Investigating the effect of network service breakdown in multilayered cognitive communication 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 a finite source retrial queueing model is considered to study the performance measures of a communication network subject to random system breakdowns. The sources of the queueing model represent the communication entities, which are divided into two groups: The first group of sources is the class of "Intelligent items", where the items are able to sense and get information about the condition of the network environment, and so they are able to retry the started processing request in the case of special network communication issues. The second class is the "Dummy sources" (e.g. automated equipements), where the source item starts the network communication, but is not able to sense the changes in the network conditions, so it is unable to retry the transmission in the case of any errors.**

A novel Markovian model is constructed focusing on the question how the breakdowns of the communication network influence the system's performance assuming that the service are divided into 3 successive steps, ie. the Service Unit has 3 servers representing the functionality of different network layers. At a time only one request can be under processing.

The main interest of the present paper is to investigate the main steady-state performance measures of the system in case of different distributions of service times (Erlang, hipo-exponential, and exponential). To obtain the results the MOSEL (MOdeling, Specification and Evaluation Language) tool is used to formulate and solve the model. 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 entities in the queue, mean orbit size, mean response time for "Intelligent" items.

I. Introduction

The performance evaluation of the network services is a hot research area today. The queueing modeling based investigation of different effects on the network performance is widely used technology in this field. In [1] the authors established a finite source queueing model in order to investigate the effect of network service breakdowns on the performance of a complex cognitive infocommunication system (where both intra-cognitive and inter-cognitive communication may appear). Definitions, synergies and applications of CogInfoCom are described e.g. [2], [3], [4] . The sources are classified into two class: The first class contains elements (typically human resources), which are able to "intelligently" sense the network state (i.e. they are able to repeat their request, if a transmission problem occurred). The second class contains "dummy" sources, which are not able to sense the network state (e.g. temperature sensor nodes, which regularly transmit the measurement result, but if a transmission error occurs the data is lost (the next measurement transmission will complement the measurement information).

The network service was modeled as a single queueing service station in [1]. The network service problems (or network breakdowns) were included into the model by specifying three service states: fully operational, partially down and not operational.

The numerical results confirmed, that the network breakdowns plays an important role and have considerable influence on the system performance.

In this paper we modify the model presented in [1] in order to produce more general model and also to produce more precise model which can be more fitting for the real life systems. The typical applications of the presented model is closer to inter-cognitive communications than intra-cognitive ones. This multilayered system modeled with multiple service units fits better to the sensor-bridging and representation-bridging types of communication (see, [2]), because there are successive steps (e.g. transfer, filter, and adapt) during a communication request between the endpoints. Our model is a finite-source queueing model, where the sources can be sensors, AI entities, human beings, etc. Almost all of the synergistic fields (described in [2]) can be modeled with the presented system, but due to the finite number of sources the closest applications are the cognitive informatics and the body area networks (BAN's).

The currently widely used Internet network architecture contains three layers below the Application Layer: Transport, Internetwork and Link (or Network Access) Layer can be found in the architecture (see [5], [6], [7]). The layered network architecture can be modeled more precisely if we divide the queueing network service into three stages (representing the three layers below the application). The request must pass all the three service stages for the complete transmission. We use three servers in this paper, which are connected to each other in a strictly ordered sequenced manner: the first server gets the request from the application, performs its task and passes the request to the second server. The second server works on the request, and transmits the job to the third server when finished. The complete request service process will be finished after the third server has done the work.

According to the detailed mathematical specification (see

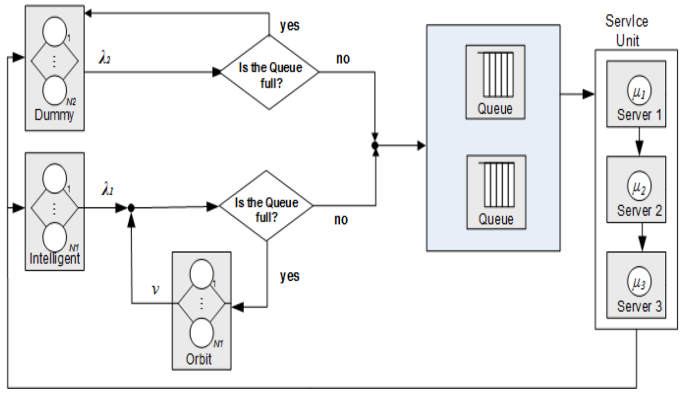

Fig. 1. A retrial queue with components

Section 2), the network service will be described by the sum of three exponential random variables, so giving a generalization to the model of [1] and [9].

The new layered structure has some influence on modeling the breakdown too: if the request is at the second service station in the queueing model, then the breakdown of the first server has no effect on the request service. The failure of the second server will stop the job service, and the request will be lost (following the rule introduced in [1] and [8]). If the network breakdown occurs at the third queueing service station, the the second server must wait for the repair of it in order to pass the request.

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

For modeling the task described above let's introduce a three-server queueing system (Service Unit - each server corresponds to the service in the specific network layer) with a finite number of sources The Service Unit has a common network transmission buffer (a FIFO queue) having a capacity constraint of B for the separated queues. Two groups of finite sources are considered in the model. The communication requests (jobs) are generated by the items of these sources towards the Service Unit. The first group of sources stands for the "Intelligent" class, (humans or machines with Artifical Intelligent features, so they can be aware of the modifications of the network functionality, while the other one refers to the "Dummy" group (e.g. temperature sensors or some input devices).

The entity life history of the jobs generated from the "Dummy" and the "Intelligent" groups are not the same. For the requests generated from the "Intelligent" class and "Dummy" class the job generation times (source times) are assumed to be exponentially distributed with parameter λ_1 and λ_2, respectively. Requests generated from the "Dummy" class can enter into the FIFO-queue (buffer) if it has free capacity, that is they simply joins the queue. In case of a full FIFO-queue (a buffer with a maximum number of B tasks) the generated request is not able to reach the queue, it will be rejected and returns to the source where a new request generation starts.

The requests of the "Intelligent" class can sense the changes in the communication (servicing) environment. In case of a full transmission FIFO-queue their requests will be retransmitted. This functionality is solved by the help of an orbit in the underlying queueing model. When a job generated from the "Intelligent" class finds the FIFO-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 is full, the job returns to the orbit again and start generating a new retrial.

This paper focuses to study the effect of the breakdowns of the Server Unit to the performance measures of the communication network. In the model of [1] the Server deterioration condition was described by 3 levels: level 1 - best condition, all features are working; level 2 - limited condition, some features are not working or the speed is decreased; level 3 - failed, the system is down. In this paper the deterioration conditions of the system are mapped into the Up and Down states of the three servers (in the Service Unit).

The Service Unit contains three servers denoted by Server 1, Server 2, and Server 3. The service times are independent and exponentially distributed with parameters μ_1, μ_2, and μ_3, respectively. That means, that the exponential distribution of the overall service time is generalized to Hipo-exponential or Erlang (in case of μ_1=μ_2=μ_3) distribution.

At a given time the Service Unit can process at most one request. Server 1 will not accept any new incoming job until Server 2 and then Server 3 finishes serving the request in the Service Unit. After completing a service a new request is selected from the "Intelligent" and "Dummy" queues if there are any in the buffer with probability 0.5, that is we have a symmetric selection rule.

Each server in Service Unit is subject to breakdown. The time periods between breakdowns of the servers are independent and exponentially distributed. The repair times are also assumed to be exponentially distributed with parameters $\beta_i, i = 1,2,3$. In the modeling process the server breakdowns are handled as follows:

- Server 1 breaks down. In case of busy Server 1 the request will stay at the service and resumed if the server is repaired. In case of idle Server 1, the currently running service continues at Server 2 and Server 3, if there are any request in the Service Unit.
- Server 2 breaks down. In case of busy Server 1 or Server 2, the request will stay at the service and resumed if the server is repaired. In case of busy Server 3, the service of the request continues.
- Server 3 breaks down. In case of idle Server 3, the service of the request at Server 1 (if any) will continue. In case of busy Server 3, the request will stay at the service and resumed if the server is repaired.

The functionality of this communication network is presented on Fig. 1.

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

- $k_1(t)$ is the number of working items in the "Intelligent" class at time t,
- $k_2(t)$ is the number of working items in the "Dummy" 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 "Dummy" class at time t,
- $o(t)$ is the number of waiting jobs in the orbit at time t,
- $c_1(t)$ denotes the state of the Server 1. $c_1(t) = 0$, if the Server 1 is idle, $c_1(t) = 1 or 2$ if the Server 1 is busy with a request coming from the "Intelligent" or the "Dummy" class, respectively. $c_1(t) = 3$ when the Server 1 is down.
- $c_2(t)$ denotes the state of the Server 2. $c_2(t) = 0$, if the Server 2 is idle, $c_2(t) = 1 or 2$ if the Server 2 is busy with a request coming from the "Intelligent" or the "Dummy" class, respectively. $c_2(t) = 3$ when the Server 2 is down.
- $c_3(t)$ denotes the state of the Server 3. $c_3(t) = 0$, if the Server 3 is idle, $c_3(t) = 1 or 2$ if the Server 3 is busy with a request coming from the "Intelligent" or the "Dummy" class, respectively. $c_3(t) = 3$ when the Server 3 is down.

It is ease to see that:

$$k_1(t) = \begin{cases} N_1 - q_1(t) - o(t) - 1, & \exists i, c_i(t) = 1 \\ N_1 - q_1(t) - o(t), & \text{other cases} \end{cases}$$

and

$$k_2(t) = \begin{cases} N_2 - q_2(t) - 1, & \exists i, c_i(t) = 2 \\ N_2 - q_2(t), & \text{other cases} \end{cases}$$

TABLE I
LIST OF NETWORK PARAMETERS

Parameter	Maximum	Value at t
Working "Intelligent" entities	N_1 (population size)	$k_1(t)$
Working "Normal" entities	N_2 (population size)	$k_2(t)$
"Intelligent" generation rate		λ_1
"Normal" generation rate		λ_2
Total gen. rate	$\lambda_1 N_1 + \lambda_2 N_2$	$\lambda_1 k_1(t) + \lambda_2 k_2(t)$
Requests in queue	B	$q(t) = q_1(t) + q_2(t)$
Service rates		$\mu_i, i = 1,2,3$
Repair rates		$\beta_i, i = 1,2,3$
Failure rate		$\delta_i, i = 1,2,3$
Cust. in service area	$B+1$	$q(t)+1$
Requests in Orbit	$N1$ (orbit size)	$o(t)$
Retrial rate		ν

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 model 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 6 dimensions, namely:

$$X(t) = (c_1(t); c_2(t); c_3(t); q_1(t); q_2(t); o(t))$$

Let us denote the steady-state distribution by

$$P(c_1, c_2, c_3, q_1, q_2, o) = \lim_{t \to \infty} P(c_1(t) = c_1;$$
$$c_2(t) = c_2; c_3(t) = c_3; q_1(t) = q_1; q_2(t) = q_2; o(t) = o)$$

Because of the number of states (state space) of the considered CTMC (Continuous Time Markovian Chain) is not infinite, there is no issue with the existence of the steady-state probabilities. Due to the fact, that the state space of this model is very large, it would be difficult to obtain the system measures in the usual way, that is writing down and solving the underlying steady-state equations. MOSEL-2 (Modeling, Specification and Evaluation Language) software package was used to simplify this procedure, to build up the model and to calculate the performance characteristics as it has been done in, for example [12], [14].

As soon as the steady-state probabilities described above have been calculated, the most important stationary system characteristics can be obtained in the following way:

- *Utilization of the Server 1*

$$U_{S_1} = \sum_{c_1=1}^{2} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(c_1, 0, 0, q_1, q_2, o)$$

- *Utilization of the Server 2*

$$U_{S_2} = \sum_{c_1=0}^{1} \sum_{c_2=1}^{2} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(3c_1, c_2, 0, q_1, q_2, o)$$

- *Utilization of the Server 3*

$$U_{S_3} = \sum_{c_1=0}^{1} \sum_{c_2=0}^{1} \sum_{c_3=1}^{2} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(3c_1, 3c_2, c_3, q_1, q_2, o)$$

- *Overall Utilization of the Service Unit*

$$U_S = \sum_{i=1}^{3} US_i$$

- *Availability of the Service Unit*

$$A_S = \sum_{c_1=0}^{2} \sum_{c_2=0}^{2} \sum_{c_3=0}^{2} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} P(c_1, c_2, c_3, q_1, q_2, o)$$

- *Mean number of request in the orbit*

$$\overline{O} = \sum_{c_1=0}^{3} \sum_{c_2=0}^{3} \sum_{c_3=0}^{3} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} oP(c_1, c_2, c_3, q_1, q_2, o)$$

- *Average number of request in the queue*

$$\overline{Q} = E(q_1(t)) + E(q_2(t)), where$$

$$E(q_1(t)) = \sum_{c_1=0}^{3} \sum_{c_2=0}^{3} \sum_{c_3=0}^{3} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} q_1 P(c_1, c_2, c_3, q_1, q_2, o),$$

$$E(q_2(t)) = \sum_{c_1=0}^{3} \sum_{c_2=0}^{3} \sum_{c_3=0}^{3} \sum_{q_1=0}^{B} \sum_{q_2=0}^{B-q_1} \sum_{o=0}^{N1-q_1} q_2 P(c_1, c_2, c_3, q_1, q_2, o)$$

- *Probability of serving an "Intelligent" source's request*

$$\overline{P_1} = \sum_{c_2=0}^{3}\sum_{c_3=0}^{3}\sum_{q_1=0}^{B}\sum_{q_2=0}^{B-q_1}\sum_{o=0}^{N1-q_1} P(1,c_2,c_3,q_1,q_2,o) + \sum_{c_1=0}^{3}\sum_{c_3=0}^{3}\sum_{q_1=0}^{B}\sum_{q_2=0}^{B-q_1}\sum_{o=0}^{N1-q_1} P(c_1,1,c_3,q_1,q_2,o) + \sum_{c_1=0}^{3}\sum_{c_2=0}^{3}\sum_{q_1=0}^{B}\sum_{q_2=0}^{B-q_1}\sum_{o=0}^{N1-q_1} P(c_1,c_2,1,q_1,q_2,o)$$

- *Probability of serving a "Dummy" source's request*

$$\overline{P_2} = \sum_{c_2=0}^{3}\sum_{c_3=0}^{3}\sum_{q_1=0}^{B}\sum_{q_2=0}^{B-q_1}\sum_{o=0}^{N1-q_1} P(2,c_2,c_3,q_1,q_2,o) + \sum_{c_1=0}^{3}\sum_{c_3=0}^{3}\sum_{q_1=0}^{B}\sum_{q_2=0}^{B-q_1}\sum_{o=0}^{N1-q_1} P(c_1,2,c_3,q_1,q_2,o) + \sum_{c_1=0}^{3}\sum_{c_2=0}^{3}\sum_{q_1=0}^{B}\sum_{q_2=0}^{B-q_1}\sum_{o=0}^{N1-q_1} P(c_1,c_2,2,q_1,q_2,o)$$

- *Average number of request in the network*

$$\overline{M} = \overline{O} + \overline{Q} + \overline{P_1} + \overline{P_2}$$

- *Average number of active "Intelligent" entities*

$$\overline{\Lambda_1} = N1 - E(q_1(t)) - \overline{P_1}$$

- *Average number of "Intelligent" entities in the network*

$$\overline{N1} = N1 - \overline{\Lambda_1}$$

- *Average number of active "Dummy" entities*

$$\overline{\Lambda_2} = N2 - E(q_2(t)) - \overline{P_2}$$

- *Average number of "Dummy" entities in the network*

$$\overline{N2} = N2 - \overline{\Lambda_2}$$

- *Average generation rate of "Intelligent" entities*

$$\overline{\lambda_1} = \lambda_1 \overline{\Lambda_1}$$

- *Mean response time for "Intelligent" entities:*

$$\overline{T_1} = \frac{\overline{N1}}{\overline{\lambda_1}}$$

III. Numerical results

To illustrate the effect of the breakdowns of the Service Unit in a communication information system some numerical results are given in this chapter. The parameters of the numerical model can be overviewed in Table II. The values of the parameters are based on the values used in [1]. In figures four different lines (sometimes they are coincided) are presented. The original $1/20$ mean service time is divided different ways between the servers of the Service Unit.

- The service time is divided into three equal parts. This case represents the Erlang distribution (Erlang - blue line dotted with rhombuses).

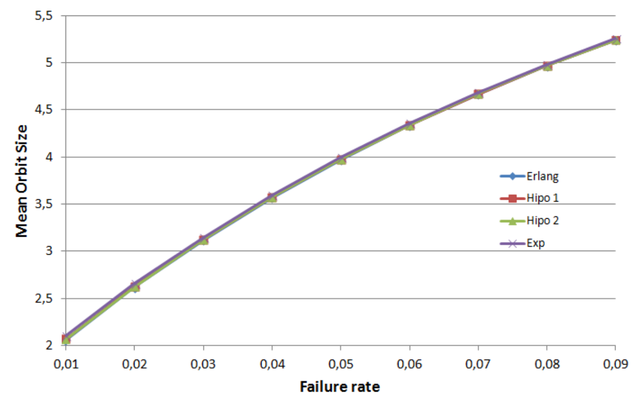

Fig. 2. Mean orbit size vs Server's failure rate

- The service times are not equal, they are smaller at Server 1, and greater at Server 3. This case represents the Hipo-exponential distribution (Hipo1 - red line dotted with squares).
- The service times are not equal, they are greater at Server 1, and smaller at Server 3. This case represents also the Hipo-exponential distribution (Hipo2 - green line dotted with triangles).
- The service time at Server 1 is the original value, and the service times at Server 2 and Server 3 are set to a very large value. This case represents approximately the exponential distribution (Exp - purple line dotted with x-es).

On Figure 2 the effect of increasing value of failure rate are displayed for the size of the orbit. It can be observed, that the distribution of the service times in the Service Unit has almost no effect for this system characteristic.

Figure 3 shows the average length of the overall (joint "Intelligent" and "Dummy") queue. The generalized exponential distributions (Erlang, Hipo1, Hipo2) have the same characteristics for this performance measure. The effect of the approximated exponential distribution differs slightly from the others. On Figure 4 the effect of the breakdown of the Service Unit with average response time of "Intelligent" items is represented. As in Figure 2, it can also be observed, that the distribution of the service times in the Service Unit has almost no effect for this performance measure. Figure 5 displays the wasted time of the intelligent entities due to the fact of the breakdowns of the Service unit. At first look there is a large difference caused by different distributions of the service times, but investigating the values of axes Y it turns out, that those differences are not significant.

IV. Conclusion

A communication system with the functionality of multiple network layers was considered. A model was built with two classes of sources. The "Intelligent" items could be aware of any changes of the network environment, while the "Dummy" ones not. The Service Unit contains three servers, each represent a network layer. The service is subject to breakdowns.

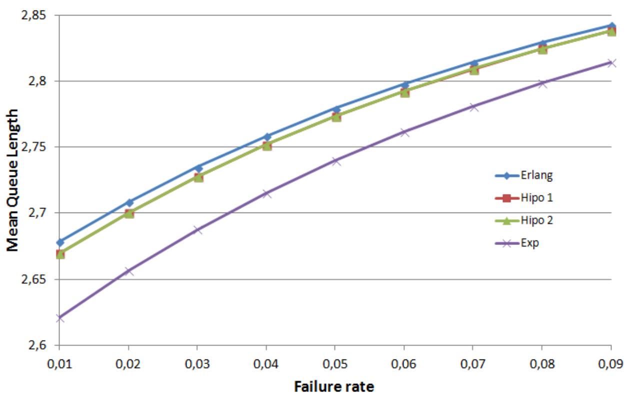

Fig. 3. Mean queue length vs Server's failure rate

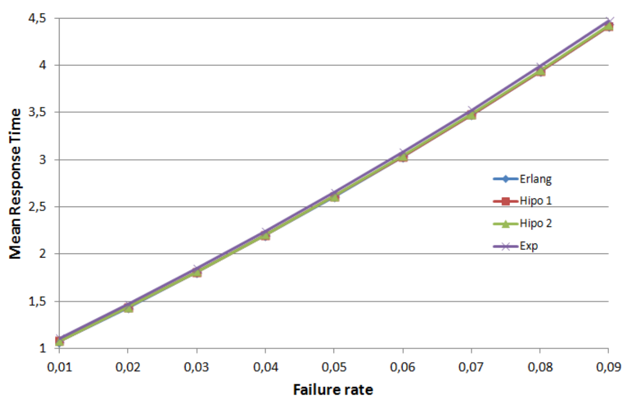

Fig. 4. Mean response time of intelligent entities vs Server's failure rate

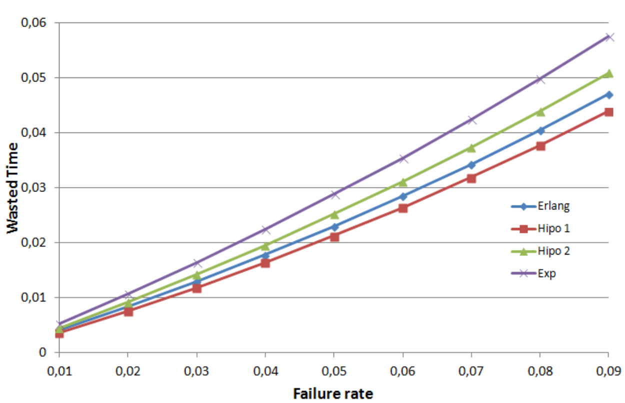

Fig. 5. Wasted time of intelligent entities vs Server's failure rate

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
"Dummy" generation rate	$\lambda_2 = \frac{8}{10}\lambda$	1.2
Number of "Intelligent" entities	$N1$	3
Number of "Dummy" entities	$N2$	50
Retrial rate	ν	4
Overall service rate	μ	20
Overall server's failure rate	δ	[0.01..0.1]
Server's repair rate	β_i	0.1
Buffer size	B	3

Different distributions of the service time was introduced and investigated. The effect of breakdowns of the Service Unit was expressed with the performance measures, focused on the distribution of the service times of the servers in the Service Units. It can be concluded, that the distribution under consideration has almost no effect to the system characteristics, thus using the exponential distribution will not damage the generalization.

ACKNOWLEDGMENT

The work 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.

REFERENCES

[1] A. Kuki, T. Bérczes, B. Almási, and J. Sztrik, "A queueing model to study the effect of network service breakdown in a CogInfoCom system ," *IEEE 4th International Conference on Cognitive Infocommunications: CogInfoCom 2013*, pp. 205-210, 2013.
[2] P. Baranyi, A. Csapó, "Definition and Synergies of Cognitive Infocommunications," *Acta Polytechnica Hungarica*, vol. 9, pp. 67-83, 2012.
[3] P. Baranyi, A. Csapó, P. Várlaki, "An Overview of Research Trends in CogInfoCom," *IEEE International Conference on Intelligent Engineering Systems, Tihany, Hungary*, pp. 181-186, 2014.
[4] G. Sallai, "Chapters of Future Internet Research," *4th IEEE International Conference on Cognitive Infocommunications, Budapest, Hungary*, pp. 161-165, 2013.
[5] J. Postel, "Transmission Control Protocol," *IETF RFC 793.*, 1981.
[6] J. Postel, "Internet Protocol (IP)," *IETF RFC 791.*, 1981.
[7] R. Branden, "Requirements for Internet Hosts – Communication Layers," *IETF RFC 1122.*, 1989.
[8] 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.
[9] 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.
[10] J. R. Artalejo, A. Gómez-Corral, *Retrial Queueing Systems: A Computational Approach*. Berlin: Springer, 2008.
[11] 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.
[12] G. Bolch, S. Greiner, H. de Meer, and K. Trivedi, *Queueing Networks and Markov Chains*, 2nd ed. New York: John Wiley & Sons, 2006.
[13] 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.
[14] 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.