The impact of servers breakdown on the performance of proxy cache servers *

Tamás Bérczes, János Sztrik, Attila Házy
Faculty of Informatics, University of Debrecen
Debrecen, Hungary, {berczes.tamas,sztrik.janos}@inf.unideb.hu
Department of Applied Mathematics, University of Miskolc
Miskolc, Hungary, matha@uni-miskolc.hu

Abstract

An open Jackson-type queuing network model is proposed to study the impact of the servers breakdown on the overall response times to Web requests. The primary aim of the present paper is to modify the performance model of the Proxy Cache Server to a more realistic case when both the Proxy Cache Server and the Web server are unreliable. The main performance and reliability measures are derived, and some numerical calculations are carried out by the help of the MOSEL tool. The numerical results are graphically displayed to illustrate the effect of the non-reliability of the servers on the mean response time.

I. INTRODUCTION

The Web quickly became an indispensable and integral part of todays life. The booming use of the Internet and the World Wide Web has caused congested networks and overloaded servers. As the traffic on the Web continues to increase rapidly, so is the response time delay to requests of Web documents. Adding more network bandwidth is not the best solution. From the user's point of view it does not matter whether the requested files are on the firm's computer or on the other side of the world. The main problem is that the same object can be requested by other users at the same time. Because of this situation, identical copies of many files pass through the same network links, resulting in an increased response time. By preventing future transfer, we can cache information and documents that reduces the network bandwidth demand on the external network, and usually reduces the average time it takes for a web page to load. In general, there are three types of caches that can be used in isolation or in a hierarchical fashion. Caching can be implemented at browser software [1]; the originating Web sites [2]; and the boundary between the local area network and the Internet [3]. Browser cache are inefficient since they cache for only one user. Web server caches can improve performance, although the requested files must delivery through the Internet, increasing the response time. In this paper we investigate the third type. Requested documents can be delivered directly from the Web server or through a Proxy Cache Server (PCS). A PCS has the same functionality as a Web server when looked at from the client and the same functionality as a client when looked at from a Web server. The primary function of a PCS is to store documents close to the users to avoid retrieving the same document several times over the same connection. It has been suggested that, given the current state of technology, the greatest improvement in response time will come from installing a PCS at the boundary between the corporate LAN and the Internet.

The purpose of recent research is to generalize the performance model of a PCS (see [5], [6], [7]) using a more realistic case when the PCS and the remote Web server are unreliable. For the easier understanding of the basic model and comparisons we follow the structure of the cited work. Our aim is to illustrate graphically the effect of the non-reliability of both PCS and Web servers on the steady-state system measures. Furthermore, we examine the difference in the performance using blocked and intelligent sources, see [8]. Because of the fact, that the state space of the describing Markov chain is very large, it is 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 [4], to formulate the model and to obtain the performance measures. By the help of MOSEL we can use various performance tools (like SPNP Stochastic Petri Net Package) to get

*Acknowledgment: This research has been supported by the Hungarian Scientific Research Fund (OTKA) Grant NK81402 and it was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund

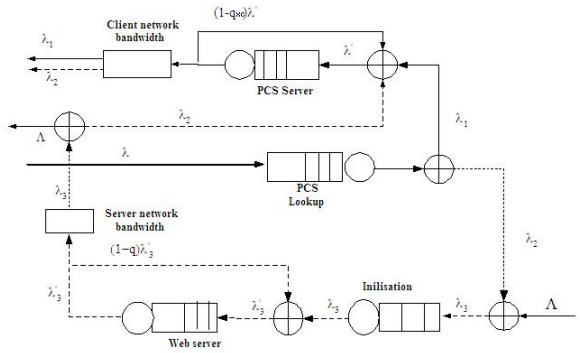

Figure 1: Network Model

these characteristics. The results of the tool can graphically be displayed using IGL (Intermediate Graphical Language) which belongs to MOSEL. The organization of the paper is as follows. Section 2 contains the queuing network model to study the dynamics of installing a PCS, the derivation of the main steady-state performance measures. In Section 3, we present some numerical examples for the models under different service disciplines. The results are graphically displayed using the IGL (Intermediate Graphical Language) interpreter which belongs to MOSEL. By the help of these figures we illustrate the effect of failure and repair rates on the mean response time. The paper ends with Conclusions.

II. A QUEUEING NETWORK MODEL OF PROXY CACHE SERVER

In this section we briefly describe the queueing network model with the suggested modifications. Using PCS, if any information or file is requested to be downloaded, first it is checked whether the document exists on the PCS. We denote the probability of this existence by p. If the document can be found on the PCS then its copy is immediately transferred to the user. In the opposite case the request will be sent to the remote Web server. After the requested document arrived to the PCS then the copy of it is delivered to the user.
Fig. 1 illustrates the path of a request in the modified model starting from the user and finishing with the return of the answer to the user. We assume that the requests of the PCS users arrive according to a Poisson process with rate λ, and the external arrivals at the remote Web server form a Poisson process with rate Λ. The solid line in Fig 1. λ_1 represents the traffic when the requested file is available on the PCS and can be delivered directly to the user. The λ_2 traffic depicted by dotted line, represents those requests which could not be served by the PCS, therefore these requests must be delivered from the remote Web server. Naturally the Web server serves not only the requests of the studied PCS but it also serves requests of other external users. Let Λ be the intensity of these external arrivals. Let λ_3 denote the intensity of the overall requests arriving to the remote Web server. The overall λ_3 traffic undergoes the process of initial handshaking to establish a one-time TCP connection as required in the persistent HTTP protocol, see [7],[11]. Let us denote by I_s this initial setup.
According to [7], "The remote Web server performance is characterized by the capacity of its output buffer B_s, the static server time Y_s, and the dynamic server rate R_s." So, the service rate is given by the the equation, where F is the file size:

$$\mu_{Web} = \frac{1}{Y_S + \frac{B_S}{R_S}} \tag{1}$$

The performance of the firm's PCS is characterized by the parameters B_{xc}, Y_{xc} and R_{xc}. The service rate of the PCS is:

$$\mu_{PCS} = \frac{1}{Y_{xc} + \frac{B_{xc}}{R_{xc}}} \tag{2}$$

If the size of the requested file is greater then the Web server's output buffer it will start a looping process until the delivery of all requested file's is completed. To model this looping, let q be the branching probability that a request from the PCS can be fulfilled at the first try, see [7].

$$q = \min\left(1, \frac{B_s}{F}\right) \tag{3}$$

Also, the PCS have to be modeled by a queue whose output is redirected with probability $1-q_{xc}$ to its input, where $q_{xc} = \min\left(1, \frac{B_{xc}}{F}\right)$

The PCS and the Web server can fail during the interval $(t, t+dt)$ with probability $\delta_{pcs}dt + o(dt)$ and $\delta_{web}dt + o(dt)$ if they are idle, and with probability $\gamma_{pcs}dt + o(dt)$ and $\gamma_{web}dt + o(dt)$ if they are busy, respectively. If the PCS or the Web server fails in busy state, it continues servicing the interrupted request after it has been repaired. The repair time is exponentially distributed with a finite mean $1/\nu_{pcs}$ and $1/\nu_{web}$. If one of the serves is failed two different cases can be treated. Namely, blocked case when during the CPU is down, no new requests come to the server buffer and unblocked case when the new requests can fill the server buffer during the breakdown, until it is full. Note, that in blocked case of the Web server the one time TCP connection will be established. All the times involved in the model are assumed to be mutually independent of each other. As it can be seen this systems is rather complicated since it involves two types of failures: busy or idle server state, blocked and unblocked case during breakdowns.
The system state at time t can be described by the processes

$$X_{PCS}(t) = (Y_{PCS}(t), C_{PCS}(t), Q_{PCS}(t)),$$

and

$$X_{Web}(t) = (Y_{Web}(t), C_{Web}(t), Q_{Web}(t)),$$

where $Y_{PCS}(t) = Y_{Web}(t) = 0$ if the server is up, $Y_{PCS}(t) = Y_{Web}(t) = 1$ if the server is failed, $C_{PCS}(t) = C_{Web}(t) = 0$ if the server is idle and $C_{PCS}(t) = C_{Web}(t) = 1$ if the server is busy, respectively. Let $Q_{PCS}(t)$ and $Q_{Web}(t)$ the number of requests in the buffer, respectively. Because of the exponentiality of the involved random variables these processes are Markov chains with finite state space. Let us define the stationary probabilities by:

$$P_{PCS}(q, r, j) = \lim_{t\to\infty} P(Y_{PCS}(t), C_{PCS}(t), Q_{PCS}(t)),$$

$$q = 0, 1, r = 0, 1, j = 0, \cdots, K_{PCS},$$

and

$$P_{Web}(q, r, j) = \lim_{t\to\infty} P(Y_{Web}(t), C_{Web}(t), Q_{Web}(t)),$$

$$q = 0, 1, r = 0, 1, j = 0, \cdots, K_{Web},$$

where K_{PCS} and K_{Web} are the buffer size of the servers. Once we have obtained the above defined probabilities, the main steady-state system performance measures can be derived as follows:

- *Utilization of the servers*

$$U_{S,PCS} = \sum_{j=0}^{K_{PCS}} P_{PCS}(0, 1, j)$$

$$U_{S,Web} = \sum_{j=0}^{K_{Web}} P_{Web}(0, 1, j)$$

- *Utilization of the repairman*

$$U_{R,PCS} = \sum_{r=0}^{1} \sum_{j=0}^{K_{Pcs}} P_{PCS}(1, r, j)$$

$$U_{R,Web} = \sum_{r=0}^{1} \sum_{j=0}^{K_{Web}} P_{Web}(1, r, j)$$

- *Availability of the servers*

$$A_{PCS} = \sum_{r=0}^{1} \sum_{j=0}^{K_{Pcs}} P_{PCS}(0, r, j) = 1 - U_{R,PCS}$$

$$A_{Web} = \sum_{r=0}^{1} \sum_{j=0}^{K_{Web}} P_{Web}(0, r, j) = 1 - U_{R,Web}$$

- *Mean number of requests at the servers*

$$M_{PCS} = \sum_{q=0}^{1} \sum_{r=0}^{1} \sum_{j=0}^{K_{Pcs}} jP_{PCS}(q, r, j)$$

$$M_{Web} = \sum_{q=0}^{1} \sum_{r=0}^{1} \sum_{j=0}^{K_{Web}} jP_{Web}(q, r, j)$$

- *Mean response times*
  Using the Little formula [9] the mean response times can be derived as follows:

$$T_{PCS} = M_{PCS}/\lambda_{PCS}$$

  where $\lambda_{PCS} = U_{S,PCS} * \mu_{PCS}$ is the mean arrival rate at the PCS,

$$T_{Web} = M_{Web}/\lambda_{Web}$$

  where $\lambda_{Web} = U_{S,Web} * \mu_{Web}$ is the mean arrival rate at the Web server-

- *Overall response time of the requests*

$$T = T_{Lookup} + p * \left(T_{PCS} + \frac{F}{N_c}\right) + (1-p) * \left(T_{Init} + T_{Web} + \frac{F}{N_s} + T_{PCS} + \frac{F}{N_c}\right),$$

  where $T_{Lookup} = \frac{1}{\frac{1}{I_{xc}} - \lambda}$ is the time to check whether the requested file is on the PCS or not and $T_{Init} = \frac{1}{\frac{1}{I_s} - \lambda_3}$ is the time to establish the TCP connection. For more details see [5].

III. NUMERICAL RESULTS

In this section, we present some numerical results in order to illustrate graphically the influence of the error and repair rates of the non-reliable servers in blocked and unblocked cases. For the numerical explorations the corresponding parameters of Cheng and Bose [7] are used. The value of the other parameters for numerical calculations are: $F = 5000$ bytes, $I_s = I_{xc} = 0.004$ seconds, $B_s = B_{xc} = 2000$ bytes, $Y_s = Y_{xc} = 0.000016$ seconds, $R_s = R_{xc} = 1250$ Mbyte/s, $N_s = 1544$ Kbit/s, and $N_c = 128$ Kbit/s. These values are chosen to conform to the performance characteristics of Web servers in [10].

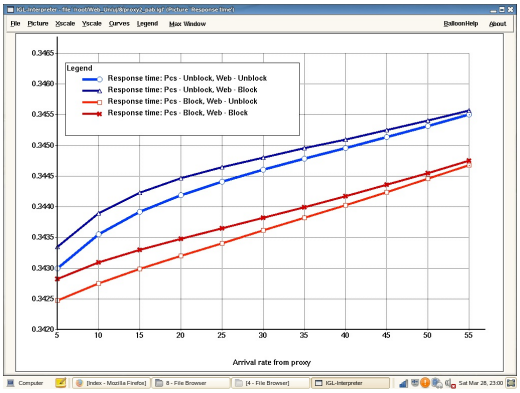

Figure 2: $p = 0.25$, $\Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{pcs} = \gamma_{web} = 0.2$

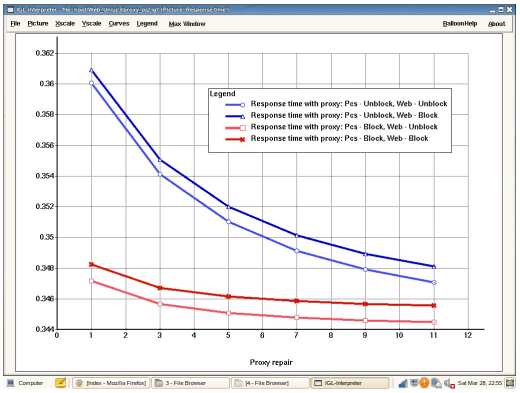

Figure 5: $p = 0.25$,$\lambda = 40, \Lambda = 10$, $\nu_{web} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{pcs} = \gamma_{web} = 2$

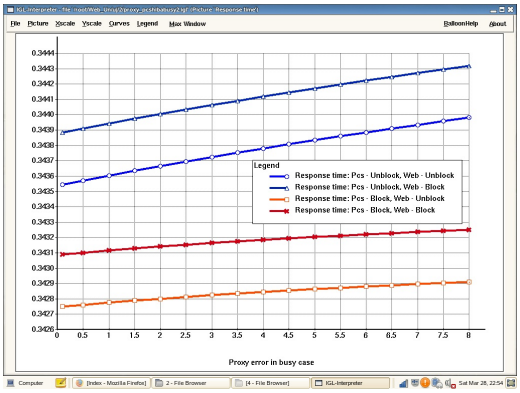

Figure 3: $p = 0.25$,$\lambda = \Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{web} = 0.2$

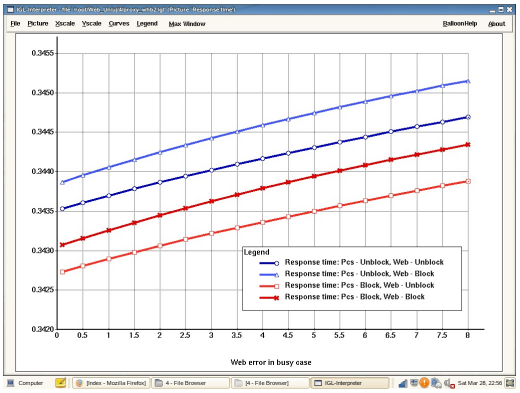

Figure 6: $\lambda = \Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{pcs} = 0.2$

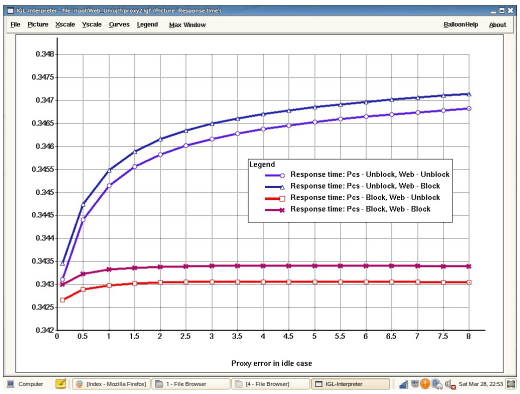

Figure 4: $p = 0.25$,$\lambda = 30, \Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{web} = \gamma_{pcs} = \gamma_{web} = 0.2$

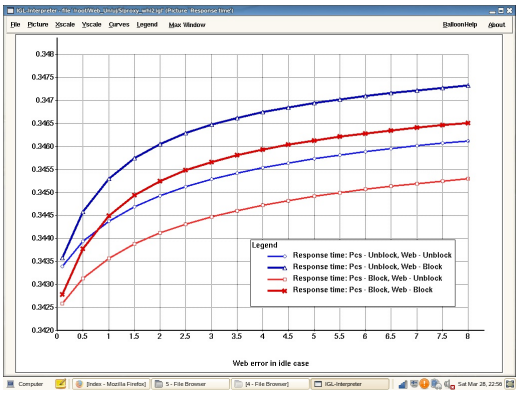

Figure 7: $\lambda = \Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{pcs} = \gamma_{pcs} = \gamma_{web} = 0.2$

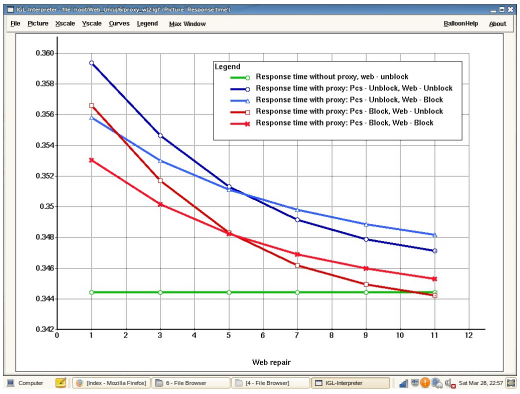

Figure 8: $\lambda = \Lambda = 10$, $\nu_{pcs} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{pcs} = \gamma_{web} = 2$

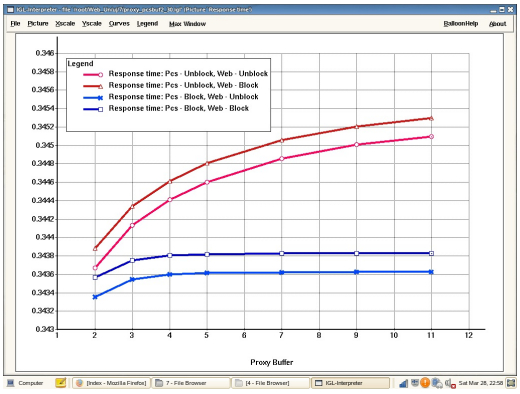

Figure 9: $\lambda = 30, \Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{pcs} = \gamma_{web} = 0.2$

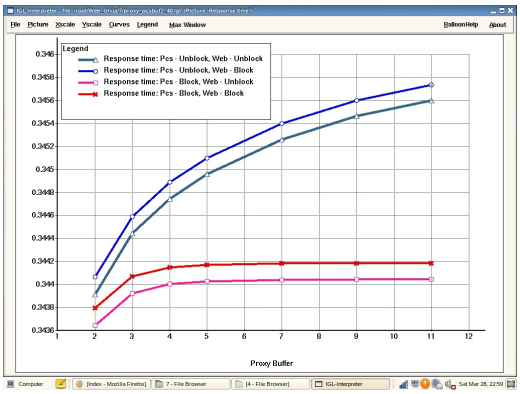

Figure 10: $\lambda = 40, \Lambda = 10$, $\nu_{pcs} = \nu_{web} = 10$, and $\delta_{pcs} = \delta_{web} = \gamma_{pcs} = \gamma_{web} = 0.2$

IV. CONCLUSIONS

Several numerical experiments have been undertaken to examine the performance behavior of the model with respect to various parameter values.

- In Figure 2 we can see the mean response time for the non-reliable system depicted as a function of the arrival rate from proxy (in blocked and unblocked cases). As we see the mean response time increasing in all cases when the arrival rate increasing. We supposed that the difference between the curves, using blocked and unblocked Web server will be vanished increasing the arrival rate. That expectation is shown in this figure.

- In Figure 3-4 the effect of the PCS failure rate is demonstrated on the response time, in busy and idle server states. As we can observe, the mean response time is smaller using unblocked web server. Investigating Figure 4 (the mean response time is depicted as a function of proxy error rate in idle case) it can be observed that when the PCS is in blocked state, the response time will be constatn using larger error rate. In these cases the error and repair rate of the web server are constant. Therefore the curves, where the blocking method of the PCS are the same (both blocked or unblocked), and the blocking method of the Web server are different, are parallel.

- In Figure 5 we investigate the effect of the mean repair time of the PCS. As we see, the response time will be smaller as we increase the repair time. In this case the error and repair rate of the web server are constant. Therefore the curves, where the blocking method of the PCS are the same (both blocked or unblocked), and the blocking method of the Web server are different, are parallel.

- In Figure 6-7 the effect of the web failure rate is demonstrated on the response time, in busy and idle web server states. As we can see, the mean response time will be higher in all cases (blocked and unblocked) using higher error rates. Note, that the main difference between the functions of the Web and PCS error rates come from the fact, that in both blocked and unblocked cases the TCP connection must be established. Investigating the failure rate of the Web server (changing only error rate of the web server in busy and idle server states), the error and repair rate of the PCS remain constant. Because of that fact, we can observe the parallelism between the figures, where the blocking method of the Web server are the same (blocked or unblocked), and the blocking method of the PCS are different.

- In Figure 8 it is shown how the increase of the Web server repair time affects the mean response time. As we could see, the response time will be smaller as we increase the repair time. The parallelism between the specified curves can be observed in this figure too, as we described in the previous comments in Fig 6-7.

- In Figure 9-10 we depict the response time as a function of PCS buffer size. In Figure 9 we use 30 requests/s for arrivals from the PCS. In Figure 10 we use the same parameters, only we use a higher arrival rate from PCS (40 requests/s). When we use smaller arrival rate we get smaller response time.

Notations

λ:	mean arrival rate at the PCS
Λ:	mean external arrival rate
F:	average file size (in byte)
p:	cache hit rate probability
B_{xc}:	PCS output buffer (in byte)
I_{xc}:	lookup time of the PCS (in second)
Y_{xc}:	static server time of the PCS (in sec.)
R_{xc}:	dynamic server time of the PCS (byte/sec.)
N_c:	client network bandwidth (in bit/second)
B_s:	Web output buffer (in byte)
I_s:	lookup time of the Web server (in second)
Y_s:	static server time of the Web server (sec.)
R_s:	dynamic server time of the Web server
N_s:	server network bandwidth (in bit/second)

REFERENCES

[1] AGGARWAL, C., WOLF, J.L. and YU, P.S. Caching on the World Wide Web. *IEEE Transactions on Knowledge and Data Engineering*, **11** (1999), 94-107.

[2] ALMEIDA, V.A.F., DE ALMEIDA, J.M. and MURTA, C.S. Performance analysis of a WWW server. *Proceedings of the 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems*, San Diego, USA, December (1996), 8-13.

[3] ARLITT, M.A. and WILLIAMSON, C.L. Internet Web servers: workload characterization and performance implications. *IEEErACM Transactions on Networking*, **5** (1997), 631-645.

[4] BEGAIN K., BOLCH G. and HEROLD H. *Practical performance modeling, application of the MOSEL language*, Kluwer Academic Publisher, Boston, (2001).

[5] BERCZES, T. and SZTRIK, J. Performance Modeling of Proxy Cache Servers. *Journal of Universal Computer Science*, **12** (2006), 1139–1153.

[6] BERCZES, T., GUTA, G., KUSPER, G., SCHREINER, W. and SZTRIK, J., Analyzing Web Server Performance Models with the Probabilistic Model Checker PRISM. *Technical report no. 08-17 in RISC Report Series* , University of Linz, Austria. November 2008

[7] BOSE, I. and CHENG, H.K. Performance models of a firms proxy cache server. *Decision Support Systems and Electronic Commerce*, **29** (2000), 45–57.

[8] SZTRIK, J., ALMÁSI, B. and ROSZIK, J. Heterogeneous finite-source retrial queues with server subject to breakdowns and repairs. *Journal of Mathematical Sciences*, **132** (2006), 677–685.

[9] LAZOWSKA, E.D., ZAHORJAN, J., GRAHAM, G.S. and SEVCIK, K.C. *Quantitative System Performance*, Prentice Hall, (1984).

[10] MENASCE, D.A. and ALMEIDA, V.A.F. *Capacity Planning for Web Performance: Metric, Models, and Methods*, Prentice Hall, (1998).

[11] SLOTHOUBER L.P. A model of Web server performance. *5th International World Wide Web Conference, Paris, France*, (1996).