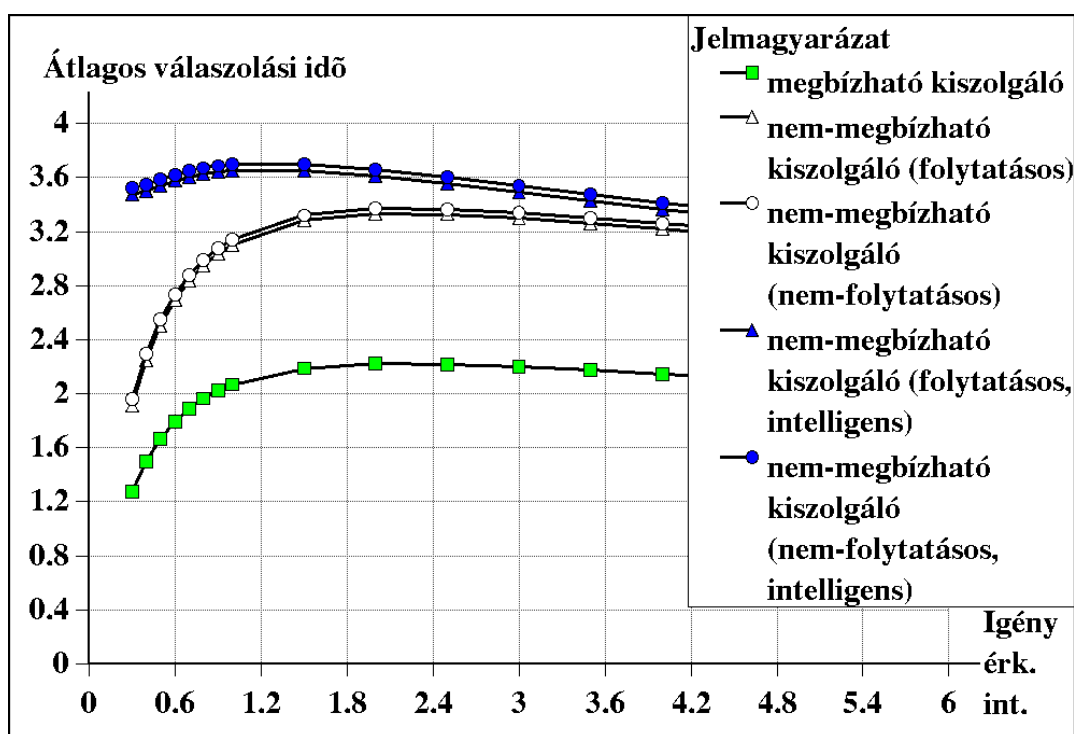
A másik módosítás a modell heterogén (megbízható) forrásokkal való általánosítása volt. Ebben az esetben az i -edik forrás a $(t, t + dt)$ időintervallumban $\lambda_i dt + o(dt)$ valószínűséggel küld új igényt. Ha a kiszolgáló szabad az igény érkezésekor, akkor a kiszolgálás azonnal elkezdődik, az igényforrás kiszolgálás alatt, a szerver pedig foglalt állapotba kerül. Ha a kiszolgáló működik, a kiszolgálás a $(t, t + dt)$ időintervallumban $\mu_i dt + o(dt)$ valószínűséggel fejeződik be. Ha viszont foglalt, akkor a forrás egy ν_i paraméterű Poisson folyamat szerint újraküldi a kérést amíg nem találja szabadnak.

1. tézis: Elemeztem néhány egykiszolgálós, nem megbízható, véges forrású visszatérési sorbanállási rendszert. A modellek leírása és a legfontosabb egyensúlyi megbízhatósági és teljesítményjellemzők megadása a megfelelő Markov-láncok segítségével történt. Az eredményeket korábbi munkák segítségével ellenőriztem, továbbá a különböző paraméterek változásának néhány jellemzőre gyakorolt hatását grafikusán ábrázoltam. Ezek a modellek a következő cikkekben lettek publikálva: [J1; J3; J4].

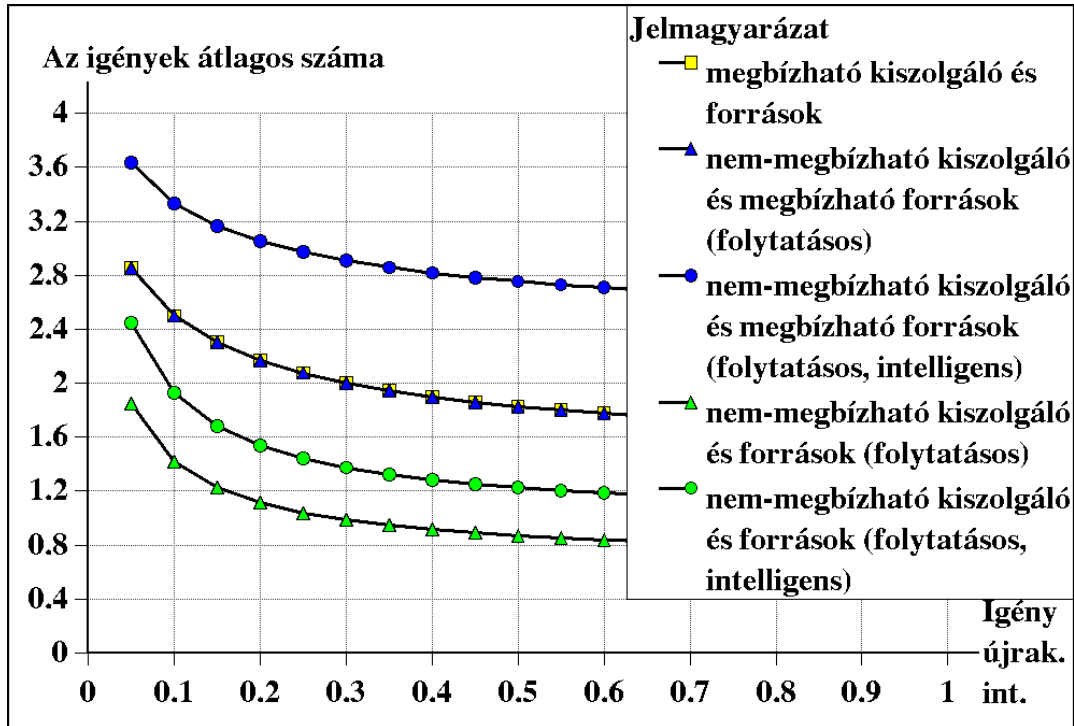
A következőkben ismertetek két elemzési példát.

A 12. ábrán az átlagos válaszolási idő látható megbízható és nem megbízható kiszolgáló esetén. Meghibásodás esetén a kiszolgálás vagy folytatódik vagy megszakad, továbbá az igényforrások vagy blokkolódnak vagy pedig küldhetnek kéréseket (intelligens eset) amíg a kiszolgáló nem működik. Az ábrán megfigyelhető az a [10]-ben is leírt meglepő jelenség, miszerint a véges forrású visszatérési sorbanállási rendszereknek van átlagos válaszidő maximuma. Továbbá jól látható az előbbi öt típusú rendszer közötti különbség.

A 13. ábrán az igény újraküldési intenzitásnak a várakozó vagy kiszolgálás alatt lévő igények átlagos számára gyakorolt hatása figyelhető meg. Jól látható a különbség a megbízható és a nem megbízható kiszolgálót, valamint a nem megbízható kiszolgálót és igényforrásokat tartalmazó esetek között. A kiszolgáló javítás után folytatja az éppen kiszolgált igényt, továbbá az igényforrások vagy blokkolódnak vagy pedig küldhetnek kéréseket a meghibásodás alatt. Érdekes kiemelni, hogy a megbízható, és a nem megbízható kiszolgálót de megbízható forrásokat tartalmazó blokkolt esethez tartozó görbék egybeesnek. Ennek az oka az, hogy itt a meghibásodás alatt az igények száma nem változik.



12. ábra. Átlagos válaszolási idő az új igény érkezési intenzitás függvényében



13. ábra. Kiszolgálás alatt lévő vagy várakozó igények átlagos száma az igény újraküldési intenzitás függvényében

9.1.2. Többkiszolgálós, nem megbízható, véges forrású visszatérési rendszerek elemzése

Tekintsünk egy véges forrású visszatérési sorbanállási rendszert c kiszolgálóval, ahol $K, c < K < \infty$ forrás generál igényeket. Mindegyik kiszolgáló lehet működő vagy meghibásodott, továbbá szabad vagy foglalt. Az igényforrások három különböző állapotban lehetnek: szabad, kiszolgálásra várakozó és valamelyik server által kiszolgálás alatt lévő. Ha egy forrás szabad a t időpillanatban, akkor a $(t, t + dt)$ időintervallumban $\lambda dt + o(dt)$ valószínűséggel küld új elsődleges igényt. Ha valamelyik kiszolgáló működik és szabad az igény beérkezésekor, akkor a kiszolgálás azonnal elkezdődik. Ha az i -edik kiszolgáló működik, a kiszolgálás a $(t, t + dt)$ időintervallumban $\mu_i dt + o(dt)$ valószínűséggel fejeződik be.

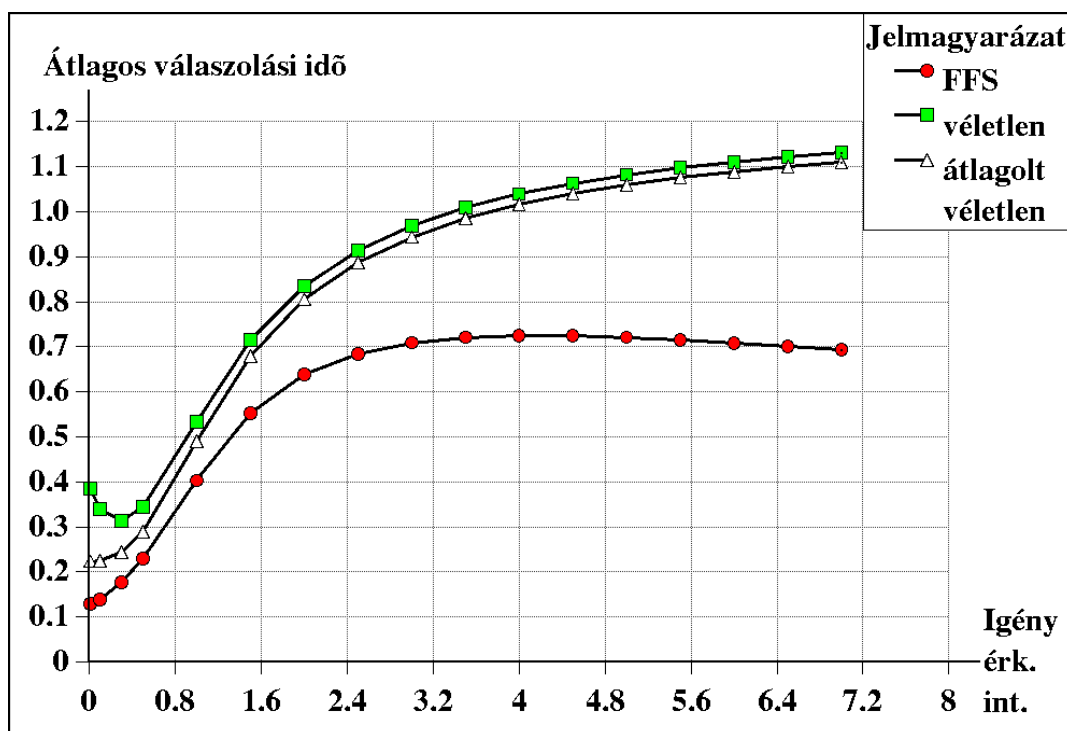
Az i -edik kiszolgáló a $(t, t + dt)$ időintervallumban $\delta_i dt + o(dt)$ valószínűséggel hibásodik meg ha szabad, és $\gamma_i dt + o(dt)$ valószínűséggel ha foglalt. Ha egy server foglalt állapotban hibásodik meg, akkor javítás után vagy folytatja a megszakadt kiszolgálást, vagy pedig az igényforrás kiszolgálásra várakozó állapotba kerül. Itt csak az utóbbi esetet tekintjük. A javítóegység FIFO elv szerint javítja a kiszolgálókat. Az i -edik server javítási ideje exponenciális eloszlású $1/\tau_i$ várható értékkel. Ha mindegyik kiszolgáló meghibásodott két különböző esetet tekintünk. Egyrészt a *blokkolt források* esetét, ahol az először meghibásodott server javításán kívül minden blokkolódik, valamint a *nem blokkolt (intelligens) forrásokét*, ahol csak a kiszolgálás szünetel.

Ha minden kiszolgáló foglalt (vagy meghibásodott a nem blokkolt esetben) az igény érkezésekor, akkor a forrás egy ν paraméterű Poisson folyamat szerint újraküldi a kérést amíg nem talál egy működő és szabad kiszolgálót. Kiszolgálás után a forrás szabaddá válik és új elsődleges igényt generálhat, a kiszolgáló pedig egy új kérést szolgálhat ki. A modellben

szereplő időket egymástól függetleneknek tekintjük.

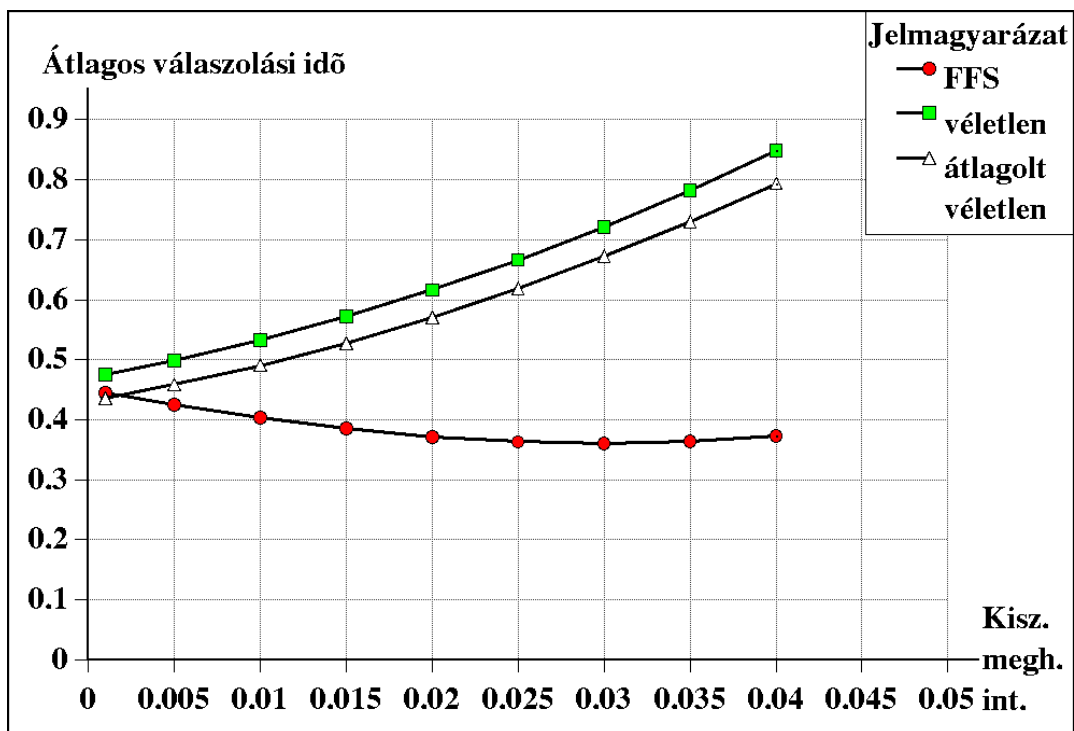
2. tézis: *Elemeztem egy többkiszolgálós, nem megbízható, véges forrású visszatéréssel sorbanállási rendszert. A rendszert nem megbízható heterogén (aszimmetrikus) kiszolgálókkal – melyek különböző kiszolgálási, meghibásodási és javítási intenzitásokkal rendelkezhetnek – vizsgáltam. A modell leírása és a legfontosabb egyensúlyi megbízhatósági és teljesítményjellemzők megadása a megfelelő Markov-lánc segítségével történt, az eredményeket pedig korábbi munkák alapján ellenőriztem. A különböző paraméterek változásának néhány jellemzőre gyakorolt hatását grafikusán ábrázoltam, továbbá két kiszolgálási elvet hasonlítottam össze ennél a modellnél. Az eredmények [J5; J6]-ban lesznek publikálva.*

A 14. ábrán a leggyorsabb szabad kiszolgáló (Fastest Free Server – FFS), a véletlen (Random) és az átlagolt véletlen (itt mindegyik kiszolgálási intenzitás a heterogén eset intenzitásainak átlaga) kiszolgálási elvek esetén láthatjuk az átlagos válaszolási időt az új igény érkezési intenzitás függvényében. A véletlen kiszolgálási elvek esetén az igények véletlenszerűen kerülnek az egyes szabad kiszolgálókhoz. A másik elv esetén viszont a beérkező igény a leggyorsabb szabad kiszolgálóhoz kerül. Látható, hogy a különbség a két véletlen eset között nem jelentős, de a leggyorsabb szabad kiszolgálót választó elv mindig jobb átlagos válaszidővel rendelkezik, különösen mikor egyre több és több igény érkezik.

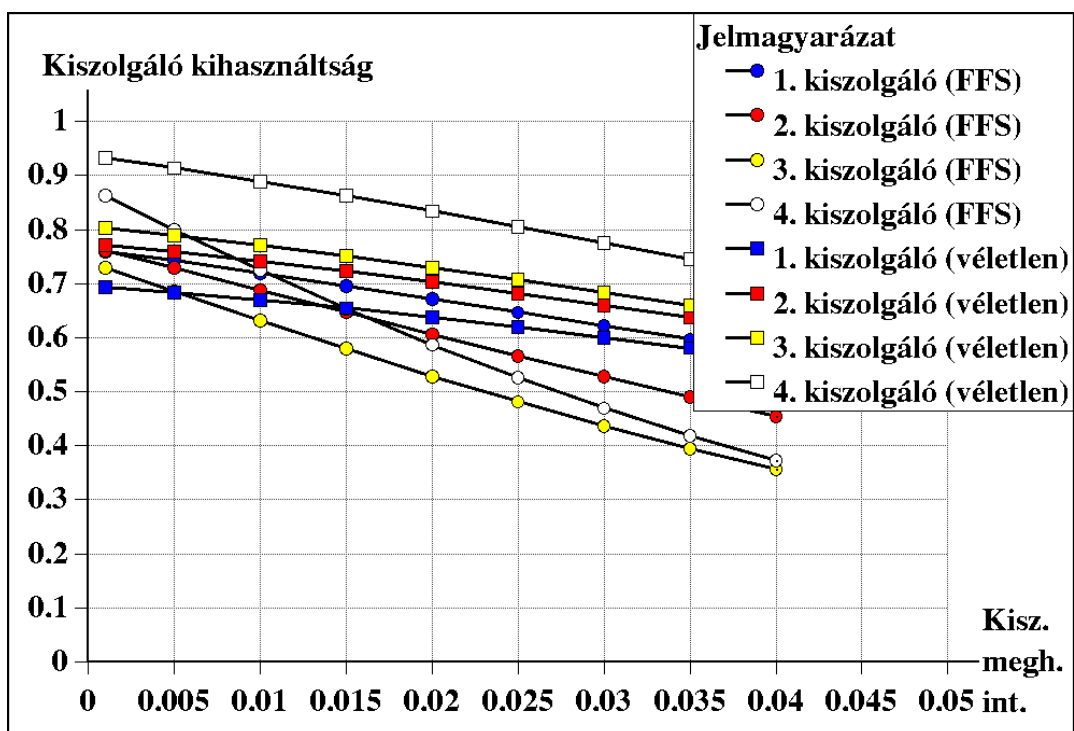


14. ábra. Átlagos válaszolási idő az új igény érkezési intenzitás függvényében

A 15. ábrán a kiszolgálók meghibásodási intenzitásának az átlagos válaszolási időre gyakorolt hatása figyelhető meg. Az átlagolt véletlen eset kissé jobb válaszidővel rendelkezik mint a nem átlagolt, hasonlóan az előző ábrához. A leggyorsabb szabad kiszolgáló esetenél történő meglepő válaszidőcsökkenés a 16. ábra segítségével magyarázható.



15. ábra. Átlagos válaszolási idő a kiszolgáló-meghibásodási intenzitás függvényében



16. ábra. Kiszolgáló-kihasználtság a kiszolgáló-meghibásodási intenzitás függvényében

A 16. ábrán a kiszolgálók kihasználtságának változása látható a meghibásodási intenzitásuknak a függvényében. A rendszer paraméterei ugyanazok mint amelyek a 15. ábránál is voltak. A véletlen esetnél a leglassabb kiszolgáló kihasználtsága a legnagyobb, és a leggyorsabbé a legkisebb, mivel az utóbbi sokkal gyorsabban szolgálja ki az igényeket, melyek viszont azonos valószínűséggel kerülnek mindegyik működő és szabad kiszolgálóhoz. Kezdetben a leggyorsabb szabad kiszolgáló elv esetén is a leglassabb kiszolgáló kihasználtsága a legnagyobb, de amint az egyre gyakrabban hibásodik meg, egyre gyorsabban veszít belőle, mivel csak akkor kap igényt kiszolgálásra, ha az összes többi már foglalt vagy meghibásodott.

9.1.3. Véletlen környezetben lévő visszatérési rendszerek elemzése

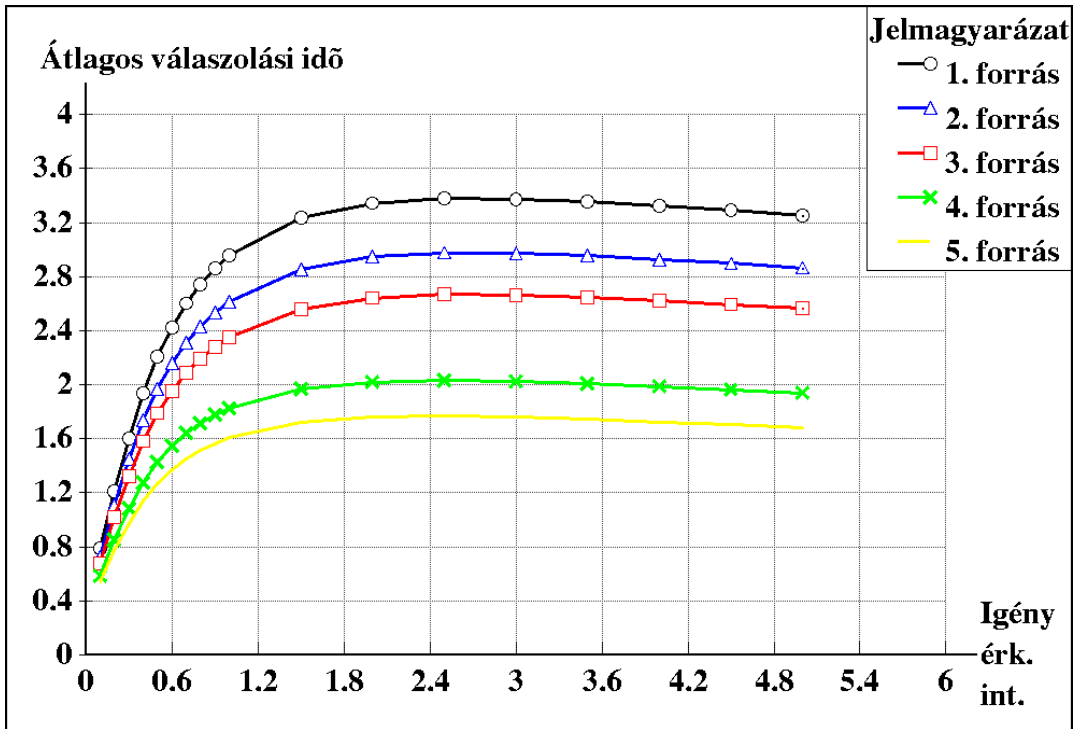
Tekintsünk egy véges forrású sorbanállási rendszert, K forrással és egy kiszolgálóval, ahol minden forrás különböző paraméterekkel rendelkezik, és a források és a kiszolgáló működését is befolyásolja egy adott véletlen környezet állapota. A kiszolgálót és a forrásokat M ($1 \leq M \leq K + 1$) független csoportba gyűjtjük. Az egyes csoportok tagjai működnek az adott véletlen környezetekben. A környezet változásai az új igény generálási, az igény újraküldési és a kiszolgálási intenzitások értékeiben tükröződnek. Feltételezzük, hogy az m -edik csoport tagjai olyan véletlen környezetben működnek, amelyet egy $(\xi_m(t); t \geq 0)$ Markov-lánc ír le $(1, \dots, r_m)$ állapotterrel és a következő átmenet-intenzitás mátrixszal:

$$\left(\tau_{i_m j_m}^{(m)} \right), \quad i_m, j_m = 1, \dots, r_m, \quad \text{ahol} \quad \tau_{i_m i_m}^{(m)} = - \sum_{k \neq i_m} \tau_{i_m k}^{(m)}.$$

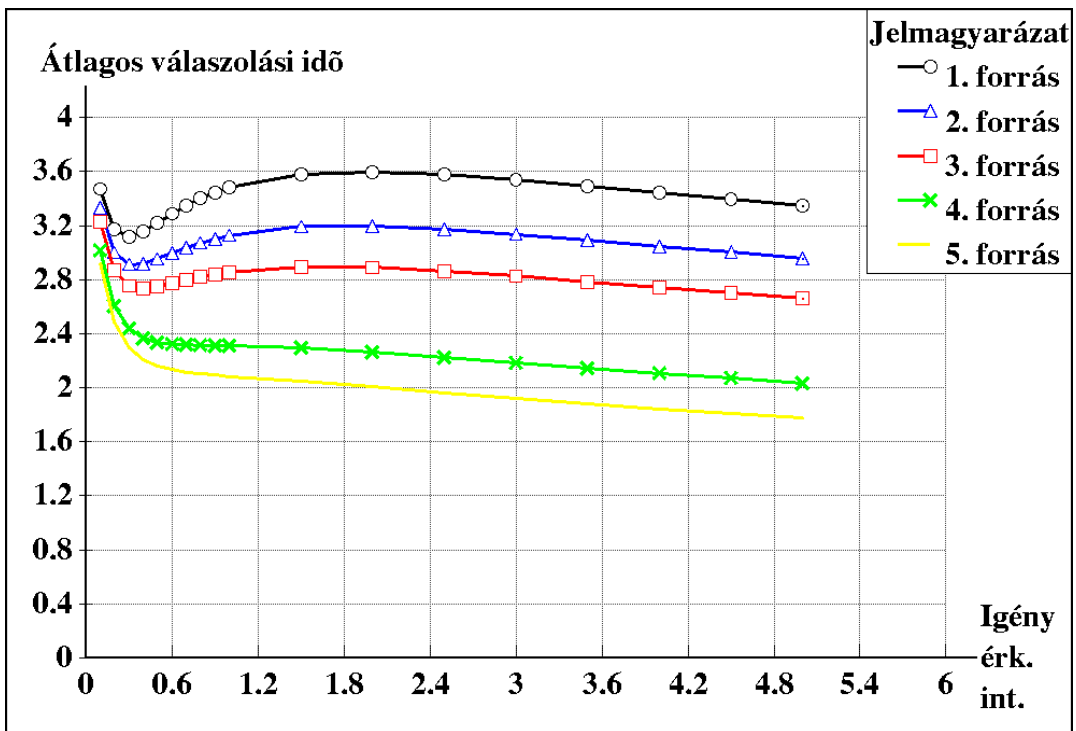
A kiszolgáló két különböző állapotban lehet: szabad és foglalt. Az igényforrások állapota pedig: szabad, kiszolgálásra várakozó és kiszolgálás alatt lévő lehet. Ha az i -edik forrás (amely az m -edik csoport tagja) szabad a t időpillanatban, és az $\xi_m(t)$ háttér folyamat a j_m állapotban van, akkor a $(t, t + dt)$ időintervallumban $\lambda_i(j_m)dt + o(dt)$, $m=1, \dots, M$ valószínűséggel küld új elsődleges igényt. Ha a kiszolgáló szabad az igény beérkezésekor, akkor a kiszolgálás azonnal elkezdődik, azaz a forrás kiszolgálás alatt lévő, a kiszolgáló pedig foglalt állapotba kerül. Feltételezve hogy a szerver az 1. csoportba tartozik, és az $\xi_1(t)$ háttér folyamat a j_1 állapotban van, az i -edik forráshoz tartozó igény kiszolgálása a $(t, t + dt)$ időintervallumban $\mu_i(j_1)dt + o(dt)$ valószínűséggel fejeződik be. Ha a kiszolgáló foglalt az igény érkezésekor, akkor a forrás egy $\nu_i(j_m)$ paraméterű Poisson folyamat szerint újraküldi a kérést amíg a kiszolgálót szabadnak nem találja. Kiszolgálás után a forrás és a kiszolgáló is szabaddá válik és új elsődleges igényt küldhet illetve új kérést szolgálhat ki. A modellben szereplő változókat és a véletlen környezeteket egymástól függetleneknek tekintjük.

3. tézis: *Elemeztem egy véges forrású visszatérési sorbanállási rendszert heterogén forrásokkal véletlen környezetben (azaz ahol a rendszer-paraméterek véletlen ingadozásoknak vannak kitéve). A modell leírása és a legfontosabb egyensúlyi megbízhatósági és teljesítmény-jellemzők megadása a megfelelő Markov-lánc segítségével történt. Az eredményeket korábbi munkák alapján ellenőriztem, és a különböző paraméterek változásának néhány jellemzőre gyakorolt hatását grafikusán ábrázoltam. Emellett a modellnek a nem megbízható rendszerek vizsgálatára való alkalmazási lehetőségét is bemutattam. A modell [J7]-ben lett publikálva.*

A 17. és 18. ábrákon az átlagos válaszolási idő látható az új igény érkezési intenzitás függvényében. Az ábrák között az a különbség, hogy az elsőnél minden blokkolódik, ha a kétállapotú háttér folyamat a második állapotában van, a másik ábra esetén viszont csak a ki-



17. ábra. Átlagos válaszolási idő az új igény érkezési intenzitás függvényében



18. ábra. Átlagos válaszolási idő az új igény érkezési intenzitás függvényében

szolgáltatás szünetel. Azonos rendszer-paraméterek esetén az eredmények összhangban vannak [13] eredményeivel.

9.2. Visszatéréses sorbanállási rendszerek alkalmazása a GSM rendszer modellezésében

A sorbanállási hálózati modelleket gyakran alkalmazzák a celluláris mobil rendszerek, mint például a GSM (Global System for Mobile Communications), a GPRS (General Packet Radio Service) és az UMTS (Universal Mobile Telecommunication System) teljesítményelemzésében. Elsősorban a klasszikus, visszatérő igényeket nem tartalmazó sorbanállási modelleket használják, de Tran-Gia és Mandjes [20] cikke után egyre többen veszik figyelembe a visszatérő igényeknek a teljesítményjellemzőkre gyakorolt hatását.

A vizsgált hálózati modellben csak egy cellát tekintünk. Az többi cellát ezzel megegyezőnek, azaz azonos forgalomparaméterekkel rendelkezőnek tekintjük, a szomszédosakból valamint a szomszédosakba történő hívásátadást pedig hívásátadási folyamatokkal írjuk le. A leggyakrabban használt, a beérkező igényeket egy beérkezési folyamatba összevonó modell helyett megkülönböztetjük az új és az átadott hívásérkezési folyamatokat, ami előnyös bonyolultabb csatornakiosztási technikák alkalmazása esetén.

Feltételezzük, hogy a csatornák száma egy cellában C , az átadott hívások számára fenntartott csatornák száma pedig g , ahol $g < C$.

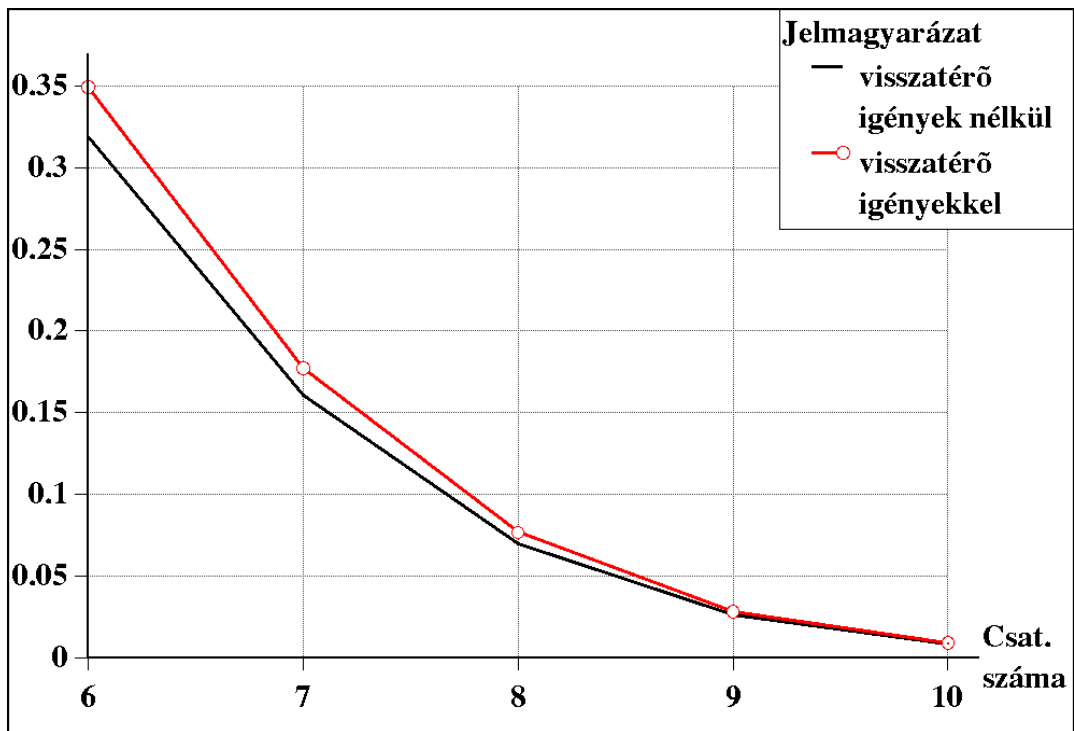
Az új hívások beérkezési folyamata egy Poisson folyamat λ_f paraméterrel. Ha az aktív felhasználók, azaz a foglalt csatornák száma kisebb mint $C - g$, akkor megkezdődik a hívás felépítése, egyébként pedig blokkolódik. Blokkolás esetén a felhasználó Θ_1 valószínűséggel újra próbálkozik, amit egy Poisson folyamattal modellezünk. Itt a várakozási idő exponenciális eloszlású, $1/\nu_{bl}$ várható értékkel. Az ismétlődő hívások az újakhoz hasonlóan kerülhetnek kiszolgálásra, valamint újra blokkolt állapotba. A hívások időtartama exponenciális eloszlású, $1/\mu$ várható értékkel.

A beérkező átadott hívások beérkezési folyamata egy λ_h paraméterű Poisson folyamat. Ha az aktív felhasználók száma kisebb mint C , akkor megkezdődik a hívás felépítése, egyébként megszakad. Ekkor a felhasználó Θ_2 valószínűséggel megpróbálja újra felépíteni a kapcsolatot, amit Poisson folyamattal írunk le. Itt a várakozási idő exponenciális eloszlású, $1/\nu_{dr}$ várható értékkel. Blokkolás esetén a felhasználó Θ_2 valószínűséggel próbálkozik újra. Az átadott hívások időtartama szintén exponenciális eloszlású, $1/\mu$ várható értékkel.

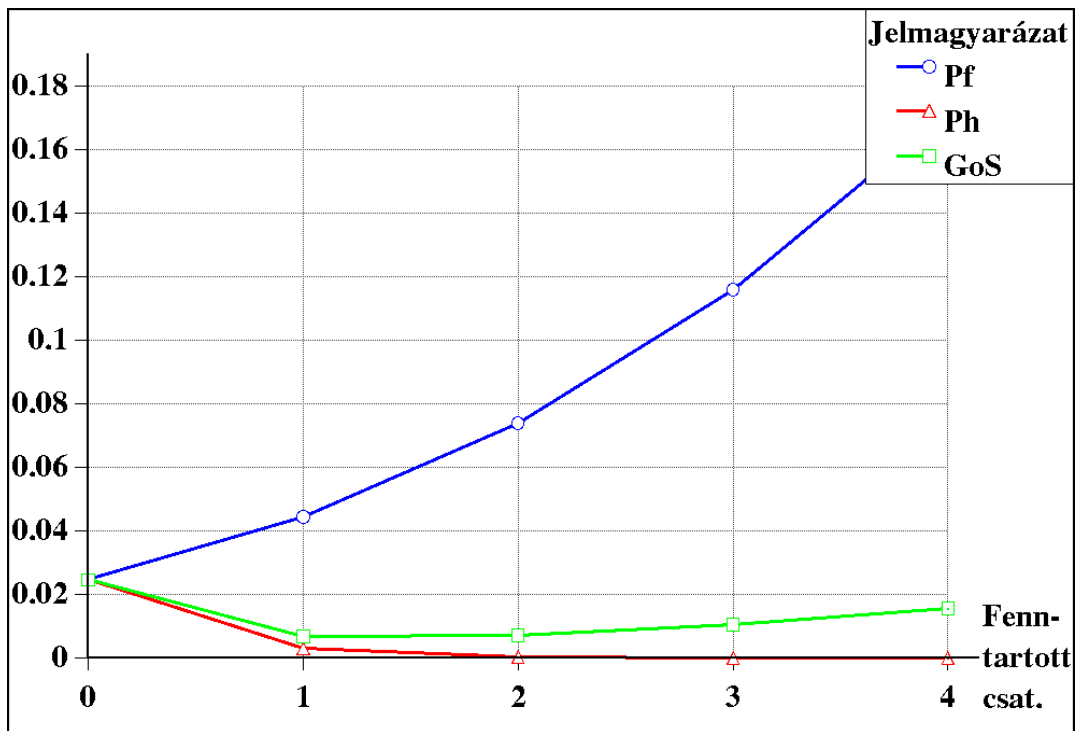
Az aktív, blokkolás és hívásmegszakadás miatt újra próbálkozó felhasználók exponenciális eloszlású idők után hagyják el az aktuális cellát, $1/\mu_a$, $1/\mu_b$ valamint $1/\mu_d$ várható értékekkel.

A blokkolás és hívásmegszakadás miatt újra próbálkozó felhasználók maximális számát megfelelően nagy N_{bl} és N_{dr} értékekkel korlátozzuk. Így az állapottér véges lesz, és lehetővé tesszük a számításokat az alkalmazott eszközök számára az egyensúlyi állapotban.

4. tézis: *Egy visszatéréses sorbanállási modell felhasználásával elemeztem a GSM rendszer teljesítményét. Megadtam a cella alapú modell pontos leírását az azt leíró Markov-lánccal együtt, majd megmutattam, hogyan fordítható le ez a MOSEL-2 leíró nyelvére. Elemeztem néhány numerikus példát, ahol az eredményeket grafikusán ábrázolva mutattam meg különböző paraméterek változásának hatását a szolgáltatás minőségének jellemzőire. A modell mások korábbi munkáira alapozva készült, és [J2]-ben lett publikálva.*



19. ábra. Az új hívások blokkolásának valószínűsége a csatornaszám függvényében



20. ábra. Rendszerjellemzők a fenntartott csatornák számának függvényében

A 19. ábrán az új hívások blokkolásának valószínűsége látható a csatornaszám függvényében, visszatérő igényekkel és visszatérő igények nélkül. Az ábrán jól megfigyelhető a visszatérési jelenség negatív hatása. A visszatérő igényeket nem tartalmazó görbéhez tartozó rendszerparaméterek megegyeznek a [8]-ban lévőkkel, ahol egy hasonló modellt tárgyalnak felhasználók általi újrAhívások nélkül. Az eredmények összhangban vannak [8] exponenciális esetbeli eredményeivel.

A 20. ábrán az új hívások blokkolásának és az átadottak megszakadásának valószínűsége, valamint a GoS (Grade of Service) figyelhető meg a cellahatárokon áthaladó felhasználók számára fenntartott csatornák számának függvényében. Látható, hogy kis számú csatorna fenntartása jelentősen javíthatja a szolgáltatás minőségi mértékét, de több ilyen csatorna esetén csak nagyon kis átadott hívás-megszakadási valószínűség-csökkenés érhető el jelentős blokkolási valószínűség-növekedés árán, és a GoS romlik.

10. Az eredmények alkalmazása

Az első rész modelljei alkalmazhatók bonyolultabb, nem csak exponenciális eloszlásokat tartalmazó szimulációs modellek ellenőrzésére.

A második részben lévő modell használható egy valós celluláris mobil hálózat teljesítményelemzésére. Emellett ez a munka tekinthető az első lépésnek is a bonyolultabb, harmadik generációs rendszereknek a MOSEL-2 eszközzel történő vizsgálata felé.

Hivatkozások

- [1] **Alfa A. S. and Li W.** PCS networks with correlated arrival process and retrial phenomenon, *IEEE Transactions on Wireless Communications*, Vol. 1 (2002) 630-637.
- [2] **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.
- [3] **Artalejo J.R.** New results in retrial queueing systems with breakdown of the servers, *Statistica Neerlandica*, Vol. 48 (1994) 23-36.
- [4] **Artalejo J.R.** Accessible bibliography on retrial queues, *Math. Comput. Modeling*, Vol. 30 (1999) 1-6.
- [5] **Begain K., Bolch G. and Herold H.** *Practical performance modeling, application of the MOSEL language*, Kluwer Academic Publisher, Boston, 2001.
- [6] **Begain K., Barner J., Bolch G., Zreikat A. I.** The Performance and Reliability Modeling Language MOSEL and its Application, *International Journal of Simulation*, Vol. 3 (2003) 66-80.
- [7] **Beutel B.** Integration of the Petri Net tool TimeNET into the MOSEL modelling environment, *MS Thesis*, Dept. of Computer Science, Univ. of Erlangen, Germany, 2003.
- [8] **Dharmaraja S., Trivedi K.S., Logothetis D.** Performance modeling of wireless networks with generally distributed handoff interarrival times, *Computer Communications*, Vol. 26 (2003) 1747-1755.

- [9] **Falin G.I.** A survey of retrial queues, *Queueing Systems*, Vol. 7 (1990) 127-168.
- [10] **Falin G.I. and Artalejo J.R.** A finite source retrial queue, *European Journal of Operational Research*, Vol. 108 (1998) 409-424.
- [11] **Falin G.I. and Templeton J.G.C.** *Retrial queues*, Chapman and Hall, London, 1997.
- [12] **Kulkarni V. G. and Choi B. D.** Retrial queues with server subject to breakdowns and repairs, *Queueing Systems Theory and Applications*, Vol. 7 (1990) 191-208.
- [13] **Sztrik J., Almási B., Roszik J.** Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Journal of Mathematical Sciences*, Vol. 132 (2006) 677-685.
- [14] **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.
- [15] **Kovalenko I.N., Kuznetsov N.Yu. and Pegg P.A.** *Mathematical theory of reliability of time dependent systems with practical applications*, John Wiley and Sons, Chichester, 1997.
- [16] **Marsan M. A., Carolis G. D., Leonardi E., Cigno R. L., Meo M.** Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials, *IEEE Journal on Selected Areas in Communications*, Vol. 19 (2001) 332-346.
- [17] **Ravichandran N.** *Stochastic methods in reliability theory*, John Wiley and Sons, New York, 1990.
- [18] **Sztrik J. and Gál T.** A recursive solution of a queueing model for a multi-terminal system subject to breakdowns, *Performance Evaluation*, Vol. 11 (1990) 1-7.
- [19] **Takagi H.** *Queueing Analysis, A Foundation of Performance Evaluation, Vol. 2., Finite Systems*, North-Holland, Amsterdam, 1993.
- [20] **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.
- [21] **Trivedi K. S.** *Probability and statistics with reliability, queueing and computer science applications*, Prentice-Hall, Englewood Cliffs, 1982.
- [22] **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.

A szerző publikációi

Nemzetközi folyóiratcikkek

- [J1] **Roszik J.** Homogeneous finite-source retrial queues with server and sources subject to breakdowns and repairs, *Annales Univ. Sci. Budapest., Sect. Comp.*, Vol. 23 (2004) 213-227.

- [J2] **Roszik J., Sztrik J., Kim C.S.** Retrial queues in the performance modeling of cellular mobile networks using MOSEL, *International Journal of Simulation: Systems, Science & Technology*, Vol. 6 (2005) 38-46.
- [J3] **Almási B., Roszik J., Sztrik J.** Homogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Mathematical and Computer Modelling*, Vol. 42 (2005) 673-682.
- [J4] **Sztrik J., Almási B., Roszik J.** Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs, *Journal of Mathematical Sciences*, Vol. 132 (2006) 677-685.
- [J5] **Roszik J., Sztrik J.** Performance analysis of finite-source retrial queues with non-reliable heterogeneous servers, *Journal of Mathematical Sciences* (publikálásra benyújtva)
- [J6] **Sztrik J., Roszik J.** Finite-source retrial queueing systems with heterogeneous non-reliable servers and different service policies, *Journal of Mathematical Sciences* (publikálásra benyújtva)
- [J7] **Roszik J., Sztrik J., Virtamo J.** Performance analysis of finite-source retrial queues operating in random environments, *International Journal of Operations Research* (publikálásra elfogadva)

Nemzetközi konferencia-cikkek

- [C1] **Roszik J., Almási B., Sztrik J.** Multiserver retrial queues with finite number of heterogeneous sources, *Proceedings of 6th International Conference on Applied Informatics*, Eger, Hungary, (2004) Vol. II, 19-26.
- [C2] **Roszik J., Sztrik J.** The effect of server's breakdown on the performance of finite-source retrial queueing systems, *Proceedings of 6th International Conference on Applied Informatics*, Eger, Hungary, (2004) Vol. II, 221-229.
- [C3] **Hyytiä E., Lassila P., Penttinen A., Roszik J.** Traffic load in dense wireless multihop network, *Proceedings of The 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, Montreal, Quebec, Canada, (2005) 9-17.

Magyar nyelvű publikációk

- [H1] **Roszik J.** Visszatérési sorbanállási rendszerek a telekommunikációs hálózatok modellezésében, *Informatika a Felsőoktatásban 2005*, Debrecen, (2005) 6 oldal
- [H2] **Roszik J.** Számítógép-hálózatok gyakorlati segédanyag, *mobiDIÁK könyvtár*, Debrecen, (2005) 43 oldal