

A queueing model to study the effect of network service breakdown in a CogInfoCom system

Attila Kuki*, Tamás Bérczes*, Béla Almási*, János Sztrik

*Faculty of Informatics, University of Debrecen, Debrecen, Hungary

E-mail: {kuki.attila,berczes.tamas,almasi.bela,sztrik.janos}@inf.unideb.hu

Abstract—In this paper we introduce a finite source retrial queueing model to investigate the performance measures of a cognitive infocommunication system subject to random network communication breakdowns. The sources of the queueing model represent the communication entities, which are divided into two classes: The first class is the class of "Intelligent sources", where the source entity is able to get information about the state of the network environment, and so they are able to retry the started request in the case of special communication problems. The second class is the "Normal sources" (e.g. sensors), where the source entity starts the network communication, but is not able to sense the state of the network environment, so it is unable to retry the transmission in the case of special problems or errors.

A novel Markovian model is constructed focusing on the question how the breakdowns of the communication network ("Network Cloud") influence the system's performance assuming that the service can be in 3 different states, fully operational, limited operation, broken offering different communication capabilities to the "Sources".

The main interest of the present paper is to investigate the main steady-state performance measures of the system. To achieve this goal the MOSEL (MODELing, Specification and Evaluation Language) tool is used to formulate and solve the problem. At the end of the paper several sample numerical examples will be shown illustrating the effect of the server's failure rate, for example on the mean number of customers in the system, mean orbit size, mean response and waiting time of the requests.

I. INTRODUCTION

In this paper we investigate how the network connection service problems can affect the performance of a complex cognitive infocommunication system. The "Network" (or "Network Cloud") is considered as an entity which offers services (infocommunication services) to the participants. The system built on the communication services can be more or less sensitive to the problems or failures that may appear in the Network.

As an example of a cognitive infocommunication system a telesurgical environment is described in [4], [10], see Fig. 1. In this example two medical doctors ("telesurgeons") located on different (and probably remote) places perform a medical operation by using a telesurgical robot. In the background different information is transmitted: The camera view of the robot's environment, the temperature, blood pressure, heartbeat rate of the patient is transmitted to the telesurgeons (inter-cognitive communication). Similarly, the talks of the doctors is transmitted also by using the Network services (intra-cognitive communication).

The sensibility of the system to the communication failures or problems can be highly different: For example if doctor "A"

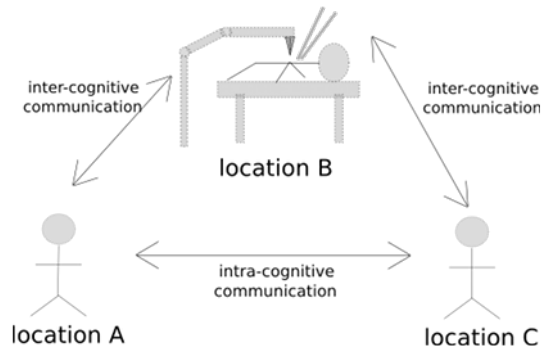


Fig. 1. A CogInfoCom Scenario

tells something, but the voice will not understandable at doctor "B", then doctor "A" will repeat the text and the operation may continue. An opposite example can be considered, for example if the camera view of the robot's environment is lost, then the operation must be stopped, or even it may cause unpredictable bad results.

In this paper a new mathematical queueing model is established to investigate how the different network problems can affect the performance of the system. The "Network" is considered as a service element in the queueing model, and will be referred as "Server", although in this case it does not mean a high performance computer as it is usual in the everyday life. The Server (communication network service) receives tasks from the sources but the common transmission buffer has finite capacity. The sources are classified into two classes: The "Intelligent" class contains humans or equipments, which may have information on the network's state, and so they can retry the transmission in some special cases (e.g. if the information was lost due to the full state of a network transmission buffer). The "Normal" class contains bare equipments (e.g. simple sensors), which transmit information to the network without having any knowledge of the state of the communication service. For this class in case the network transmission buffer is full, the sensor cannot send the sensed or measured data to the Server but is rejected and sent back to the Source.

The main interest of the present paper is to study the effect of the communication network breakdowns to the system performance. To model the network service level 3 different states are introduced. Each state of the Server represent the service capabilities of the communication network infrastructure. State 1,2,3 denotes "fully operational", "partial (or limited) availability", "totally failed", respectively. It is assumed that

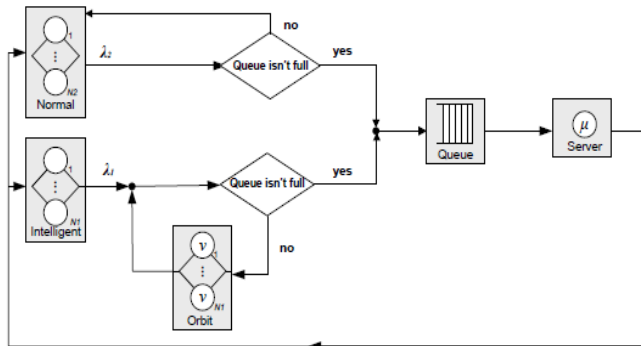


Fig. 2. A retrial queue with components

the single Server can fail either it is idle or busy. The operation times in level 1,2 are supposed to be exponentially distributed random variables with parameter δ . In case of a breakdown the communication is interrupted but the request stays at the service facility. However, during the breakdown only the service is stopped all other operations (request generations, retrials) are continued and after repair the service is resumed. The proposed new model is originated from complex retrial queueing systems treated in, for example [1], [2], [3], [6], [8], [9].

The rest of this paper is organized as follows. In Section 2 we present the corresponding queueing model. Numerical results and their discussion are provided in Section 3. Finally, Section 4 concludes the paper.

II. SYSTEM MODEL

Let's consider a special finite-source queueing model with a single Server (communication network service) having a common network transmission buffer having a limited capacity of B for the separated queues. The system contains two sets of finite sources. These sources generate requests (jobs) towards the Server. The first group of sources represents the "Intelligent" class, (humans or machines with AI capabilities, so they can respond to the changes of the communication environment, while the second one refers to the "Normal" group (e.g. sensors or some input devices).

The life cycles of the requests arriving from the "Normal" and the "Intelligent" classes are different. For the entities in "Intelligent" class and "Normal" class the request generation times (source times) are assumed to be exponentially distributed with parameter λ_1 and λ_2 , respectively. Entities arriving from the "Normal" class can enter into the buffer if it has free capacity, that is they simply joins the queue. In case of a full buffer (a queue with a maximum number of B tasks) the generated request cannot reach the queue, it will be rejected and returns to the source where a new request generation starts.

The entities of the "Intelligent" class can react to the changes in the servicing environment. In case of a full transmission buffer their requests will be retransmitted. An orbit is used for this feature in the underlying queueing system. When a request originated from the "Intelligent" class finds the queue to be full, it goes to the orbit. After an exponentially distributed time period (with parameter ν) the request retries joining the

queue. If it full, the request returns to the orbit again and start generating a new retrial.

The main goal of this paper to investigate the effect of the Server states and breakdowns to the performance characteristics of the system. We assume that the Server deterioration condition can be described by 3 levels. Level 1 denotes the best condition meaning that the system is as good as a new, and the service rate is the highest. Level 2 denotes a less effective service and finally in level 3 the service and the system break down, that is the service stops.

The network communication service times are exponentially distributed with parameters $\mu_1 > \mu_2$. The operating condition of the Server may switch from a specific level to any higher level, or in case of a repair, to level 1. The operation times at level 1,2 are assumed to be exponentially distributed with parameter δ . The transitions are controlled by transition probabilities. More precisely, in case of a failure the server's state changes from level 1 to level 2,3 with probability p and $1 - p$, respectively. It is clear that from level 2 it goes to level 3 and from level 3 it changes to level 1. After completing a service the request is selected from the Intelligent and Normal ones if there any in the queue with probability 0.5, that is we have a symmetric selection rule.

If the server breaks down in busy state, the interrupted request will stay at the service and resumed if the server is repaired. The repair times are assumed to exponentially distributed with parameters $\beta_i, i = 2, 3$. If the server is broken only service is interrupted but all the other operations are continued (new and repeated requests can be generated).

Thus, the server can be in three states:

- *fully operational*: The server level is 1. The service of a request is the most efficient in this case.
- *limited operation*: The server level is 2. The service of a request is less efficient than in the previous case.
- *totally failed, broken*: The server is broken, it can not start serving any arriving requests until it is repaired.

The functionality of this communication network is presented on Fig. 2.

The following notations are introduced for the queueing model (Table I contains the overview of parameters of the model of the CoginfoCom system):

- $k_1(t)$ is the number of working entities in the "Intelligent" class at time t ,
- $k_2(t)$ is the number of working entities in the "Normal" class at time t ,
- $q_1(t)$ denotes the number of requests in the queue for requests from "Intelligent" class at time t ,
- $q_2(t)$ denotes the number of requests in the queue for requests from "Normal" class at time t ,
- $o(t)$ is the number of waiting jobs in the orbit at time t ,
- $y(t)$ is the operation level of the Server. $y(t)=1$ if the Server is in fully operation case, $y(t)=2$ if the Server

is in limited operation case, and $y(t)=3$ if the Server is broken at time t ,

- $c(t) = 0$ if the server is idle, $c(t) = 1$ if the server is busy with a request coming from the "Intelligent" class, $c(t) = 2$ when the server is busy with a request coming from the "Normal" class.

It is ease to see that:

$$k_1(t) = \begin{cases} N1 - q_1(t) - o(t) - 1, & c(t) = 1 \\ N1 - q_1(t) - o(t), & c(t) = 0, 2 \end{cases}$$

and

$$k_2(t) = \begin{cases} N2 - q_2(t) - 1, & c(t) = 2 \\ N2 - q_2(t), & c(t) = 0, 1 \end{cases}$$

TABLE I. LIST OF NETWORK PARAMETERS

Parameter	Maximum	Value at t
Working "Intelligent" entities	$N1$ (population size)	$k_1(t)$
Working "Normal" entities	$N2$ (population size)	$k_2(t)$
"Intelligent" generation rate		λ_1
"Normal" generation rate		λ_2
Total gen. rate	$\lambda_1 N1 + \lambda_2 N2$	$\lambda_1 k_1(t) + \lambda_2 k_2(t)$
Requests in queue	B	$q(t) = q_1(t) + q_2(t)$
Service rates		$\mu_i, i = 1, 2$
Repair rates		$\beta_i, i = 2, 3$
Failure rate		δ
Cust. in service area	$B + 1$	$q(t) + 1$
Requests in Orbit	$N1$ (orbit size)	$o(t)$
Retrial rate		v

To create a Markovian model, the distributions of inter-event times (i.e., request generation times, operation times, service times, retrial times, repair times) presented in the network are assumed to be exponentially distributed and totally independent of each other. The state of the communication network at a time t corresponds to a Continuous Time Markov Chain (CTMC) with 5 dimensions, namely:

$$X(t) = (y(t); c(t); q_1(t); q_2(t); o(t))$$

Let us denote the steady-state distribution by

$$P(y, c, q_1, q_2, o) = \lim_{t \rightarrow \infty} P(y(t) = y; c(t) = c; q_1(t) = q_1; q_2(t) = q_2; o(t) = o)$$

Note, that the state space of this Continuous Time Markovian Chain is finite, so the steady-state probabilities surely exist. Because of the fact, that the state space of this chain is very large, it is rather difficult to calculate the system measures in the traditional way of writing down and solving the underlying steady-state equations. To simplify this procedure we used the software tool MOSEL (Modeling, Specification and Evaluation Language), see Begain et al. [5], to formulate the model and to obtain the performance measures as it has been done in, for example [7], [9].

As soon as we have calculated the distributions defined above, the most important steady-state system characteristics can be obtained in the following way:

- *Utilization of the server*

$$U_S = \sum_{y=1}^2 \sum_{c=1}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, c, q_1, q_2, o)$$

- *Availability of the server*

$$A_S = \sum_{y=1}^2 \sum_{c=0}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, c, q_1, q_2, o)$$

- *Average number of request in the orbit*

$$\begin{aligned} \bar{O} &= E(o(t)) = \\ &= \sum_{y=1}^3 \sum_{c=0}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} o P(y, c, q_1, q_2, o) \end{aligned}$$

- *Average number of request in the queue*

$$\begin{aligned} \bar{Q} &= E(q_1(t) + q_2(t)) = \\ &= \sum_{y=1}^3 \sum_{c=0}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} q_1 P(y, c, q_1, q_2, o) \\ &+ \sum_{y=1}^3 \sum_{c=0}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} q_2 P(y, c, q_1, q_2, o) \end{aligned}$$

- *Average number of request in the network*

$$\begin{aligned} \bar{M} &= \bar{O} + \bar{Q} + \\ &+ \sum_{y=1}^2 \sum_{c=1}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, c, q_1, q_2, o) \end{aligned}$$

- *Average number of active Intelligent entities*

$$\begin{aligned} \bar{\Lambda}_1 &= N1 - E(q_1(t)) - \sum_{y=1}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, 1, q_1, q_2, o) \\ &= N1 - \sum_{y=1}^3 \sum_{c=0}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} q_1 P(y, c, q_1, q_2, o) - \\ &- \sum_{y=1}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, 1, q_1, q_2, o) \end{aligned}$$

- *Average number of Intelligent entities in the network*

$$\bar{N1} = N1 - \bar{\Lambda}_1$$

- *Average number of active Normal entities*

$$\begin{aligned} \bar{\Lambda}_2 &= N2 - E(q_2(t)) - \sum_{y=1}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, 2, q_1, q_2, o) \\ &= N2 - \sum_{y=1}^3 \sum_{c=0}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} q_2 P(y, c, q_1, q_2, o) - \\ &- \sum_{y=1}^2 \sum_{q_1=0}^B \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(y, 2, q_1, q_2, o) \end{aligned}$$

- *Average number of Normal entities in the network*

$$\bar{N2} = N2 - \bar{\Lambda}_2$$

- *Average generation rate of Intelligent entities*

$$\bar{\lambda}_1 = \lambda_1 \bar{\Lambda}_1$$

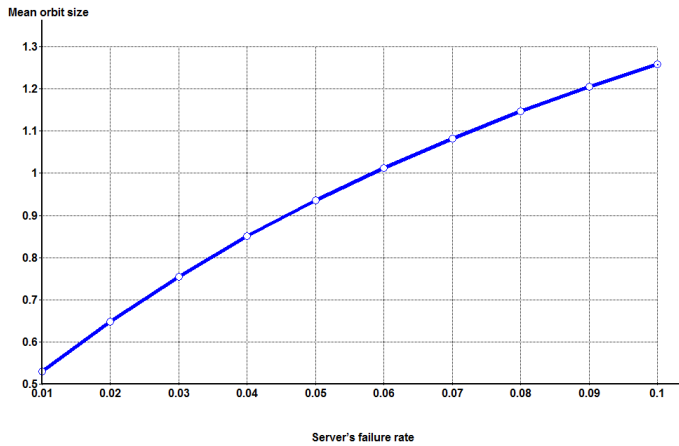


Fig. 3. Mean orbit size vs Server's failure rate

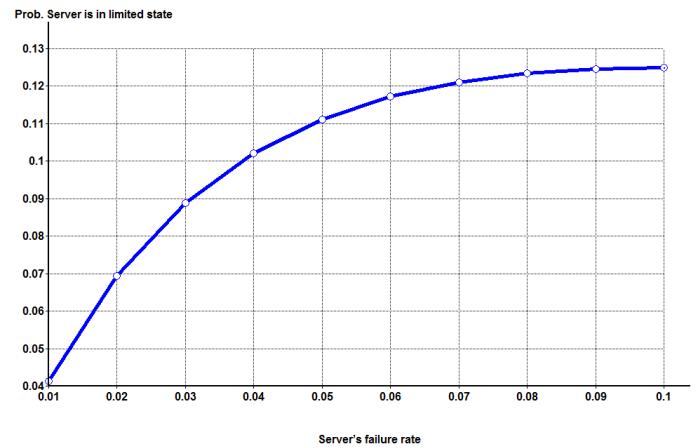


Fig. 4. Probability that the server is in limited state vs Server's failure rate

- Mean response time for Intelligent entities:

$$\overline{T}_1 = \frac{\overline{N}_1}{\overline{\lambda}_1}$$

III. NUMERICAL RESULTS

To demonstrate the effect of the Server's breakdown in a cognitive information system some computational results are presented in this chapter. The corresponding parameters can be overviewed in Table II. Numerous interactions of parameters were investigated by using the model. The most interesting results are displayed on the following figures.

On Figure 3 one can see the effect of increasing value of failure rate for the size of the orbit. It can be observed, how the Server's breakdown fills up the orbit.

Figure 4 displays the limited operation mode of the Server, i.e. the Server is in state 2. The probability of this state is computed as a function of the failure rate.

Figure 5 deals with the broken state of the Server. In this state the service is interrupted but all other operations are still performed. As on the previous Figure, the probability of this state is computed.

On Figure 6 the effect of Server's breakdown with mean response time of intelligent entities is displayed.

Figure 7 displays the wasted time of the intelligent entities in function of server's failure rate. The entities can waste time in the case, when the server become un-operable while the server is busy. The wasted time could be calculated in the following way:

$$\overline{T}_1 - \overline{W}_1 = \frac{1}{\mu_1}$$

It can be observed, how the Server's breakdown grease the wasted time.

IV. CONCLUSION

A cognitive communication system was considered. A model was built with two classes of entities. The "Intelligent"

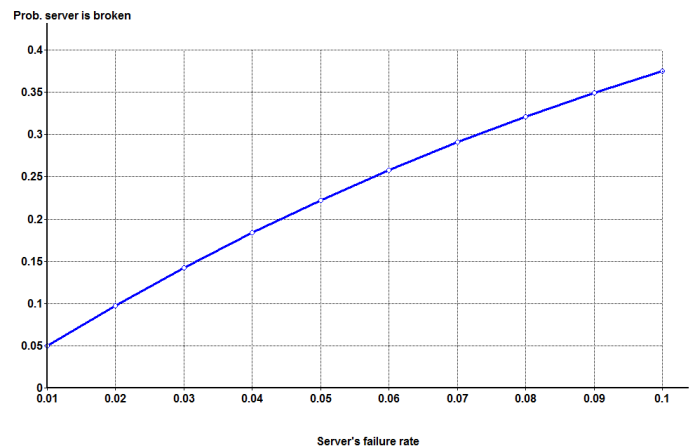


Fig. 5. Probability that the server is failed vs Server's failure rate

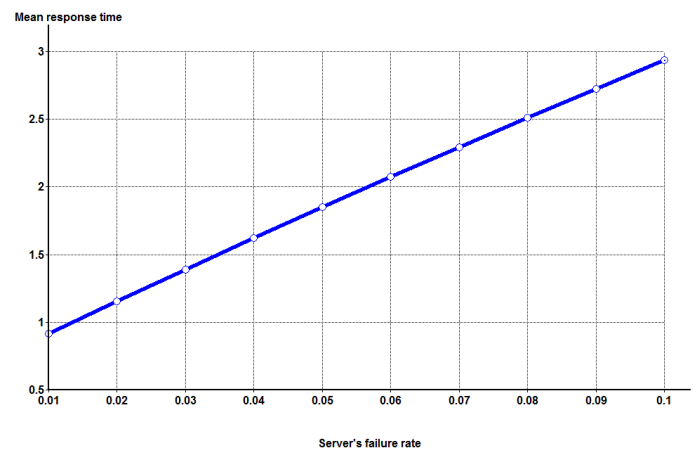


Fig. 6. Mean response time of intelligent entities vs Server's failure rate

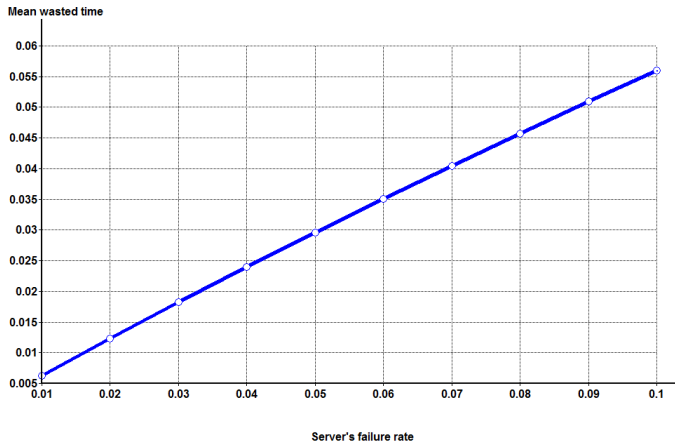


Fig. 7. Wasted time of intelligent entities vs Server's failure rate

TABLE II. NUMERICAL VALUES OF MODEL PARAMETERS

Parameter	Symbol	Value
Overall generation rate	λ	1.5
Intelligent generation rate	$\lambda_1 = \frac{2}{10}\lambda$	0.3
Normal generation rate	$\lambda_2 = \frac{8}{10}\lambda$	1.2
Number of Intelligent entities	$N1$	3
Number of Normal entities	$N2$	50
Retrial rate	ν	4
Service rate	μ_1	20
Service rate in limited state	μ_2	5
Server's failure rate	δ	[0.01..0.1]
Server's repair rate	$\beta_2 = \beta_3$	0.1
Prob. server's state changes from level 1 to 2	p	0.5
Buffer size	B	3

items could be aware of any changes of the network environment, while the "Normal" ones not. The Server is subject to breakdowns. Multiple deterioration levels of Server was introduced and investigated. The effect of Server's breakdowns was expressed with the performance characteristics, focused on the orbit size and the probabilities of the Server's states.

Further task could be to study the cases of the blocked requests (when the Server is failed, all other operations are stopped), or the case of the interrupted service which will continue after repair.

ACKNOWLEDGMENT

The publication was supported by the TÁMOP 4.2.2. C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund,

The work of János Sztrik was realized in the frames of TÁMOP 4.2.4. A/2-11-1-2012-0001 National Excellence Program - Elaborating and operating an inland student and researcher personal support system. The project was subsidized by the European Union and co-financed by the European Social Fund.

The work of Tamás Bérczes was realized in the frames of TÁMOP 4.2.4. A/2-11-1-2012-0001 National Excellence Program Elaborating and operating an inland student and researcher personal support system. The project was subsidized by the European Union and co-financed by the European Social Fund.

REFERENCES

- [1] B. Almási, J. Roszik, and J. Sztrik, "Homogeneous finite-source retrial queues with server subject to breakdowns and repairs," *Mathematical and Computer Modelling*, vol. 42, pp. 673-682, 2005.
- [2] B. Almási, T. Bérczes, A. Kuki, and J. Sztrik, "A contribution to modeling sensor communication networks by using finite-source queueing systems," *Proceedings of 8th IEEE International Symposium on Applied Computational Intelligence and Informatics*, pp. 89-93, 2013.
- [3] J. R. Artalejo, A. Gómez-Corral, *Retrial Queueing Systems: A Computational Approach*. Berlin: Springer, 2008.
- [4] P. Baranyi, and A. Csapó, "Definition and Synergies of Cognitive Infocommunications," *Acta Polytechnica Hungarica*, vol. 9, no. 1, pp. 67-83, 2012.
- [5] K. Begain, G. Bolch, and H. Herold, *Practical performance modeling, application of the MOSEL language*. Boston: Kluwer Academic Publisher, 2001.
- [6] T. Bérczes, A. Kuki, B. Almási, J. Sztrik and P. Moyal, "A new model of finite-source retrial queues with multi-state servers breakdown," *Pre-Proceedings of 9th International Conference on Applied Mathematics*, pp. 63-66, 2013.
- [7] G. Bolch, S. Greiner, H. de Meer, and K. Trivedi, *Queueing Networks and Markov Chains*, 2nd ed. New York: John Wiley & Sons, 2006.
- [8] T. V. Do, N. H. Do, and J. Zhang, "An enhanced algorithm to solve multiserver retrial queueing systems with impatient customers," *Computers & Industrial Engineering*, vol. 65, issue 4, pp. 719-728, 2013.
- [9] P. Wuechner, J. Sztrik, and H. de Meer, "Modeling wireless sensor networks using finite-source retrial queues with unreliable orbit," *Springer Lecture Notes in Computer Science*, vol. 6821, pp. 275-285, 2011.
- [10] "CogInfoCom - Cognitive Infocommunications," official website, <http://www.coginfocom.hu/>, downloaded 09.09.2013

